IntechOpen

Self Organizing Maps
Applications and Novel Algorithm Design

Edited by Josphat Igadwa Mwasiagi

WEB OF SCIENCE™

SELF ORGANIZING MAPS - APPLICATIONS AND NOVEL ALGORITHM DESIGN

Edited by **Josphat Igadwa Mwasiagi**

Self Organizing Maps - Applications and Novel Algorithm Design
http://dx.doi.org/10.5772/566
Edited by Josphat Igadwa Mwasiagi

Contributors

Yonggang Liu, Robert H. Weisberg, Wattanapong Kurdthongmee, Magnus Johnsson, David Gil, Germund Hesslow, Max Martinsson, Maria Ferentinou, Michael Sakellariou, Eleni Charou, Efthimios Karymbalis, Ryotaro Kamimura, Alessio Plebe, Hector Benitez-Perez, Dimitris Petrilis, Constantin Halatsis, Lidia Angulo Meza, Luiz Biondi Neto, Luana Carneiro Brandão, Fernando Do Valle Silva Andrade, João Carlos Soares de Mello, Pedro Henrique Gouvea Coelho, Hideki Mori, Shingo Kawamura, Minoru Uehara, Dongming Hu, Cheng-Yuan Liou, Wei-Chen Cheng, Masaki Ishii, Mithun J. Sharma, Song Jin Yu, Masaharu Seno, Takayuki Kudoh, Sugii Yuh, Heizo Tokutaka, Takayuki Otani, Masashi Ikeda, Marek B. Zaremba, Agus Buono, Benyamin Kusumoputro, Wisnu Jatmiko, Flavius Luz Gorgonio, Jose Alfredo Ferreira Costa, Hironobu Fukai, Hironori Takimoto, Yasue Mitsukura, Minoru Fukumi, Mahmudur Rahman, Maria Bernardete Cordeiro De Sousa, Adriao Duarte Doria Neto, Allan De Medeiros Martins, Adriano Castro Leão, Dijenaide Chaves De Castro, Dimitris Kontogiorgos, Alexandros Leontitsis, Vilson Luiz Dalle Mole, Aluízio F. R. Araújo, Andrzej Zak, Akira Hirose, Yukimasa Nakano, Akito Sakurai, Koichiro Ishikawa, Yoshihisa Shinozawa, Mohamed Lamjed Bouazizi, Samir Ghanmi, Rachid Nasri, Ahmad Masoud, Josphat Igadwa Mwasiagi, Geovany Borges, Marcio Goncalves, Marcio L. A. Netto, Khadir Tarek, Amir Hajjam El Hassani, Guilherme Barreto, Antonio Varela, Amauri Souza Júnior

Notice

First published in Croatia, 2011 by INTECH d.o.o.
eBook (PDF) Published by IN TECH d.o.o.
Place and year of publication of eBook (PDF): Rijeka, 2019. IntechOpen is the global imprint of IN TECH d.o.o.
Printed in Croatia

Legal deposit, Croatia: National and University Library in Zagreb

Additional hard and PDF copies can be obtained from orders@intechopen.com

Self Organizing Maps - Applications and Novel Algorithm Design
Edited by Josphat Igadwa Mwasiagi

p. cm.

ISBN 978-953-307-546-4

eBook (PDF) ISBN 978-953-51-4526-4

Meet the editor

Dr. Mwasiagi was born in Kenya, where he received his basic and secondary education. Dr. Mwasiagi did his Bachelors degree in PSG College of Technology (Coimbatore-India). He graduated from China Textile University and Donghua University with a masters and doctoral degrees respectively. His doctoral thesis covered modeling of cotton fiber to yarn spinning process using Artificial Neural Network (ANN), modified ANN models and Extreme learning Machines (ELM) algorithms. His industrial work experience includes working in Thika Cloth Mills (Thika) and Rivatex (Eldoret) where he worked as a Spinning Manager. Dr. Mwasiagi joined Moi University in 1997, where he works as a senior lecturer, in the School of Engineering. His area of research includes natural products (fibers, dyes etc), garment design, industrial modeling, re-newable energy, cleaner production and industrial policy formulation. Dr. Mwasiagi has published journal and conference papers covering modeling of fiber to yarn properties, tea dyes, enzymatic desizing, cotton industry in Kenya and the dimensions of the human body.

Contents

Preface

The advent of Self Organizing Maps (SOM) provided an opportunity for scientists to experiment with its ability to solve hitherto complicated problems in all spheres of life. SOM has found application in practically all fields, especially those which tend to handle high dimensional data. SOM can be used for the clustering of genes, in the medical field, the study of multimedia and web-based content and in the transportation industry, just to name a few. The complex data found in meteorological and remotely sensed images commonly acquired using satellite sensing can also be analyzed using SOM.

The impact of SOM in the improvement of human life can not be overstated. The wide application of SOM in many other areas which include data management, data envelopment analysis and manufacturing engineering has enabled a thorough study of its strength and weakness. This has resulted in the design of novel variants of SOM algorithms aimed at addressing some of the weaknesses of SOM.

This book seeks to highlight the application of SOM in varied types of industries. Novel variants of the SOM algorithms will also be discussed.

Dr. Josphat Igadwa Mwasiagi
School of Engineering,
Moi University, Eldoret,
Kenya

Part 1

Data Interpretation and Management

1

Information-Theoretic Approach to Interpret Internal Representations of Self-Organizing Maps

Ryotaro Kamimura
IT Education Center, 1117 Kitakaname Hiratsuka Kanagawa 259-1292 Japan

1. Introduction

In this chapter, we propose a new method to measure the importance of input variables and to examine the effect of the input variables on other components. We applied the method to competitive learning, in particular, self-organizing maps, to demonstrate the performance of our method. Because our method is based upon our information-theoretic competitive learning, it is easy to incorporate the idea of the importance of input variables into the method. In addition, by using the SOM, we demonstrate visually how the importance of input variables affects the outputs from the other components, such as competitive units. In this section, we first state that our objective is to interpret the network configurations as clearly as possible. Then, we show why the importance of input variables should be taken into account. Finally, we will briefly survey our information-theoretic competitive learning and its relation to the importance of input variables.

The objective of the new method is to interpret network configurations, focusing upon the meaning of input variables in particular, because we think that one of the most important tasks in neural learning is that of interpreting network configurations explicitly (Rumelhart et al., 1986; Gorman & Sejnowski, 1988). In neural networks' applications, we have had much difficulty to explain how neural networks respond to input patterns and produce their outputs due to the complexity and non-linear nature of data transformation (Mak & Munakata, 2002), namely, the low degree of human comprehensibility (Thrun, 1995; Kahramanli & Allahverdi, 2009) in neural networks. One of the major approaches for interpretation is rule extraction from trained neural networks by symbolic interpretations with three types of methods, namely, *decompositional*, *pedagogical* and *eclectic* (Kahramanli & Allahverdi, 2009). In the decompositional approach (Towell & Shavlik, 1993; Andrews et al., 1993; Tsukimoto, 2000; Garcez et al., 2001), we analyze the hidden unit activations and connection weights for better understanding of network configurations. On the other hand, in the pedagogical approach (Andrews et al., 1993), the neural network is considered to be a black box, and we only focus upon the imitation of input-output relations exhibited by the neural networks. Finally, in the eclectic approach (Andrews et al., 1993; Barakat & Diederich, 2005), both pedagogical and decompositional approaches are incorporated. In the popular decompositional approach, much attention has been paid to hidden units as well as connection weights. The importance of input variables has been implicitly taken into account. For example, Tsukimoto (Tsukimoto, 2000) used the absolute values of connection weights or the squared connection weights to input variables (attributes) for measuring the importance of input variables. In addition,

(Garcez et al., 2001) pointed out that the pruning of input vectors maintained the highest possible precision.
On the other hand, in machine learning, variable selection or the interpretation of input variables has received much attention. In data processing, the number of input variables has become extremely large (Guyon & Elisseeff, 2003). Thus, it is important to estimate which input variable should be taken into account in actual data processing. Variable selection aims to improve the prediction performance, to reduce the cost in prediction and to understand the main mechanism of data processing (Guyon & Elisseeff, 2003). The third aim is more related to the present paper. To cope with this variable selection, many methods have been developed (Steppe & K. W. Bauer, 1997; Belue & K. W. Bauer, 1995; Petersen et al., 1998) so far. However, we have had few attempts made in the field of unsupervised learning, for example, competitive learning and SOM, to take into account the effect of input variables. The methods for input variables in neural networks are mainly related to supervised learning, because of the easy implementation of the measures to represent the importance of input variables (Guyon & Elisseeff, 2003). Few attempts have been made to apply variable selection to unsupervised learning. Thus, it is necessary to examine the effect of input variables through the visualization abilities of the SOM.
In unsupervised learning, explicit evaluation functions have not been established for variable selection (Guyon & Elisseeff, 2003). We have introduced variable selection in unsupervised competitive learning by introducing a method of information loss (Kamimura, 2007; 2008b;a) or information enhancement (Kamimura, 2008c; 2009). In the information loss method, a specific input unit or variable is temporarily deleted, and the change in mutual information between competitive units and input patterns is measured. If the difference between mutual information with and without the input unit is increased, the target input unit certainly plays a very important role. On the other hand, in information enhancement, a specific input unit is used to enhance competitive units or to increase the selectivity of competitive units. If the selectivity measured by mutual information between competitive units and input patterns is large, the target input unit is important to increase the selectivity.
One of the major difficulties with these information-theoretic methods is that it is extremely difficult to determine how much information should be contained in explicit ways. In those methods, there are some parameters to determine how much information should be acquired. However, there are no ways to adjust the parameters and to determine the appropriate amount of information to be acquired. We must adjust the parameters heuristically by examining final results such as competitive unit output and connection weights. In this context, we propose a new method to measure information content to be stored in input variables. The parameters in the methods are changed to increase this information content as much as possible. The basic principle to determine the parameters is how these parameters can maximize the information of the input variables. Compared with the previous methods, the criterion to determine the parameters is more explicit. With the ability to explicitly determine the information content, we can interpret network configurations with more confidence, because our method presents a network configuration with maximum possible information state.
Our method has been developed based on information-theoretic competitive learning. Thus, our method is the most suited for competitive learning. However, we applied the method to the self-organizing maps, for two reasons. First, the self-organizing map is a convenient tool to visualize the good performance of our method, better than pure competitive learning because the good performance can be intuitively understood by visualization techniques related to the SOM. Second, we think that the self-organizing map is also an attempt to

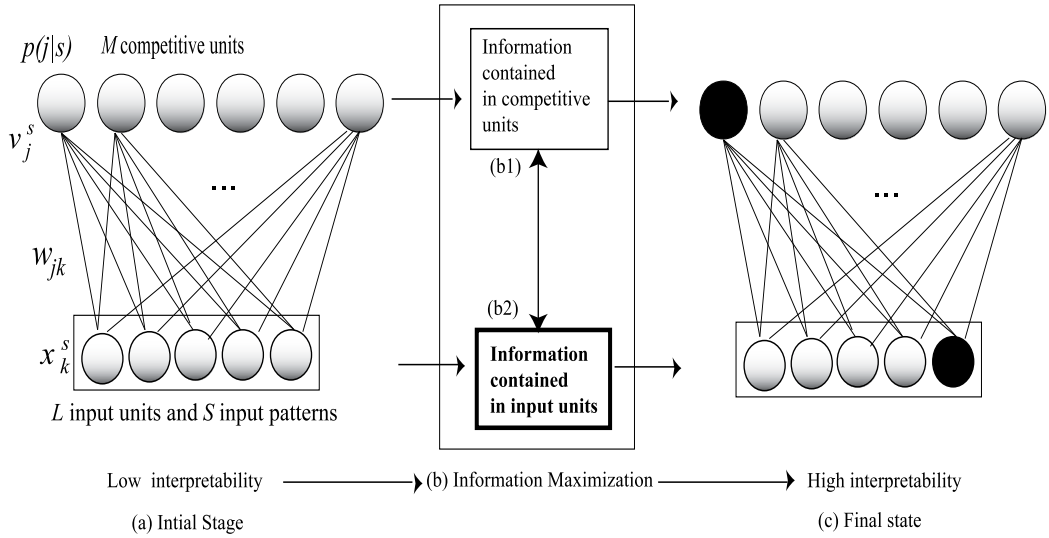


Fig. 1. A concept of the information-theoretic approach.

interpret network configurations not by symbolic but by visual representation. Though the SOM has been developed for clustering and data mining of high-dimensional data (Kohonen, 1988; 1995; Tasdemir & Merenyi, 2009), the SOM's main contribution consists in the visualization of high dimensional data in terms of the lower dimensions with various visualization techniques. In the SOM, different final configurations are made explicit by using various visualization techniques, taking into account codebooks and data distribution (Polzlbauer et al., 2006; Vesanto, 1999; Kaski et al., 1998; Mao & Jain, 1995; Ultsch & Siemon, 1990; Ultsch, 2003). From our point of view, the approach of visual representations to interpret network configurations corresponds conceptually to the decompositional approach in rule extraction, though symbolic representations are not extracted. We think that visualization is an effective tool for interpreting final configurations, corresponding to the extraction of symbolic rules in rule extraction.

2. Theory and computational methods

2.1 Information-theoretic approach

We aim to apply our information-theoretic principle to the detection of the importance of input variables. Principally, our objective is to maximize any information contained in components in a network, hoping that condensed information contained in the components is simpler and more interpretable than that before information maximization. In our sense, information maximization means strictly that information on input patterns is represented in a small number of components, such as competitive units and input units. Figure 1 shows a schematic diagram of our objective. In the figure, from the initial to the final state, the number of important units represented in black is smaller. First, information contained in competitive units must be as large as possible, as shown in Figure 1(b1). We have already shown that this information on competitive units, more exactly, mutual information between competitive units and input patterns, represents competitive processes (Kamimura & Kamimura, 2000; Kamimura et al., 2001; Kamimura, 2003a;b;c;d). Thus, this information, or more exactly, mutual information, should be as large as possible. On the other hand, we can consider

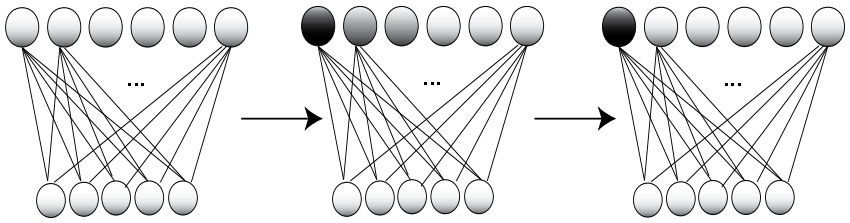


Fig. 2. Competitive unit outputs for an initial state (a), an intermediate state (b) and a state with maximum mutual information (c). The black and white competitive units represent the strong and weak firing rates, respectively.

information content in input units. As shown in Figure 1(b2), this information should be increased as much as possible. When this information is increased, the number of important input variables is decreased. We focus here on input units, or variables, and then information maximization should be biased toward information contained in input units. Thus, mutual information in competitive units should be increased under the condition that the increase in the mutual information prevents a network from increasing information in input units. In the following section, we first explain mutual information between competitive units and input patterns. Then, using the mutual information, we define the importance of input units, by which the information of input variables is defined. Finally, we explain how to compromise these two types of information.

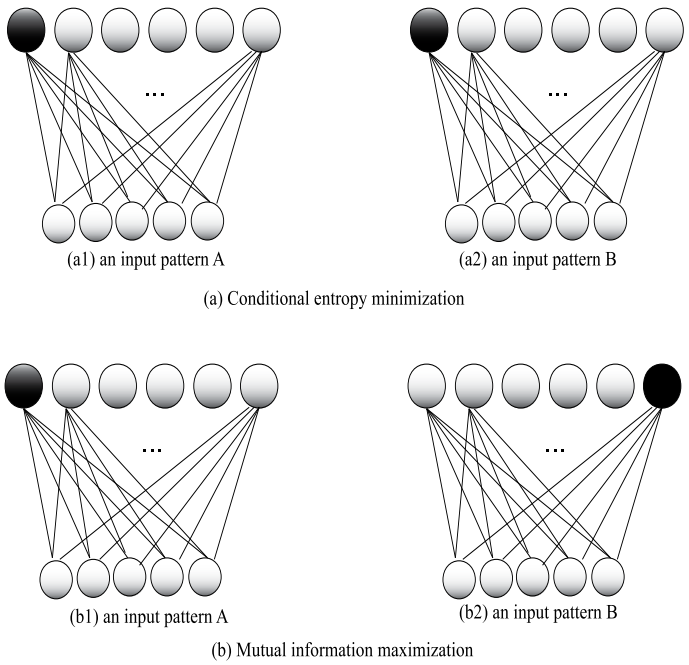


Fig. 3. Competitive unit outputs for conditional entropy minimization (a) and mutual information maximization (b). The black and white competitive units represent the strong and weak firing rates, respectively.

2.2 Information-theoretic competitive learning

We begin with information for competitive units, because information of input units is defined based upon the information for competitive units. We have so far demonstrated that competitive processes in competitive learning can be described by using the mutual information between competitive units and input patterns(Kamimura & Kamimura, 2000; Kamimura et al., 2001; Kamimura, 2003a;b;c;d). In other words, the degree of organization of competitive units can be described by using mutual information between competitive units and input patterns. Figures 2 (a), (b) and (c) show three states that depend on the amount of information stored in competitive unit outputs. Figure 2(a) shows an initial state without any information on input patterns, where competitive unit outputs respond equally to all input patterns. When some quantity of information is stored in competitive unit outputs, several neurons tend to fire at the corners, shown in Figure 2(b). When mutual information between input patterns and competitive units is maximized, shown in Figure 2(c), only one competitive unit is turned on for specific input patterns.

We explain this mutual information more exactly by using the network architecture shown in Figure 1. In the network, x_k^s, w_{jk} and v_j^s represent the kth element of the sth input pattern, connection weights from the kth input to the jth competitive unit and the jth competitive unit output for the sth input pattern. The competitive unit outputs can be normalized as $p(j \mid s)$ to represent the firing probability of the jth competitive unit. In the network, we have L input units, M competitive units and S input patterns.

First, the jth competitive unit outputs v_j^s for the sth input pattern can be computed by

$$v_j^s = \exp\left(-\frac{\sum_{k=1}^{L} p(k)(x_k^s - w_{jk})^2}{2\sigma^2}\right). \tag{1}$$

The firing probability of the jth competitive unit for the sth input pattern can be obtained by normalizing these competitive unit outputs

$$p(j \mid s) = \frac{v_j^s}{\sum_{m=1}^{M} v_m^s}. \tag{2}$$

Then, mutual information between competitive units and input patterns can be defined by

$$\begin{aligned} MI &= \sum_{s=1}^{S}\sum_{j=1}^{M} p(s)p(j \mid s)\log\frac{p(j \mid s)}{p(j)} \\ &= -\sum_{j=1}^{M} p(j)\log p(j) + \sum_{s=1}^{S}\sum_{j=1}^{M} p(s)p(j \mid s)\log p(j \mid s). \end{aligned} \tag{3}$$

Mutual information is decomposed into the first term of entropy and the second term of conditional entropy. As shown in Figure 3(a), when only conditional entropy is minimized, we have the high possibility that only one competitive unit at the corner in the figure is always turned on. On the other hand, when mutual information is maximized, different competitive units respond to different input patterns, as shown in Figure 2(b). Thus, mutual information maximization can realize a process of competition in competitive learning.

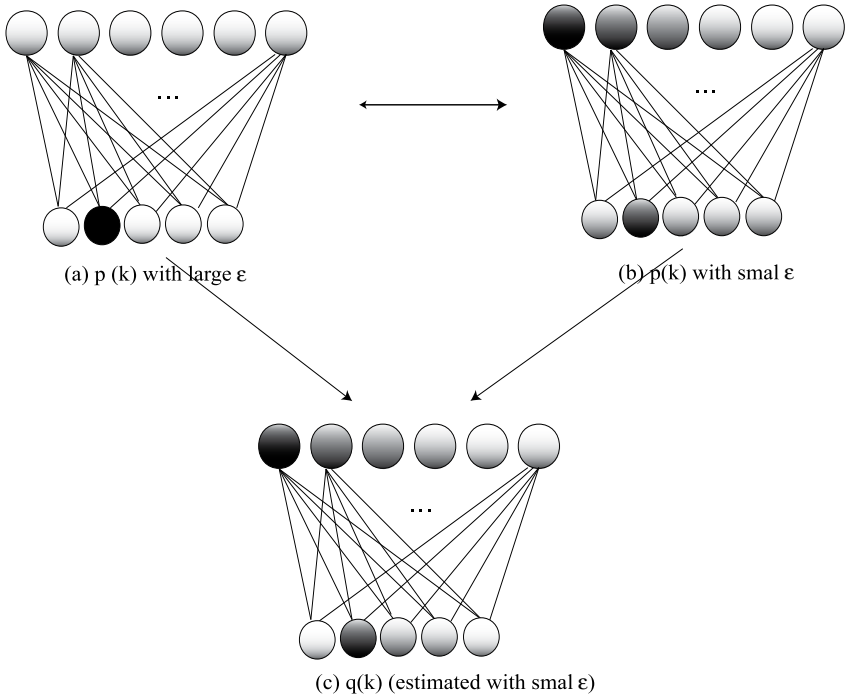


Fig. 4. Importance $p(k)$ with large ϵ (a), small ϵ and estimated importance (c).

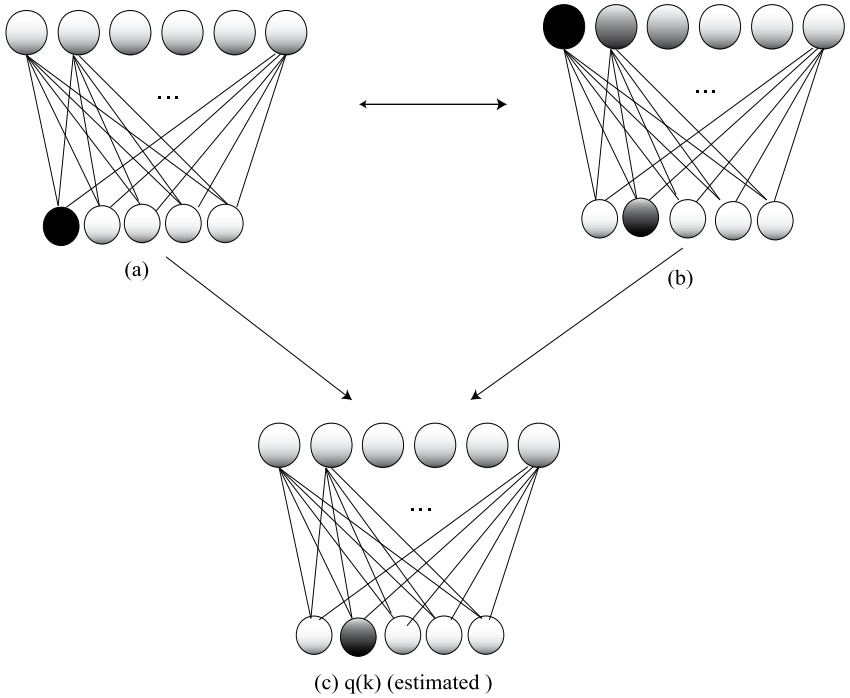


Fig. 5. Importance $p(k)$ with large ϵ (a), small ϵ and estimated importance (c).

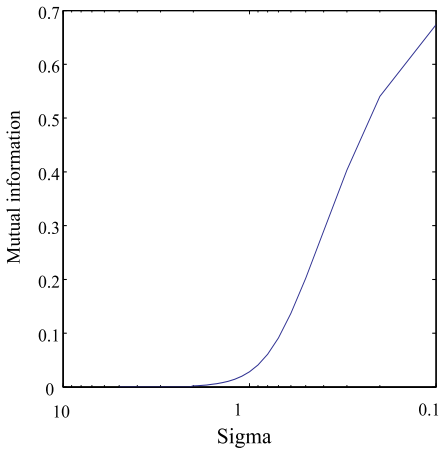


Fig. 6. Mutual information as a function of the parameter σ.

2.3 Estimated information for input variables

Using the mutual information described in the previous section, we try to estimate the importance of input variables. For this purpose, we initially suppose the importance of input units by using the parameter ϵ

$$p(k;t,\epsilon) = \begin{cases} \epsilon, & \text{if } k = t; \\ (1-\epsilon)/(L-1), & \text{otherwise,} \end{cases}$$

where ϵ is a parameter to determine the degree of *attention* paid to the kth input unit. As the parameter ϵ is increased, more attention is paid to the kth target input unit or variable. Figure 4(a) shows a case where the parameter ϵ is the largest value, one, for the second input unit, and the importance of the second input unit is the largest. However, no explicit patterns in terms of competitive unit outputs can be seen. On the other hand, in Figure 4(b), the parameter ϵ is small, the intensity of the second competitive unit is weakened and the other competitive units fires slightly. However, competitive unit outputs are slightly organized. Thus, in this case, the small parameter value of ϵ is better to organize competitive units. Then, the actual importance shown in Figure 4(c) can be modeled by using this small parameter value. Figure 5 shows a case where the first input unit produces no effect on competitive unit output patterns (a), while the second unit produces an organized competitive unit output pattern (b). Thus, the second input unit of the estimated ones is large (c).

To estimate the information, we must introduce the mutual information between competitive units and input patterns. Now, the distance between input patterns and connection weights, when focusing upon the tth input unit, is computed by

$$d_j^s(t,\epsilon) = \sum_{k=1}^{L} p(k;t,\epsilon)(x_k^s - w_{jk})^2. \tag{4}$$

By using this equation, we have competitive unit outputs for the tth input unit

$$v_j^s(t;\sigma,\epsilon) \quad = \quad \exp\left(-\frac{\sum_{k=1}^{L} p(k;t,\epsilon)(x_k^s - w_{jk})^2}{2\sigma^2}\right). \tag{5}$$

Normalizing these outputs, we have

$$p(j \mid s;t,\sigma,\epsilon) = \frac{v_j^s(t;\sigma,\epsilon)}{\sum_{m=1}^{M} v_m^s(t;\sigma,\epsilon)}. \tag{6}$$

The firing probability of the jth competitive unit is defined by

$$p(j;t,\sigma,\epsilon) = \sum_{s=1}^{S} p(s)p(j \mid s;t,\sigma,\epsilon). \tag{7}$$

By using these probabilities, we have mutual information MI when the tth input unit is focused on:

$$MI(t;\sigma,\epsilon) = \sum_{s=1}^{S}\sum_{j=1}^{M} p(s)p(j \mid s;t,\sigma,\epsilon)\log\frac{p(j \mid s;t,\sigma,\epsilon)}{p(j;t,\sigma,\epsilon)}. \tag{8}$$

This mutual information shows how well the tth input unit contributes to a process of competition among competitive units (Kamimura, 2003b).

2.4 Importance of input variables

Mutual information $MI(t;\sigma,\epsilon)$ represents how well the tth input variable contributes to the process of competition. As this mutual information gets larger, the tth input variable plays a more essential role in realizing competitive processes, and the variable should be considered to be important in competition. We approximate the importance of input units with this mutual information, and we have

$$q(t;\sigma,\epsilon) \approx \frac{MI(t;\sigma,\epsilon)}{\sum_{l=1}^{L} MI(l;\sigma,\epsilon)}. \tag{9}$$

Then, using the importance, $q(t;\sigma,\epsilon)$, the estimated information can be defined by

$$EI(\sigma,\epsilon) = \sum_{k=1}^{L} q(k;\sigma,\epsilon)\log\frac{q(k;\sigma,\epsilon)}{q_0(k;\sigma,\epsilon)}. \tag{10}$$

In this equation, q_0 is supposed to be equi-probable, namely, 1/L. As this estimated information gets larger, the number of important input variables gets smaller. Thus, we must increase this estimated information as much as possible, because we are trying to find a small number of important input variables.

2.5 Ratio to determine the parameters

This estimated information EI is based upon mutual information between competitive units and input patterns. Then, mutual information is dependent on the spread parameter σ and ϵ, and in particular, the mutual information is changed by the spread parameter σ. Generally, mutual information can be increased by decreasing the spread parameter σ. Thus, for the parameter σ, the parameter should be as small as possible, meaning that mutual information is as large as possible. Mutual information between competitive units and input patterns represents the degree of organization of a network; as the parameter σ gets smaller, the corresponding mutual information gets larger. This means that, when the parameter σ is small, the organization of a network is large. In addition, the importance of input variables

must be increased as much as possible. Thus, we introduce the ratio RE of the estimated information to the parameter σ

$$RE(\sigma,\epsilon) = \frac{EI(\sigma,\epsilon)}{\sigma}. \tag{11}$$

We try to increase this ratio as much as possible by changing the parameter σ and ϵ. This ratio means that we must increase the estimated information as much as possible. In addition, the mutual information between competitive units and input patterns must be as large as possible, which is realized by the property that, when the parameter σ is smaller, the mutual information is larger.

2.6 Self-organizing maps

Finally, we should note the conventional self-organizing maps (SOM) used in this chapter. Principally, the SOM is a method to increase mutual information that takes into account interaction among competitive units. The reason why we use the SOM as a basic learning method is that we have some difficulty in implementing lateral interactions in competitive output units from information-theoretic points of view[1]. In the SOM, at each training step, the data set is partitioned according to the Voronoi regions of map vectors. First, we must select the best matching unit (BMU), denoted by c:

$$c = \mathrm{argmin}_j \sum_{k=1}^{L} (x_k^s - w_{jk})^2. \tag{12}$$

This selection of the BMU corresponds to a case where mutual information between competitive units and input patterns is maximized. Then, we must compute a neighborhood kernel, h, around the winning unit c.

$$h_{jc} = \exp\left(-\frac{\| \mathbf{r}_c - \mathbf{r}_j \|^2}{2\sigma^2}\right), \tag{13}$$

where $\mathbf{r}_c$ and $\mathbf{r}_j$ denote vectors representing the position of the winner and jth competitive unit, respectively, and σ is a neighborhood radius. Connection weights w_{jk} are computed by

$$w_{jk} = \frac{\sum_{s=1}^{S} h_{jc} x_k^s}{\sum_{s=1}^{S} h_{jc}}. \tag{14}$$

We can say that the SOM is also one of the methods that increases mutual information between competitive units and input patterns.

3. Results and discussion

3.1 Experimental results

3.1.1 Symmetry data

We first applied the method to symmetric data in which input patterns are symmetric, as shown in Figure 7(a). Therefore, the method must detect this symmetric property at least. Figure 7(b) and (c) show a U-matrix and labels obtained by the conventional SOM. As can be seen in the figure, in the middle of the U-matrix, clear boundaries in warmer colors can be

[1]We will discuss this problem in the discussion section.

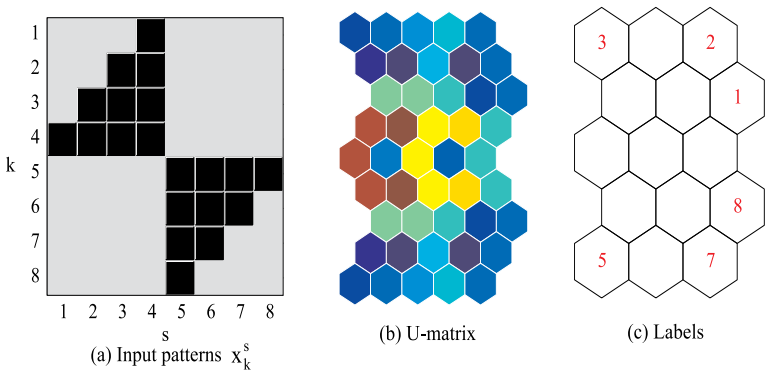


Fig. 7. Original data x_k^s (a), U-matrix (b) and labels (c) for the symmetric data obtained by the SOM.

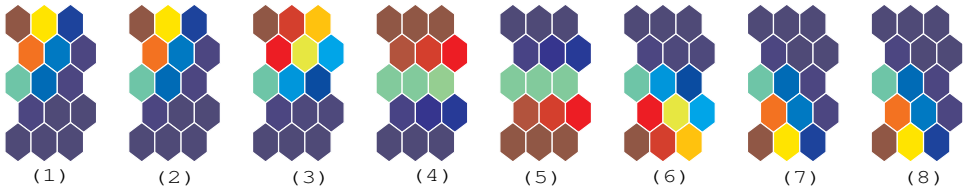


Fig. 8. Component planes along eight input variables obtained by the SOM for the symmetric data.

detected. We can see from the labels in Figure 7(c) that input patterns are naturally classified into two classes. Figure 8 shows component planes along six input units. As component planes move from the first input unit to fourth input unit, they show a gradual increase in the number of strong connection weights (in warmer colors) on the upper part of the map. On the other hand, component planes move from the fifth input unit to the eighth input unit, and then they show a gradual increase in the number of strong connection weights on the lower part of the map. This means that the importance of component planes becomes larger as the component planes move to the center. This property of component planes explains well the symmetric property of the original data.

Figure 9(a) shows estimated information $EI(\sigma,\epsilon)$ as a function of the spread parameter σ for six different values of the parameter ϵ. The computational procedure is as follows. First, the parameter ϵ is chosen; for example, ϵ is 0.2. Then, we try to increase the estimated information EI as much as possible. As shown in Figure 9(a), when the parameter ϵ is set to 0.2, then the other parameter σ is increased up to 1.1, where the estimated information reaches its steady state. Beyond this point, the estimated information cannot be increased. Learning is considered to be finished when the difference in estimated information between the present and the previous state is less than 0.001. We can see that, when the parameter ϵ is larger, the estimated information is larger. In other words, when we focus upon a specific input variable more intensely, the estimated information becomes larger. In addition, we can see that, when the estimated information is larger, the other parameter σ is also increased. To see the situation more exactly, we plot the relations between the two parameters, σ and ϵ. Figure 9(b) shows the final estimated information, with the final value of the parameter σ as a function of the parameter ϵ. The estimated information is increased and reaches its steady state as the parameter ϵ is increased. Figure 9(c) shows the values of the parameter σ as a

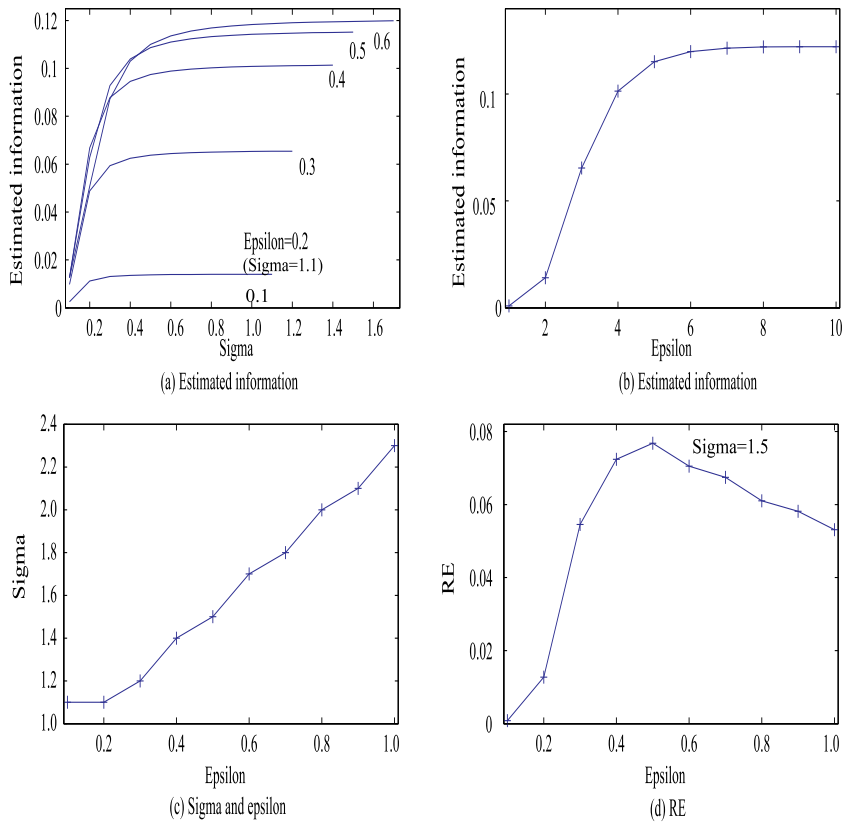


Fig. 9. Information as a function of the parameter σ (a) and the parameter ϵ (b). Optimal values of the parameter σ as a function of the parameter ϵ (c). The ratio RE as a function of the parameter ϵ.

function of the other parameter ϵ. The parameter σ is increased constantly as the parameter ϵ is increased. As mentioned above, for the mutual information between competitive units and input patterns to be increased, the parameter σ should be as small as possible. Therefore, we have introduced the ratio RE. This ratio is gradually increased, and it reaches its peak when the parameter ϵ is 0.5. Thus, this value of 0.5 produced the optimal information, where the estimated information is sufficiently high, and in addition, mutual information between competitive units and input patterns is not so small, because the parameter σ is relative small. Figure 10 shows the estimated importance $q(k)$ for four different values of the parameter ϵ. Figure 10(a) shows the importance when the parameter ϵ is 0.1. The estimated importance is flat, and little difference can be seen. As the parameter is increased from 0.2 (b) to 0.3 (c), gradually, the importance of different input units is made clearer. Finally, when the parameter ϵ is 0.5 (d, optimal), the range of the importance is the largest, and we can easily see the symmetric property of the data. As the input variable moves to the center, the importance of input variable naturally increases. These results demonstrate that the ratio of the estimated information to the parameter σ shows the most interpretable importance of input variables. In addition, we plot the estimated firing probability $p(j)$ with the optimal values of the parameters in Figure 10(e) and (f). As the probability $p(j)$ increases, the corresponding

competitive unit responds to the larger number of input patterns. As can be seen in the figure, the higher values of the probability can be seen in the middle of the map. This means that competitive units in the middle respond to many input patterns. On the other hand, competitive units on the upper and lower parts of the map respond to a fewer number of input patterns, which means that competitive units on the upper and lower parts of the map respond very selectively to input patterns. Thus, we can say that the high probabilities represent a boundary between classes. As can be seen in the figure, input patterns can be classified into two groups by the competitive units with high probabilities in the middle.
Figure 11 shows results when the network size is small (a) and large (b). When the network is small, the same results in terms of U-matrix, RE, importance and $p(j)$ can be obtained. On the other hand, when the network is large, the obtained U-matrix in Figure 11(b1) shows the detailed classification of input patterns, while the probability $p(j)$ (Figure 11(b4)) clearly shows a boundary in the middle of the map. The ratio RE and the importance show almost the same results as those obtained by the normal-sized network.

3.1.2 Student survey No. 1: an image of a university

Second, we applied the method to a student survey in which 39 students at a university were asked to answer nine questions on the good points of a university[2]. The evaluation scores ranged between five (the most favorable) and one (least favorable). Figures 12(a) and (b) show a U-matrix and labels by the conventional SOM. As can be seen in the figure, some class boundaries seem to be present on the lower part of the matrix. Figure 13 shows component planes along nine input variables. Component planes become larger for the lower part of the map. This means that the lower part of the map is a group of students with a more favorable image concerning the input variables. However, we could not estimate the characteristics of student groups separated by these boundaries. Thus, the visualization ability of the SOM is incompetent at dealing with this problem.
Figure 14(a) shows the estimated information as a function of the parameter σ for six different values of the other parameter ϵ. As the parameter ϵ is increased, the estimated information is increased, and the corresponding value of the parameter σ tends to increase also. Figure 14(b) shows the final estimated information for the parameter σ as a function of the parameter ϵ. Information is increased gradually and reaches its steady state as the parameter ϵ is increased. Figure 14(c) shows the value of the parameter σ as a function of the parameter ϵ. As the parameter ϵ is increased, the parameter σ is linearly increased. Figure 14(d) shows the ratio of the estimated information by the parameter σ. When the parameter ϵ is 0.4, the largest value of the ratio is obtained. This means that with this value of 0.4, the estimated information is the largest, with reasonable high mutual information. Figure 15 shows the importance $q(k)$ of nine input variables. The values of the importance are gradually increased from $\epsilon = 0.1$ (a) to $\epsilon = 0.4$ (optimal). Then, the largest range of the importance can be obtained when the ratio takes its optimal value of 0.4. As can be seen in the figures, gradually, input variable No. 4 (teachers' attitude toward students) becomes larger and plays important roles to make self-organizing maps more organized. Finally, Figure 15(e) shows that, in the middle of the map, there are competitive units with higher probabilities $p(j)$. This means that input patterns are classified on this boundary in the middle of the map.
Then, we examine where these measures of the importance and estimated information are independent of map size. For this purpose, we prepare a small- and large-sized network and

[2]This survey was conducted by Mr. Kenta Aoyama in 2010 for 41 students. We deleted two students whose evaluation scores toward the variables were zero or five for all questions.

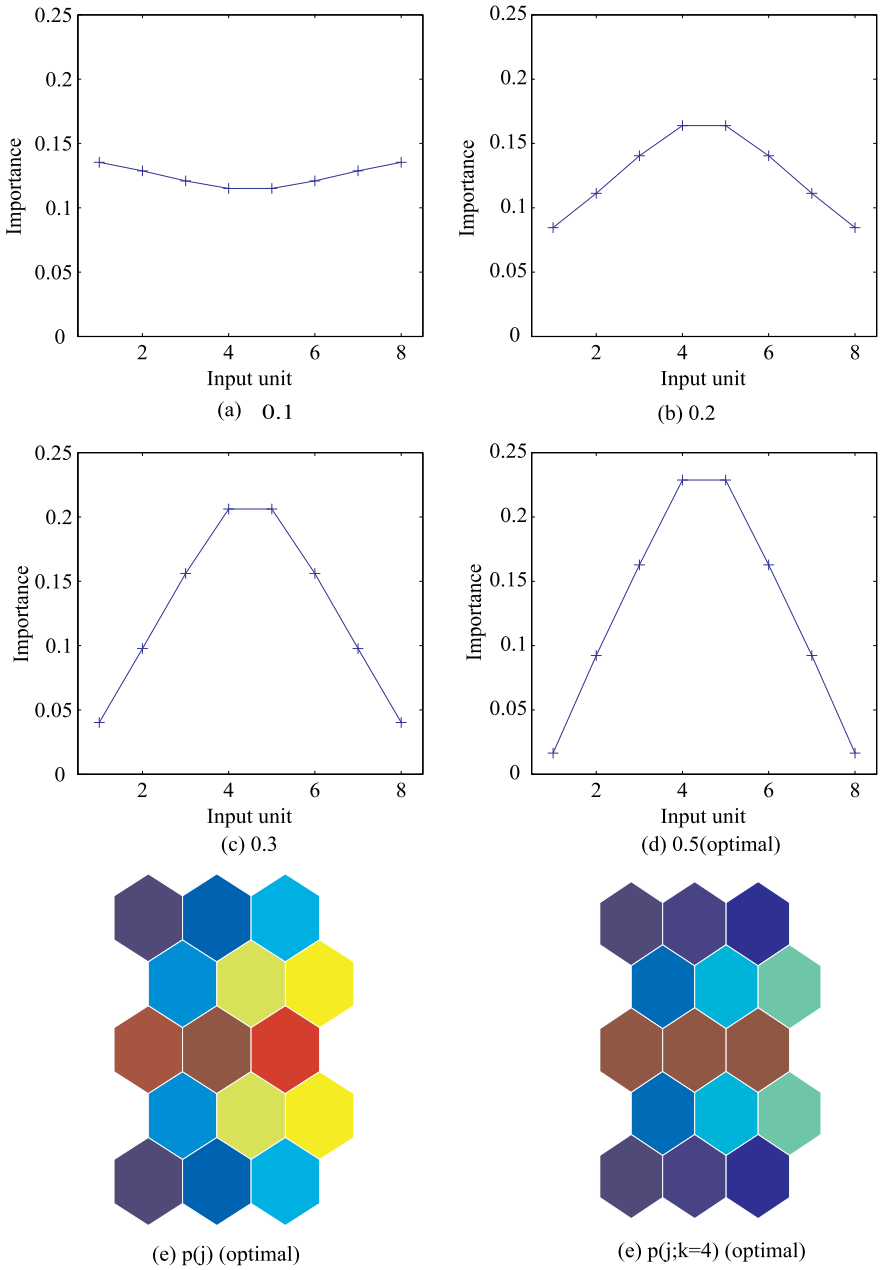


Fig. 10. Estimated importance for four different values of the parameter ϵ (a)-(d) and the estimated $p(j)$ with the optimal values of two parameters (e), (f).

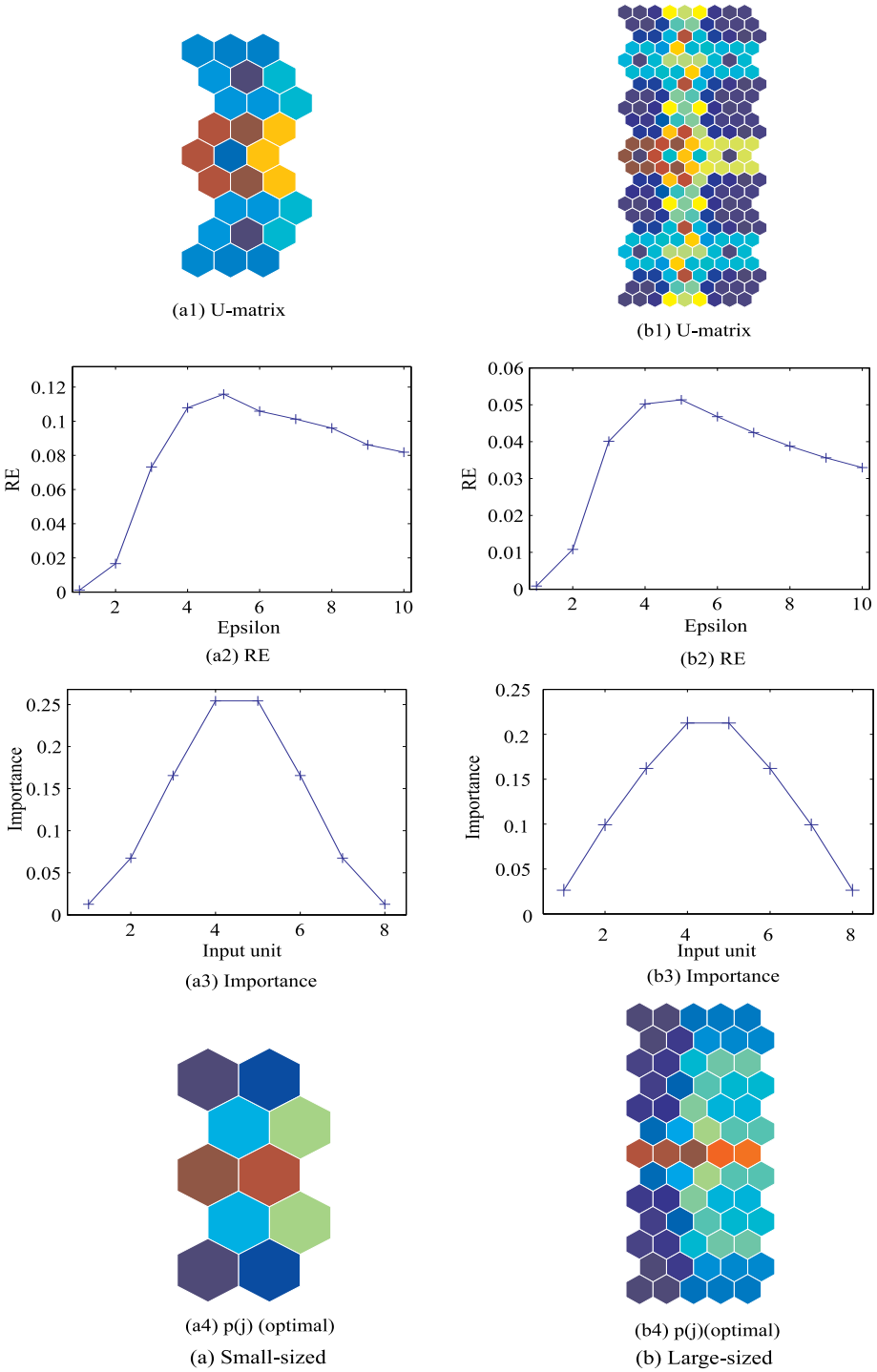


Fig. 11. U-matrices (1), the ratio RE (2), the values of importance (3) and probabilities $p(j)$ for the small-sized (a) and large-sized network (b).

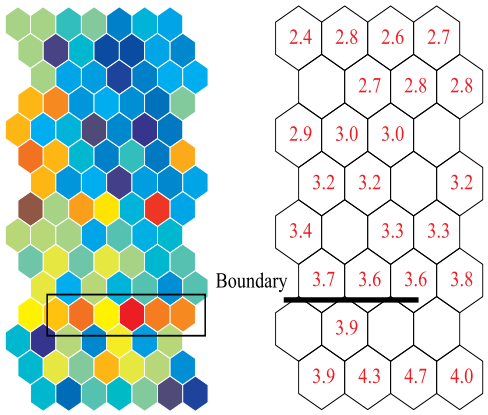


Fig. 12. Original data (a), U-matrix (b) and labels (student No.) (c) obtained by the SOM for student survey No. 1.

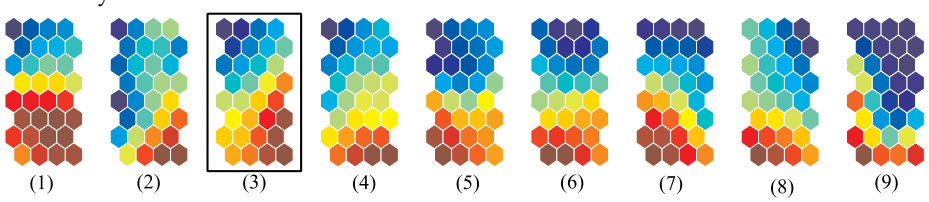


Fig. 13. Component planes along nine input variables obtained by the the SOM for student survey No.1.

then compute the estimated information. Figures 16(a1) and (b1) show final U-matrices for the small and large network, respectively. Figures 16(a2) and (b2) show the ratio RE as a function of the parameter ϵ. We can see that the best values are obtained when the parameter ϵ is 0.4, which is the same as that with the normal-sized network. Figures 16(a3) and (b3) show the values of importance for nine input variables. As can be seen in the figures, variable No. 4 (teachers' attitude) plays the important role, as is the case with the normal-sized network. In addition, we can observe that the range of importance $q(k)$ is slightly decreased when the network size is larger. Figures 16(a4) and (b4) show the probabilities $p(j)$. When the size is small, in Figure 16(a4), a clear boundary in the middle can be seen. However, when the size is large, in Figure 16(b4), the boundary becomes very wide.

3.1.3 Student survey No. 2: mass media and urbanization

We conducted a student survey on to what extent students wanted to live in an urban area or abroad and this desire's relation to mass media[3]. Figure 17 shows the U-matrix (a) and the corresponding labels (b). Some class boundaries in warmer colors seem to be located on the upper part of the map. Figure 18 shows component planes along eight input variables. Though those on the lower part of the maps tend to be stronger, in warmer colors, eight component planes are different from each other. In this problem, we cannot detect the major characteristics among groups separated by the supposed boundaries in warmer colors in the U-matrix.

[3]This survey was also conducted by Kenta Aoyama, December 2009.

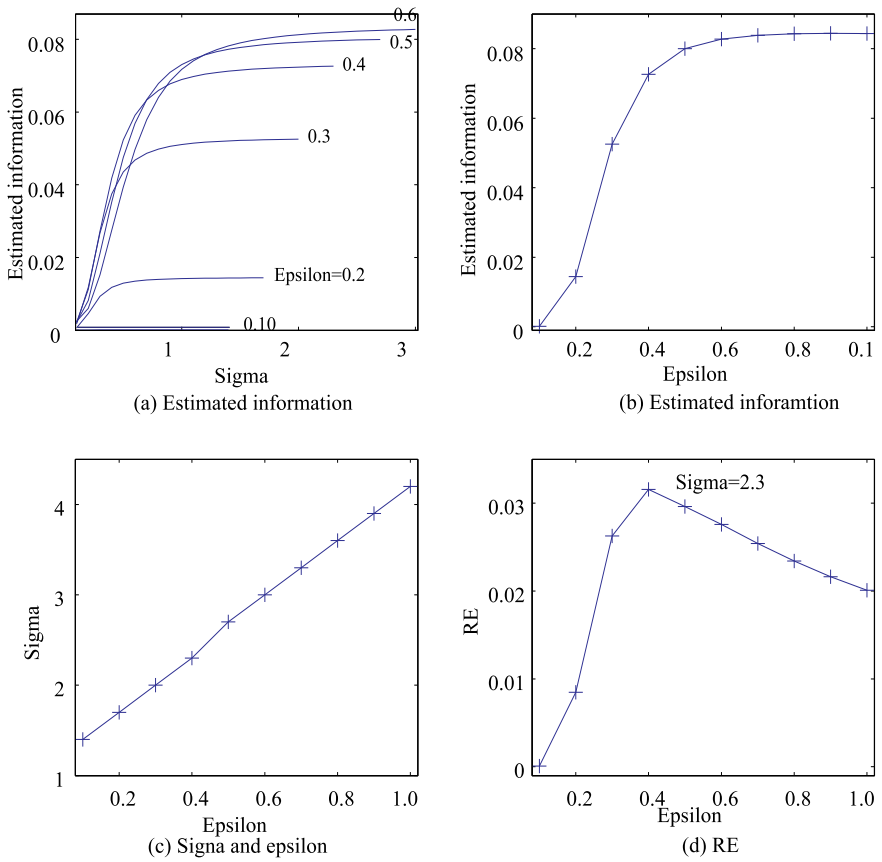


Fig. 14. Information as a function of the parameter σ (a) and the parameter ϵ (b); optimal values of the parameter σ as a function of the parameter ϵ (c); the ratio RE as a function of the parameter ϵ; and competitive unit output $p(j)$ when the information is maximized (d).

Figure 19(a) shows the estimated information as a function of the parameter σ for six different values of the parameter ϵ. As the parameter ϵ gets larger, the estimated information becomes larger and the other parameter σ is larger. Figure 19(b) shows the estimated information as a function of the parameter ϵ. As can be seen in the figure, the estimated information is gradually increased as the parameter ϵ grows larger. Figure 19(c) shows the values of the parameter σ as a function of the parameter ϵ. As the parameter ϵ is increased, the corresponding values of the parameter σ become larger. Figure 19(d) shows the ratio RE of the estimated information to the parameter σ. When the parameter ϵ is 0.4, the largest value can be obtained. Figure 20 shows the importance $p(k)$ for eight input variables. When the parameter ϵ is 0.1, the values of importance is quit small, and we cannot see any characteristics in the figure. As the parameter ϵ is increased from 0.2 (b) and 0.3 (c), the importance of the value of input variable No. 4 rises gradually. Then, when the parameter ϵ is 0.4, we have the largest value of the importance. As can be seen in the figures, input variable No. 4 shows the largest value of importance, namely, "to live abroad." This feature plays the most important role in organizing competitive unit outputs. This means that students are classified into several groups based upon this feature. Figure 20(e) shows the probability $p(j)$ with the optimal

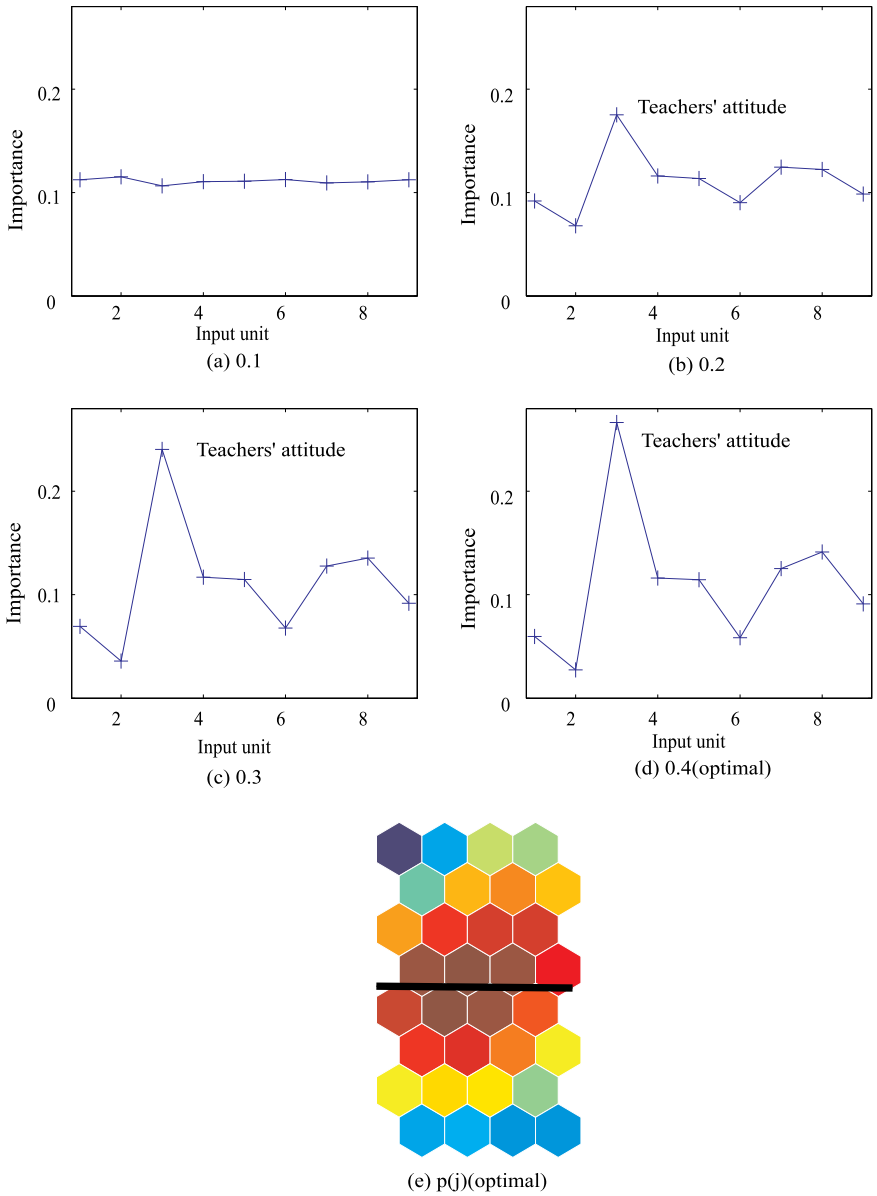


Fig. 15. Estimated importance for four different values of the parameter ϵ (a)-(d) and the probability $p(j)$ (e).

values of the parameters. As can be seen in the figure, a clear, diagonal boundary in warmer colors is located on the map. Students with higher scores and lower scores are classified by this boundary.

Then, we try to see whether these characteristics can be obtained when the network size is different. Figure 21 shows the U-matrices, the ratio RE and the importance for two different sizes of network. When a network is small or large, the optimal value of the parameter ϵ is 0.4, as shown in Figures 21(a2) and (b2), and the importance of input variable No. 4 is the

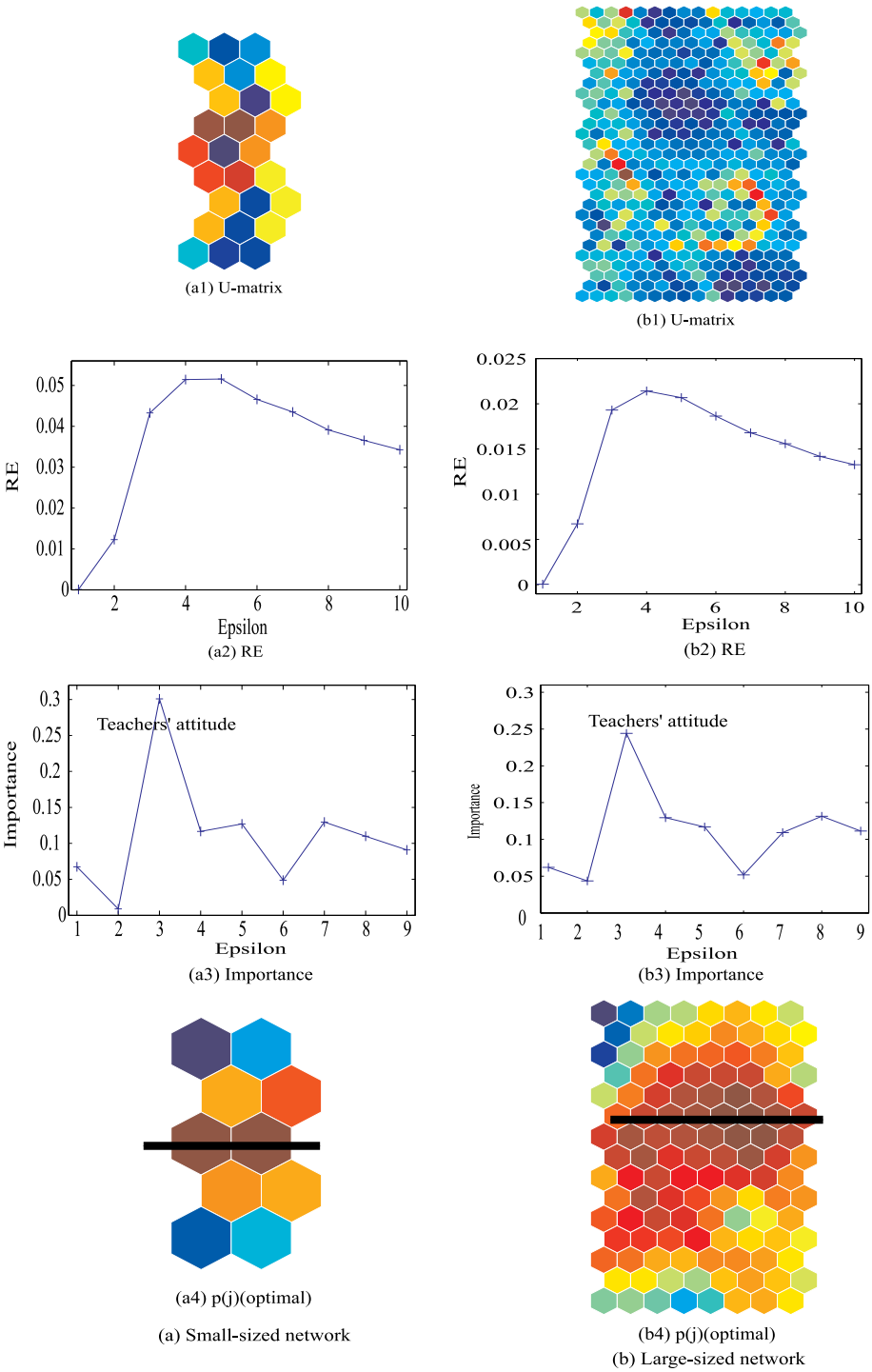


Fig. 16. U-matrices (1), the ratio RE (2), the values of importance (3) and the probability $p(j)$ (4) for the small-sized (a) and large-sized network (b).

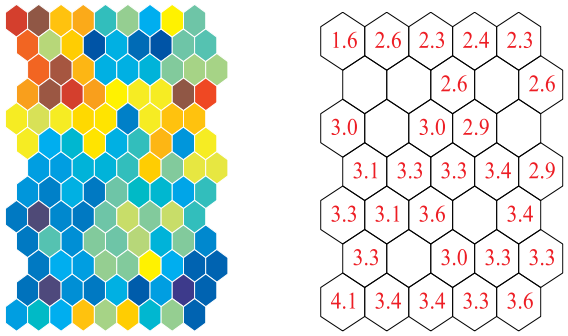


Fig. 17. Original data (a), U-matrix (b) and labels (c) obtained by the SOM for student survey No. 2.

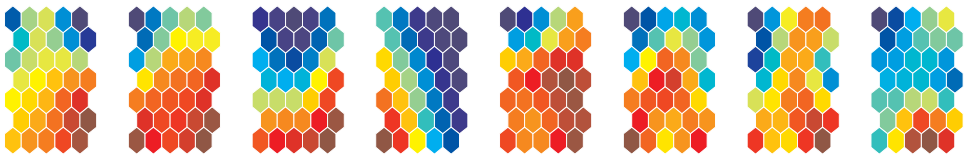


Fig. 18. Component planes obtained by the SOM for student survey No. 2.

largest for both networks in Figures 21(c1) and (c2). These values are exactly the same as those obtained by the normal-sized network, though the range of the importance values becomes smaller as the network size grows larger. In addition, when the network is large, in Figure 21(b4), a clear boundary is diagonally located. On the other hand, in the U-matrix in Figure 21(b1), this boundary cannot be seen. The experimental results suggest that the characteristics obtained by our methods are greatly independent of the size of the network.

3.2 Voting attitude problem

In this experiment, we use the voting attitude data from the machine learning data base[4]. The data set includes votes for each of the U.S. House of Representatives Congresspersons on 16 key votes, and the number of input patterns is 435. Figures 22(a) and (b) show the U-matrix and labels for the voting attitude problem obtained by the conventional SOM. As can be seen in Figure 22(a), a boundary in warmer colors can be detected in the middle of the U-matrix. The labels in Figure 22(b) show that input patterns are classified into two groups, Republicans and Democrats, in the middle of the map. Figure 23 shows component planes along the 16 input variables. As later shown in Figure 25, the input variables No.5, No.8 and No.4 have the larger values of information. They clearly represent two groups in the component planes. Figure 24(a) shows the estimated information as a function of σ for six different values of the parameter σ. We notice two important points. First, the parameter σ is decreased when the parameter ϵ is increased from 0.1 to 0.3, and then the parameter σ is increased. Second, the estimated information is not necessarily increased as the parameter ϵ is increased. Figure 24(b) shows the estimated information as a function of the parameter ϵ. As can be seen in the figure, the information is increased to a maximum point when the parameter ϵ is increased to 0.3, and

[4]http://www1.ics.uci.edu/ mlearn/MLRepository.html

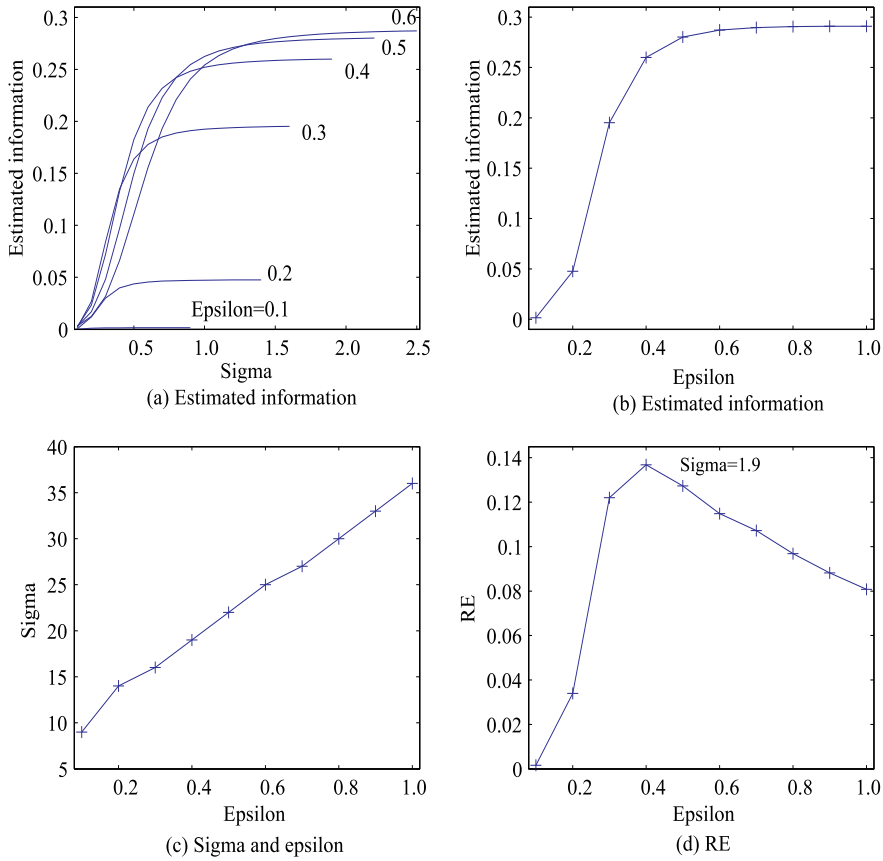
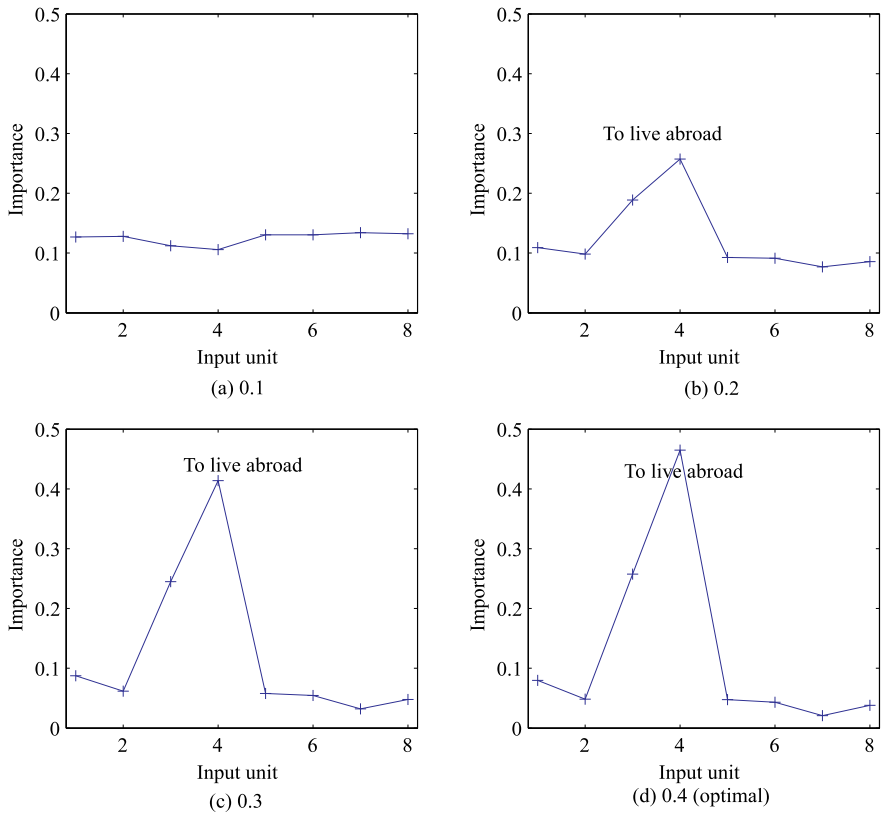


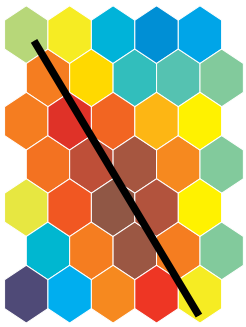
Fig. 19. Information as a function of the parameter σ (a) and the parameter ϵ (b). Optimal values of the parameter σ as a function of the parameter ϵ (c). The ratio RE as a function of the parameter ϵ (d).

then information slowly decreases. Figure 24(c) shows a relation between the parameter σ and the ϵ. When the parameter ϵ is 0.2 and 0.3, the parameter σ takes the lowest value. Figure 24(d) shows the ratio of information to the parameter σ. As can be seen in the figure, the ratio takes a maximum value when the parameter ϵ is 0.3.

Figure 25 shows the importance of 16 input variables. As the parameter ϵ gets larger, the range of values of the importance is increased. Finally, when the parameter ϵ becomes 0.3, the range of the importance becomes the largest. As can be seen in the figure, variables No. 5, No. 8 and No. 4 have larger importance. Figures 26(a) and (b) show the probability $p(j)$ and $p(j;k=5)$ with the optimal values of the parameters. In the figures, a clear boundary in warmer colors can be seen in the middle of the map.

Figure 27 shows results when the network size is small (a) and big (b). Figures 27(a1) and (b1) shows the U-matrices for the small size and big size, respectively. As can be seen in the figures, for the small network, a clear boundary in warmer colors clearly shows two groups, while for the large network, smaller boundaries are scattered on the U-matrix and clear boundaries disappear. Figures 27(a2) and (b2) show the ratio RE as a function of the parameter ϵ. Though the values for the large network become smaller, almost the same tendency can be detected.





(e) p(j) (optimal)

Fig. 20. Estimated importance for four different values of the parameter ϵ (a)-(d) and the probability $p(j)$.

Figures 27(a3) and (b3) show the values of the importance. As can be seen in the figure, the values for the large network become smaller, but the tendency of the importance remains the same. Figures 27(a4) and (b4) show the values of $p(j)$ for the small and large size. As can be seen in the figure, even if the network is large, a boundary in the middle of the map can be seen, while for the small size, a clear boundary can be generated.

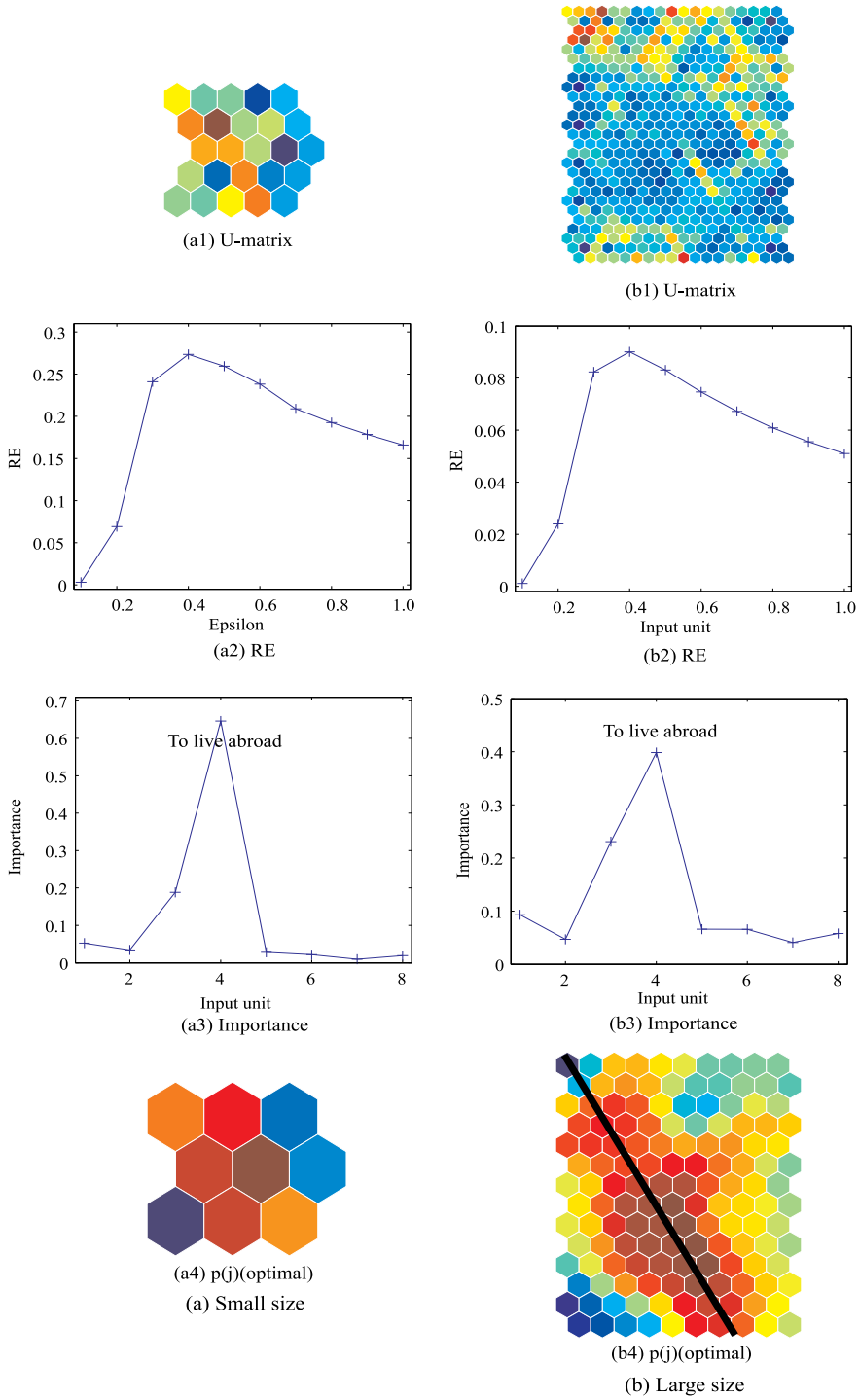
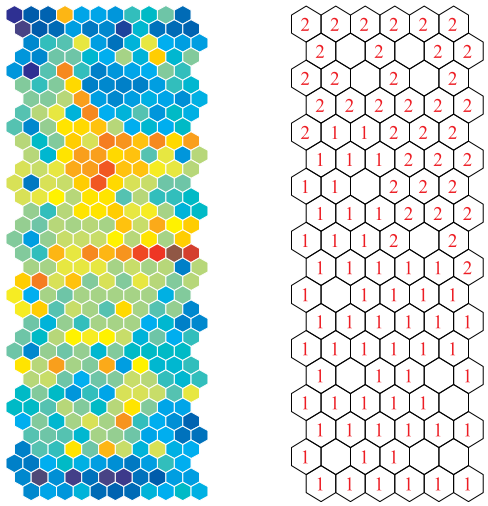


Fig. 21. U-matrices (1), the ratio RE (2), the values of importance (3) and $p(j)$ (4) for the small-sized (a) and large-sized network (b).



(a) U-matrix (b) Labels

Fig. 22. Original data (a), U-matrix (b) and labels (c) obtained by the SOM for the voting attitude problem.

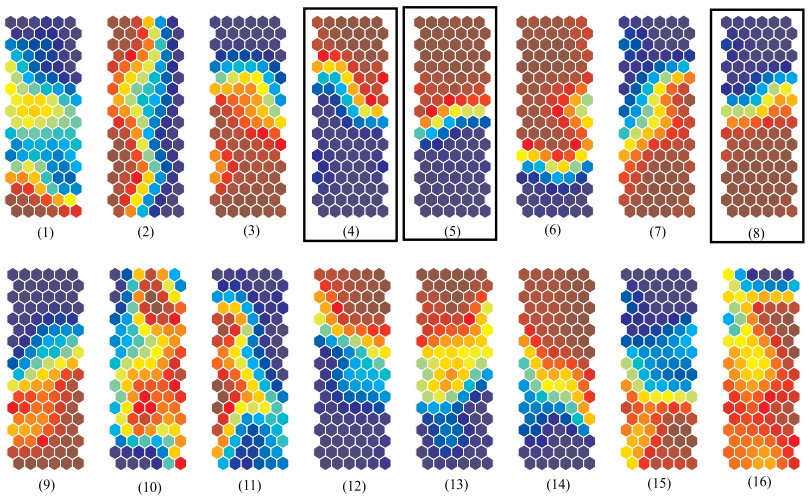


Fig. 23. Component planes obtained by the SOM for the voting attitude problem obtained.

3.3 Discussion

3.3.1 Validity of experimental results

In this paper, we have proposed a new type of information-theoretic method to measure the importance of input variables. The importance of input variables is approximated by mutual information, focusing upon a specific input variable. Then, using this importance, the information content of input variables is computed. As the information gets larger, the number of important input variables becomes smaller. Thus, we try to increase this information as much as possible. We have applied the method to four problems, namely, a symmetric data set, two actual student survey data sets and the voting attitude problem. Experimental results have clarified four points, namely, the number of important input variables, the determination of optimal information, the independency from the network size

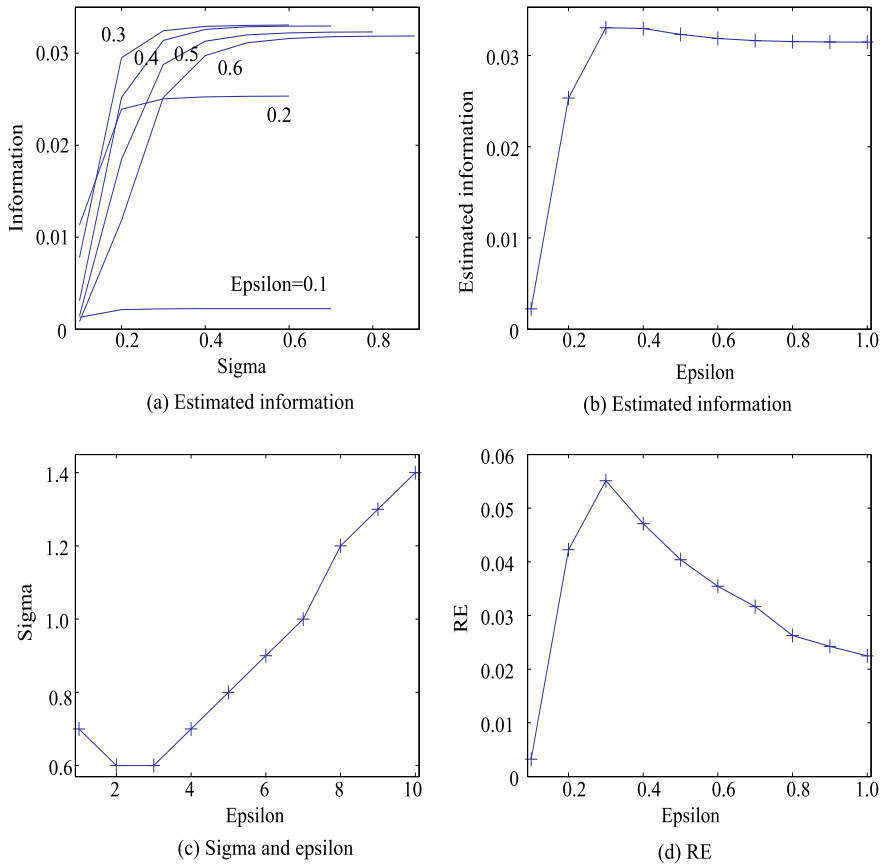


Fig. 24. Information as a function of the parameter σ (a) and the parameter ϵ (b). Optimal values of the parameter σ as a function of the parameter ϵ (c) for the voting attitude problem. The ratio RE as a function of the parameter ϵ (d).

and relations to variance. First, experimental results have confirmed that the smaller number of important input variables is detected for two data sets of an actual student survey. In student survey No. 1, the input variable representing "teachers' attitude" shows by far the largest value of importance. On the other hand, in student survey No. 2, the input variable "to live abroad" has by far the largest value of importance. These input variables are measured by the mutual information between competitive units and input patterns. The large importance also means large mutual information, meaning that input variables with large importance play more important roles to make competitive units fire in more organized ways. Second, the optimal amount of information can be estimated. To determine the optimal value of the estimated information, we have proposed the ratio of the estimated information to the spread parameter σ

$$RE = \frac{EI(\sigma, \epsilon)}{\sigma}. \tag{15}$$

When the estimated information is larger, the number of important input variables is smaller. Because the number of important variables must be as small as possible, we must increase the estimated information as much as possible. In addition, to make the ratio large, we

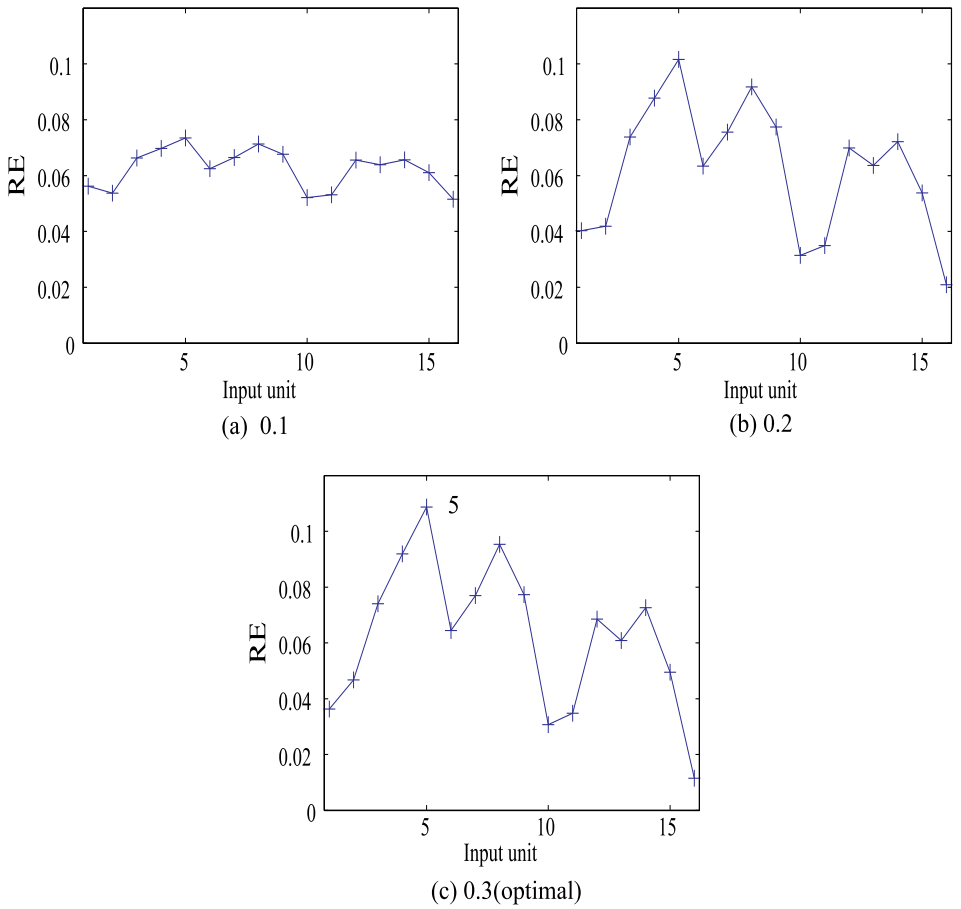


Fig. 25. Estimated importance for four different values of the parameter ϵ for the voting attitude problem.

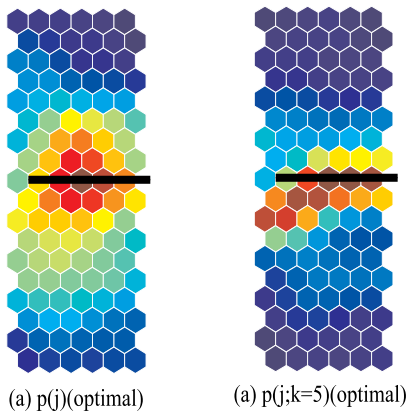


Fig. 26. The estimated probability $p(j)$ (a) and $p(j;k=5)$ for the voting attitude problem.

must decrease the parameter σ as much as possible. The parameter σ is related to the amount of mutual information between competitive units and input patterns. To increase

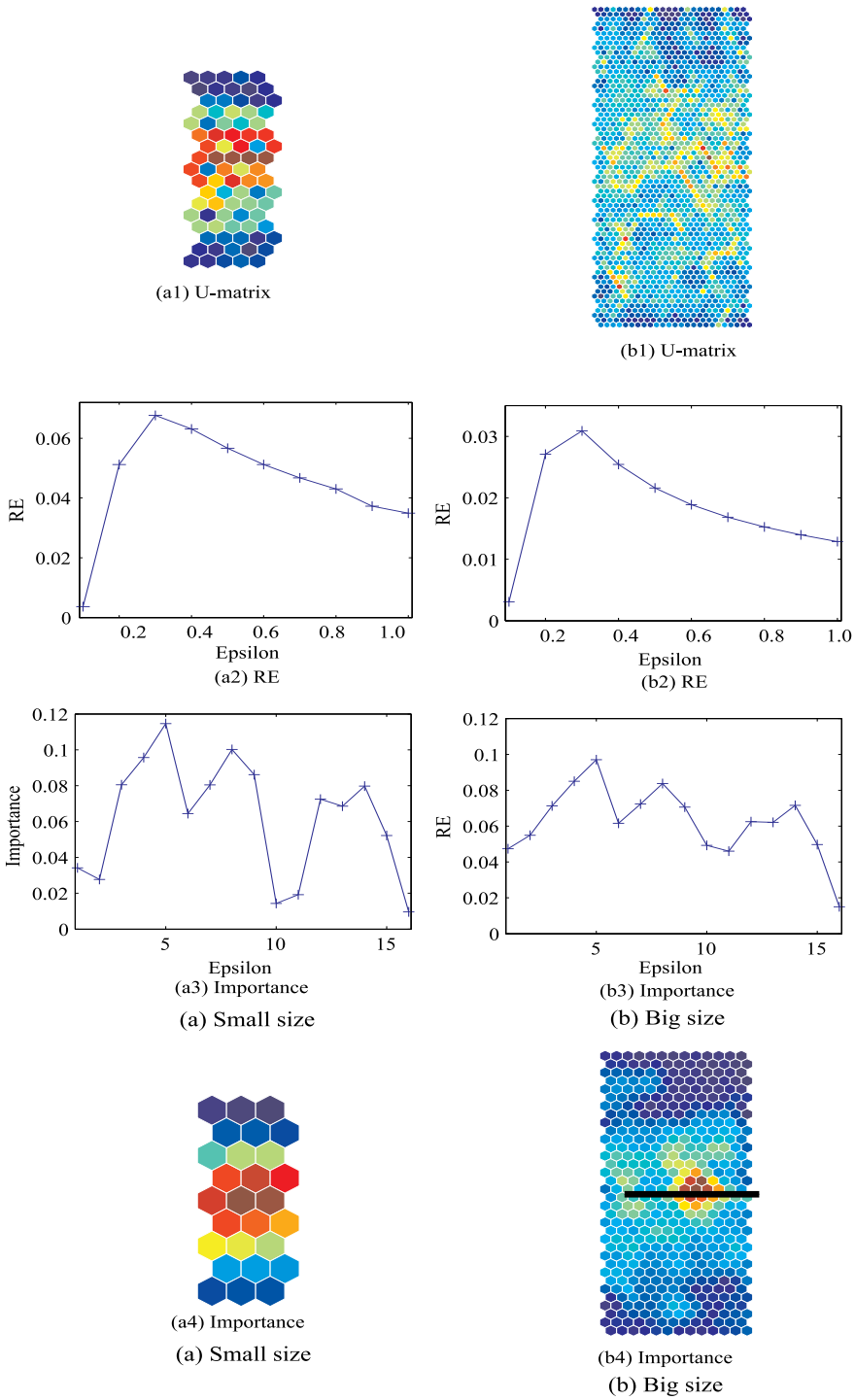


Fig. 27. U-matrices (1), the ratio RE (2), the values of importance (3) and $p(j)$ (4) for the small-sized (a) and large-sized network (b).

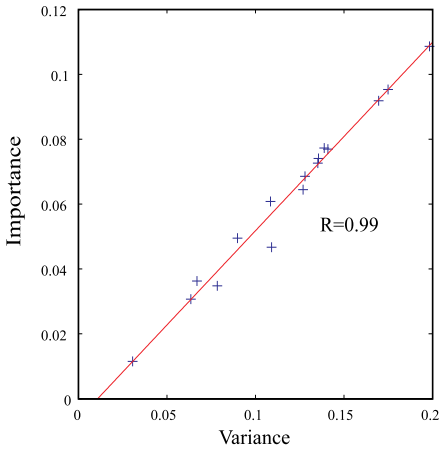


Fig. 28. Relations between importance and variance of connection weights for the voting attitude problem.

this mutual information, we must decrease the parameter σ as much as possible. Thus, the ratio is a reasonable criterion to obtain the optimal amount of estimated information. Four experimental results have shown clearly that the ratio can show peak values by which we can specify explicitly the values of two parameters. The validity of these optimal parameter values is endorsed by our obtaining the largest range of importance when the ratios have the maximum values. Third, the characteristics obtained by the optimal values of the parameters have been confirmed to be independent of the network size. To examine this property, we prepared larg- and small-sized networks for the two student survey data sets and the voting attitude problem. We observed that, even if the network size is changed, the optimal parameter values are exactly the same for the three problems. Though the values of the importance tend to be smaller when the network size is larger, the obtained values of the importance are completely the same independently of the network size, except for the range of the importance. This property of independency from the network size is important for the application to practical problems. Fourth, we have observed that the importance is closely correlated with the variance of connection weights. Figure 28 shows relations between importance and variance of connection weights for the voting attitude problem. As can be seen in the figure, the correlation coefficient between the two measures amounts to 0.99 for the problem. Generally speaking, the variance is a kind of measure to represent information in input patterns; this fact of a high correlation has supported the validity of our importance measure.

3.3.2 Limitation of the method

In our experiments, we have observed three problems or limitations of our method, namely, a problem of the parameter setting, information maximization for competitive units and the wrapper method. First, the parameters are precisely determined by examining the ratio of the estimated information to the parameter σ. However, to obtain the best parameter values, we must change the two parameters σ and ϵ extensively. Thus, this extensive search for the parameters may be a burden when we try to apply the method to practical and large-scaled problems. We think that, to reduce the computational complexity, we need to simplify the parameter setting as much as possible. Second, we have the problem of information

maximization for competitive units. As mentioned in the introduction section, because we have focused on the importance of input units, information in input units is more strongly maximized compared with information in competitive units. However, mutual information between competitive units and input patterns shows a kind of organization of competitive units. As this mutual information is more increased, more organized patterns of competitive units are generated. Because we focus upon information maximization in input units, we have paid restrained attention to the increase in this mutual information. Thus, we need to maximize mutual information in competitive units more strongly in addition to information maximization in input units. The third problem is closely related to the second one. Our method is a kind of wrapper method; we can use any learning method for learning, and then we use the information-theoretic method. In our method, we suppose two types of information, namely, mutual information between competitive units and input patterns. If it is possible to maximize two types information simultaneously, the final network is one with much information included in input units as well as competitive units. To realize this situation, we must train a network in learning, while increasing two types of information. Thus, we need an embedded system in which both learning and information maximization are simultaneously applied.

3.3.3 Possibility of the method

One of the main possibilities of our method can be summarized by two points, namely, its simplicity and the possibility of new learning. First, the importance is actually defined by focusing upon a specific input pattern. This means that the measure of information-theoretic importance can be applied to any elements or components of a network, such as connection weights, competitive units and so on. All we have to do is focus upon a specific element or component and compute mutual information between competitive units and input patterns. In particular, the applicability to the components in which several elements are combined with each other is one of the main possibilities or potentialities of our method. Second, our method opens up a new perspective for learning. In the present study, we have restricted ourselves to the detection of the importance of input variables. Now that the importance can be determined by the mutual information between competitive units and input patterns, the obtained information on the importance of input variables can be used to train networks. In that case, the learning can be done with due consideration to the importance of input variables.

4. Conclusion

In this chapter, we have proposed a new type of information-theoretic method to estimate the importance of input variables. This importance is estimated by mutual information between input patterns and competitive units, with attention paid to the specific input units. As this mutual information becomes larger, more organized competitive units are generated by the input units. Then, the information content of input variables is computed by using the importance. When this information is maximized, only one input variable plays an important role. Thus, we should increase this information as much as possible to obtain a smaller number of important input variables. To increase this information on input variables and mutual information between competitive units and input patterns, we have proposed the ratio RE of the information to the parameter ϵ to determine an optimal state. As this ratio is increased, the information on input variables is naturally increased and the corresponding mutual information between competitive units and input patterns is increased. We applied the

method to four problems, namely, a symmetric data, two data sets of actual of student surveys and the voting attitude problem. In all the problems, we have shown that, by maximizing the ratio, we can have the largest values of importance for easy interpretation. In addition, these values of the importance are independent of the network size. Finally, experimental results have confirmed that the importance of input variables is strictly correlated with the variance of connection weights. Though the parameter tuning requires an extensive search procedure to find an optimal state of information, these results certainly show that our information-theoretic method can be applied to many practical problems, because the importance can be determined based upon an explicit criterion and its meaning assured in terms of the variance of connection weights.

5. Acknowledgment

The author is very grateful to Kenta Aoyama and Mitali Das for their valuable comments.

6. References

Andrews, R., Diederich, J. & Tickle, A. B. (1993). Survey and critique of techniques for extracting rules from trained artificial neural networks, *Knowledge-based systems* 8(6): 373–389.

Barakat, N. & Diederich, J. (2005). Eclectic rule-extraction from support vector machines, *International Journal of Computational Intelligence* 2(1): 59–62.

Belue, L. M. & K. W. Bauer, J. (1995). Determining input features for multiplayer perceptrons, *Neurocomputing* 7: 111–121.

Garcez, A. S. d., Broda, K. & Gabbay, D. (2001). Symbolic knowledge extraction from trained neural networks: a sound approach, *Artificial Intelligence* 125: 155–207.

Gorman, R. P. & Sejnowski, T. J. (1988). Analysis of hidden units in a layered network trained to classify sonar targets, *Neural Networks* 1: 75–89.

Guyon, I. & Elisseeff, A. (2003). An introduction to variable and feature selection, *Journal of Machine Learning Research* 3: 1157–1182.

Kahramanli, H. & Allahverdi, N. (2009). Rule extraction from trained adaptive networks using artificial immune systems, *Expert Systems with Applications* 36: 1513–1522.

Kamimura, R. (2003a). Information theoretic competitive learning in self-adaptive multi-layered networks, *Connection Science* 13(4): 323–347.

Kamimura, R. (2003b). Information-theoretic competitive learning with inverse Euclidean distance output units, *Neural Processing Letters* 18: 163–184.

Kamimura, R. (2003c). Progressive feature extraction by greedy network-growing algorithm, *Complex Systems* 14(2): 127–153.

Kamimura, R. (2003d). Teacher-directed learning: information-theoretic competitive learning in supervised multi-layered networks, *Connection Science* 15: 117–140.

Kamimura, R. (2007). Information loss to extract distinctive features in competitive learning, *Proceedings of IEEE Conference on Systems, Man, and Cybernetics*, pp. 1217–1222.

Kamimura, R. (2008a). Conditional information and information loss for flexible feature extraction, *Proceedings of the international joint conference on neural networks(IJCNN2008)*, pp. 2047–2083.

Kamimura, R. (2008b). Feature detection and information loss in competitive learning, *Proceedings of the international conference on soft computing and intelligent systems and the international symposium on advanced intelligent systems(SCIS and ISIS2008)*,

pp. 1144–1148.
Kamimura, R. (2008c). Feature discovery by enhancement and relaxation of competitive units, *Intelligent data engineering and automated learning-IDEAL2008(LNCS)*, Vol. LNCS5326, Springer, pp. 148–155.
Kamimura, R. (2009). Enhancing and relaxing competitive units for feature discovery, *Neural Processing Letters* 30(1): 37–57.
Kamimura, R. & Kamimura, T. (2000). Structural information and linguistic rule extraction, *Proceedings of ICONIP-2000*, pp. 720–726.
Kamimura, R., Kamimura, T. & Uchida, O. (2001). Flexible feature discovery and structural information control, *Connection Science* 13(4): 323–347.
Kaski, S., Nikkila, J. & Kohonen, T. (1998). Methods for interpreting a self-organized map in data analysis, *Proceedings of European Symposium on Artificial Neural Networks*, Bruges, Belgium.
Kohonen, T. (1988). *Self-Organization and Associative Memory*, Springer-Verlag, New York.
Kohonen, T. (1995). *Self-Organizing Maps*, Springer-Verlag.
Mak, B. & Munakata, T. (2002). Rule extraction from expert heuristics: a comparative study of rough sets with neural network and ID3, *European journal of operational research* 136: 212–229.
Mao, I. & Jain, A. K. (1995). Artificial neural networks for feature extraction and multivariate data projection, *IEEE Transactions on Neural Networks* 6(2): 296–317.
Petersen, M., Talmoon, J. L., Hasman, A. & Ambergen, A. W. (1998). Assessing the importance of features for multi-layer perceptrons, *Neural Networks* 11: 623–635.
Polzlbauer, G., Dittenbach, M. & Rauber, A. (2006). Advanced visualization of self-organizing maps with vector fields, *Neural Networks* 19: 911–922.
Rumelhart, D. E., Hinton, G. E. & Williams, R. (1986). Learning internal representations by error progagation, *in* D. E. Rumelhart & G. E. H. et al. (eds), *Parallel Distributed Processing*, Vol. 1, MIT Press, Cambridge, pp. 318–362.
Steppe, J. M. & K. W. Bauer, J. (1997). Feature saliency measures, *Computers and Mathematics with Applications* 33(8): 109–126.
Tasdemir, K. & Merenyi, E. (2009). Exploiting data topology in visualizations and clustering of self-organizing maps, *IEEE Transactions on Neural Networks* 20(4): 549–562.
Thrun, S. (1995). Extracting rules from artificial neural networks with distributed representations, *Advances in Neural Processing Systems*.
Towell, G. G. & Shavlik, J. W. (1993). Extracting refined rules from knowledge-based neural networks, *Machine learning* 13: 71–101.
Tsukimoto, H. (2000). Extracting rules from trained neural networks, *IEEE Transactions on Neural Networks* 11(2): 377–389.
Ultsch, A. (2003). U*-matrix: a tool to visualize clusters in high dimensional data, *Technical Report 36*, Department of Computer Science, University of Marburg.
Ultsch, A. & Siemon, H. P. (1990). Kohonen self-organization feature maps for exploratory data analysis, *Proceedings of International Neural Network Conference*, Kulwer Academic Publisher, Dordrecht, pp. 305–308.
Vesanto, J. (1999). SOM-based data visualization methods, *Intelligent Data Analysis* 3: 111–126.

2

Privacy-Preserving Clustering on Distributed Databases: A Review and Some Contributions

Flavius L. Gorgônio and José Alfredo F. Costa
Federal University of Rio Grande do Norte
Brazil

1. Introduction

Clustering is the process of discovering groups within high-dimensional databases, based on similarities, with a minimal knowledge of their structure. Traditional clustering algorithms perform it over centralized databases, however, recent applications require datasets distributed among several sites. Therefore, in distributed database environments, all distributed data must be concentrated on a central site before applying traditional algorithms.

There is a series of limitations which hinder the utilization of traditional data mining techniques on distributed databases. The approach commonly taken, the gathering of all distributed databases in a central unit, followed by algorithm application, is strongly criticized, because in these cases, it is important to take into consideration some issues, namely: the possibility of existence of similar data with different names and formats, differences in data structures, and conflicts between one and another database (Zhang et al., 2003). Besides, the unification of all of the registers in a single database may take to the loss of meaningful information, once that statistically interesting values in a local context may be ignored when gathered to other ones in a larger volume.

On the other hand, integration of several database in a single location is not suggested when it is composed of very large databases. If a great organization has large disperse databases and needs to gather all the data in order to apply on them data mining algorithms, this process may demand great data transference, which may be slow and costly (Forman & Zhang, 2000). Moreover, any change that may occur in distributed data, for instance inclusion of new information or alteration of those already existing will have to be updated along with the central database. This requires a very complex data updating strategy, with overload of information transference in the system. Furthermore, in some domains such as medical and business areas whereas distributed databases occurs, transferring raw datasets among parties can be insecure because confidential information can be obtained, putting in risk privacy preserving and security requirements.

Due to all of these problems related to database integration, research for algorithms that perform data mining in a distributed way is not recent. In the end of the 90s, several researches about algorithms to effectuate distributed data mining started to appear, having been strengthened mainly by the rise of the distributed database managing systems and of the need for an analysis of such data in the way that they were dispersed (DeWitt & Gray, 1992; Souza, 1998). Currently, there is an increasing demand for methods with the ability to

process clustering securely that has motivated the development of algorithms to analyze each database separately and to combine the partial results to obtain a final result. An updated bibliography about the matter can be obtained in (Bhaduri et al., 2006).
This chapter presents a wide bibliographical review on privacy-preserving data clustering. Initially, different alternatives for data partitioning are discussed, as well as issues related to the utilization of classification and clustering ensembles. Further, some techniques of information merging used in literature to combine results that come from multiple clustering processes are analyzed. Then, are discussed several papers about security and privacy-preserving in distributed data clustering, highlighting the most widely used techniques, as well as their advantages and limitations. Finally, authors present an alternative approach to this problem based on the partSOM architecture and discuss about the confidentiality of the information that is analyzed through application of this approach in geographically distributed database cluster analysis.

2. Bibliographic review

Currently, a growing number of companies have strived to obtain a competitive advantage through participation in corporative organizations, as local productive arrangements, cooperatives networks and franchises. Insofar as these companies come together to overcome new challenges, their particular knowledge about the market needs to be shared among all of them. However, no company wants to share information about their customer and transact business with other companies and even competitors, because it is needed to maintain commercial confidentiality and due to local legislation matters.
Hence, a large number of studies in this research area, called privacy preserving data mining – where security and confidentiality of data must be maintained throughout the process – have been prompted by the need of sharing information about a particular business segment among several companies involved in this process, avoiding jeopardizing the privacy of its customers. A comprehensive review of these studies is presented below.

2.1 Data partitioning methods

There are two distinct situations that demand the need for effecting cluster analysis in a distributed way. The first occurs when the volume of data to be analyzed is relatively great, which demand a considerable computational effort, which sometimes is even unfeasible, to accomplish this task. The best alternative, then, is splitting data, cluster them in a distributed way and unify the results. The second occurs when data is naturally distributed among several geographically distributed units and the cost associated to its centralization is very high.
Certain current applications hold databases so large, that it is not possible to keep them integrally in the main memory, even using robust machines. Kantardzic (2002) presents three approaches to solve this problem:

i. Storing data in a secondary memory and clustering data subsets separately. Partial results are kept and, in a posterior stage, are gathered to cluster the whole set;
ii. Using an incremental clustering algorithm, in which every element is individually brought to the main memory and associated to one of the existing clusters or allocated in a new cluster. The results are kept and the element is discarded, in order to grant space to the other one;
iii. Using parallel implementation, in which several algorithms work simultaneously on stored data, increasing efficacy.

In cases in which the data set is unified and needs to be divided in subsets, due to its size, two approaches are normally used: horizontal and vertical partitioning (Figure 1). The first approach is more used and consists in horizontally splitting database, creating homogeneous data subsets, so that each algorithm operates on different records considering, however, the same set of attributes. Another approach is vertically dividing the database, creating heterogeneous data subsets; in this case, each algorithm operates on the same records, dealing, however, with different attributes.

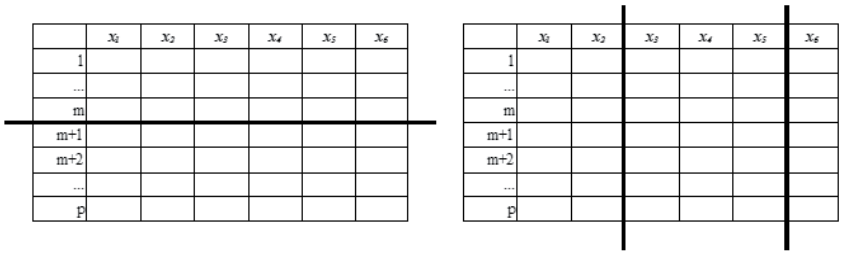


Fig. 1. Horizontal and vertical partitioning

In cases in which the data set is already partitioned, as in applications which possess distributed databases, besides the two mentioned approaches, it is still possible meet situations in which data is simultaneously disperse in both forms, denominated arbitrary data partitioning which is a generalization of the previous approaches (Jagannathan & Wright, 2005).

Both horizontal and vertical database partitioning are common in several areas of research, mainly in environments with distributed systems and/or databases, to which commercial application belongs. The way how data is disperse in a geographically distributed database environment depends on a series of factors which not always regard the task of clustering analysis as a priority inside the process. Operational needs of these systems may directly influence in the form of data distribution and data mining algorithms must be robust enough to cope with these limitations. For instance, in a distributed databases project, it is important to generate fragments which contain strongly related attributes, in order to guarantee a good performance in storage operations and information recovery (Son & Kin, 2004).

Recent studies on data partitioning technologies seek to meet this demand, particularly in situations in which incompatibilities between data distribution and queries carried out may affect system performance. When applied to distributed databases, vertical partitioning offers two great advantages which may influence system performance. First the frequency of queries necessary to access different data fragments may be reduced, once that it is possible to obtain necessary information with a smaller number of SQL queries. Second, the amount of recovered and transferred unnecessary information in a traditional query to memory may also be reduced (Son & Kin, 2004).

If, on one hand, data partition methods keeps focusing on queries performance, seeking for the more suitable number of partitions to make the recovery process of stored data quicker, the presence of redundant or strongly correlated variables in a process of cluster analysis with self-organizing maps, on the other hand, is not recommended (Kohonen, 2001). Therefore, in order to obtain better results in data analysis, the most recommended is geographically distributing data so that correlated variables stay in different units. Nonetheless in situations in which databases are already geographically distributed – not

being possible to alter their structure – and the existence of strongly correlated structures may impair results, it is possible to utilize statistical techniques, such as Principal Components Analysis (PCA) or Factor Analysis to select a more suitable subset of variables and reduce these problems.

2.2 Classification and cluster ensemble

Cluster ensembles may shortly be defined as a combination of two or more solutions come from application of different algorithms or variations of a same algorithm on a dataset, or even, on subsets thereof. The combination of several clustering algorithms has the objective of producing more consistent and reliable results than the utilization of individual algorithms does, which is why cluster ensembles have been proposed in several application which involve data clustering and classification.

The definition of cluster ensembles presented in the previous paragraph is deliberately generic, in order to include several possibilities of utilization of cluster algorithms and combination of results existing in the literature. In fact, Kuncheva (2004) suggests four approaches for classifying system development, which may be extended to cluster ensemble development:

i. Application of several instances of a same algorithm on the same database, changing the initialization parameters of the algorithm and combining its results;
ii. Application of different clustering algorithms on a same database, intending to analyze which algorithm obtains the best data clustering;
iii. Application of several instances of a same clustering algorithm on subsets of slightly different samples, obtained with or without reposition;
iv. Application of several instances of a same clustering algorithm on different subset of attributes.

Combining the result of several clustering methods, creating a cluster ensemble, appeared as a direct extension of the systems which use multiple classifiers (Kuncheva, 2004). Using of the multiple classifiers systems, based on the combination of the results of different classification algorithms, has been proposed as a method for developing high-performance classifiers systems with applications in the field of pattern recognition (Roli et al., 2001).

Theoretical and practical studies confirm that different kinds of data require different kinds of classifiers (Ho, 2000), which, at least theoretically, justifies ensembles utilization. Nevertheless, far from being consensual, the use of multiple classifier systems and cluster ensembles is questioned by several authors, both for requiring a greater computing effort, and for requiring the utilization of intricate mechanism of result combination (Kuncheva, 2003).

Roli et al. (2001) assert that the increasing interest in multiple classifier systems results from difficulties in deciding the best individual classifier for a specific problem. These authors analyze and compare six methods to project multiple classifier systems and conclude that, even though these methods have interesting characteristics, none of them is able to ensure an ideal project of a multiple classifier system.

Ho (2002) criticizes the multiple classifier systems, stating that, instead of concentrating efforts in seeking for the best set of attributes and the best classifier, the problem becomes seeking for the best set of classifiers and the best method of combining them. He also states that, later, the challenge becomes seeking for the best set of combining methods of results and the best way of using them. The focus of the problem is, then, forgotten and, more and more, the challenge becomes the usage of more complicated combining theories and schemes.

Strehl (2002) states as widely known the conception that the combination of multiple classifiers or multiple regression models may offer better results if compared to a single model. However, he alerts that there are no acknowledged effective approaches to combine clustering multiple non-hierarchical algorithms. In this work, the author proposes a solution to this problem using a framework to segmentation of consumers based on behavioural data.

In spite of all reported, both multiple classifier systems and cluster ensembles have been more and more used. Zhao et al. (2005) present a good review on the area, thus reporting several applications for classifiers ensembles based on neural networks, which include recognition of patterns, illness diagnostics and classification tasks. Oza & Tumer (2008) do the same in a more recent work, in which they present real applications, where using classifier ensembles has been obtaining a greater success in comparison to using individual classifiers, including remote sensoring, medicine and pattern recognition. Fern (2008) analyses how to combine several available solutions to create a more effective cluster ensemble, based on two critical factors in the performance of a cluster ensemble: quality and diversity of solutions.

Leisch (1998), one of the pioneers in the branch of cluster ensembles, introduced an algorithm named bagged clustering, which performs several instances of K-means algorithm, in the attempt of obtaining a certain stability in the results and combines partial results through a hierarchical partitioning method.

In another introductory work on distributed clustering analysis, Forman & Zhang (2000) present a tendency which parallelizes multiple algorithms based on centroids, like K-means and expectation maximization (EM) in order to obtain a greater efficacy in the process of data mining in multiple distributed databases. The authors reinforce the need for worrying about reducing the communication overload among the bases, reduce processing time and minimize the necessity for powerful machines with broad storage capacity.

Kargupta et al. (2001) highlight the absence of algorithms which effect clustering analysis in heterogeneous data sets using Principal Component Analysis (PCA) in a distributed way and present an algorithm denominated Collective Principal Component Analysis (CPCA) to analyze high dimension heterogeneous data clusters. The authors also discuss the effort of reducing the rate of data transference in a distributed data environment.

Haykin (2001) describes the neural networks as processors massively distributed in a parallel way, which suggests that the training of a cluster ensemble based on neural network may be done in a distributed way (Vrusias et al., 2007). Besides, there are, in literature, several researches striving to approach parallel neural network training, in particular, of self-organizing maps (Yang & Ahuja, 1999; Calvert & Guan, 2005; Vin et al., 2005).

This type of training generates innumerable challenges, once that, as a rule, neural network algorithms are non-deterministic and based on a set of initialization and training parameters. Thus, as neural networks normally are highly responsive to initialization parameters, choices done during the training process end up directly influencing the achieved results.

Some researches in this area exploit this particularity pertaining to neural networks to create ensembles based on the execution of a same algorithm with different initialization and training sets. In this approach, bootstrap aggregating, bagging and boosting are some of the techniques which have been used with some relative success in ensemble training, as described in (Breiman, 1996; Freud & Schapire, 1999; Frossyniotiset al., 2004; Vin et al., 2005). Even though such techniques have been demonstrating the existing variation of

probabilities and the benefits of these approaches, some problems became evident, which need to be considered while training ensembles concurrently with subsets of distinct inputs, such as computational cost and result fusion mechanisms.

The utilization of clusters of computers and computational grids has been frequently considered in performing distributed training of several types of neural networks, as multilayer perceptron networks and self-organizing maps (SOM), as well as radial base function networks (RBF) (Calvert & Guan, 2005). Hämäläinen (2002) presents a review on several parallel implementations utilizing self-organizing maps.

Neagoe & Ropot (2001) present as neural classifying model, denominated concurrent self-organizing maps (CSOM), which is composed of a collection of small SOM networks. CSOM model present some conceptual differences from tradition SOM model – the major is in the training algorithm, which is supervised. The number of SOM networks used in the model must be equal to the number of output space classes. To each individual SOM network, a specific training subset is used, so that the network is trained to have expertise in a certain output space class. Hence, in the end of the training stage, each SOM became expert on the class that it represents.

During the classifier utilization, the map which presents the lesser quantified error is declared winner and its index is the index of the class to which the pattern belongs. In tests performed with CSOM model, the authors consider three applications in which this model presents fair results: face recognition, speech recognition and multi-spectral satellite images (Neagoe & Ropot, 2002; Neagoe & Ropot, 2004).

Arroyave et al. (2002) present a parallel implementation of multiple SOM networks using a Beowulf cluster, with application on the organization of text files. In this approach, a huge self-organizing map is divided into several parts with the same size and distributed among the machines of the cluster. The training is also performed in a distributed way, so that every slave unit receives each of the input data from the master unit and returns to its own best match unit, which is shared with the other machines in a cooperative process.

Vrusias et al. (2007) propose an algorithm to train self-organizing maps, in a distributed way, by utilizing a computational grid. The authors propose a SOM cluster training architecture and methodology distributed along a computational grid, in which it is considered: the ideal number of maps in the ensemble, the impact of the different kinds of data used in the training and the most appropriate period for weight updating.

The training foresees periodical updates in map weight, in which the partial results of each units are sent to the master unit in the beginning of each training stage, and the latter is responsible for effecting the mean of received data and send them to the slaves units. Once that there is much integration among the parts along the training, time spent in this operation may be long, directly influencing in the map training time. Therefore, according to the authors, this approach only has results in dedicated clusters.

The authors performed a series of experiments and obtained important conclusions which can be extended to other SOM network parallel training algorithms:

i. If the latency time of the ensemble members the periodical weight adjusts and the synchrony time of the maps are very short, in comparison to the computational time of each training stage, the utilization of a SOM ensemble brigs about good results, regarding training time and accuracy;
ii. In the performed tests, the ideal number of maps in an ensemble was between 5 and 10 networks;

iii. The choice of the several utilized parameters in the training (learning and decrement rate) and the frequency which calculations of the map average are also factors of great importance in reducing mean square error;
iv. SOM ensemble presents quite superior results as the dimension of the data set increases.

Georgakis et al. (2005) propose the utilization of a self-organizing ensemble in attempt to increase performance in document organization and recovery. Several maps are simultaneously trained, with slightly different subsets. In posterior stage, maps are compared and the neurons of the ensemble members are lined to create the final map. The most similar neurons of each map are combined, through an arithmetic mean of their synaptic weights, to create a new neuron in the final map. During the training, uniformly distributed samples are taken from the data set to feed each of the members of the ensemble. The algorithm is used to partition a cluster document repository, according to its semantic contents. The performed experiments show that the performance of this algorithm is superior to the performance of traditional SOM, regarding to data recovery accuracy based on its semantic contents.

Cluster ensemble application on different attribute subsets has been analyzed mainly in image segmentation. Picture SOM or PicSOM in a hierarchical architecture in which several algorithms and methods can be applied jointly for image recovering based on contents (Laaksonen et al., 1999; Laaksonen et al., 2000; Laaksonen et al., 2002). Originally, PicSOM utilizes multiple instances of TS-SOM algorithm – which is composed by structured trees of self-organizing maps, hierarchically organized (Koikkalainen, 1994). Each TS-SOM is trained with a different set of characteristics, such as colour, texture or form.

PicSOM architecture is an example of SOM network combination, whose result is a solid system for image recovery based on content similarity. Georgakis & Li (2006) propose a PicSOM modification using a technique named bootstrapping during training stage. This technique divides randomly input space in a series of subsets which are used in the training stage of SOM. Then, the trained maps are combined into a single map to create the final result. According to the authors, this approach obtains more accurate results that original PicSOM.

Yu et al. (2007) propose an architecture to segment images based on an expectation maximization algorithm ensemble. This architecture starts extracting colour and texture information from the image, which are processed separately. A posterior stage combines the neighbouring regions individually segmented, taking into consideration information related to the position of pixels in an image. Jiang & Zhou (2004) present another proposal of SOM network ensemble usage to image segmentation, based only on information about colour and pixel position. The proposed approach combines partial results through a weighted voting scheme evaluated through mutual information index, which measures similitude among partitions.

Most of cluster analysis algorithms deals only with number data, even though there are some varieties of these algorithms specifically developed to handle with categorical data. Concerning databases with both kinds of values, some adjusts are necessary during the stage prior to processing, like, for instance, categorical data conversion in mutually exclusive binary data. Such a conversion elevates database dimensionality even more, once that it creates an additional column for each possible attribute value. Some alternative approaches for coding categorical variables into number variables are presented in the literature. Rosario et al. (2004) propose a method which analyzes how to determine order

and spacing among nominal variables and how to reduce the number of distinct values to be considered, based on Distance-Quantification-Classing approach.
He et al. (2005) analyze the influence of data types in the process of clustering and propose a different approach, effecting a division of the set of attributes into two subsets – one only with number attributes and another with only categorical attributes. Thereafter, they propose clustering of each of the subset in an isolated way, using algorithms suitable for each of the types. Eventually, each results of clustering process are to be combined into a new database, which is submitted, again, to a categorical data clustering algorithm.
Luo et al. (2007) propose an alternative method to data partitioning for generating ensemble training subsets, based in adding noise to original data. This method proposes utilizing artificial noise to produce variability in data during execution of clustering algorithms. The artificial data generated are an approximation of real data, in which are computed the mean and standard-deviation of the sample in order to generate data from Gaussian distribution found.
Recently, several works on the branch of cluster ensembles applied to bioinformatics, particularly to genic expression analysis. Silva (2006) investigates the utilization of cluster ensembles in genic expression analysis. Data is analyzed through three different cluster algorithms (K-means, EM and average linkage hierarchical clustering) and results are combined through different techniques, such as voting, relabeling and graphs. Results show that this approach obtains a superior result than the utilization of individual techniques, particularly when composite ensembles are used by several algorithms.
Faceli (2006) proposes an architecture for exploratory data analysis through clustering techniques. Such an architecture is composed by a multi-objective cluster ensemble, which executes several conceptually different algorithms with several parameter configurations, combines partitions resulting from this algorithm and selects partitions with the best results with different validation measures. Among the databases used for validation of the proposal, some genes expressions are also included.

2.3 Combining ensemble results

A problem which is inherent to cluster ensembles is partial results combining. Strehl (2002) describes efficiently this matter and presents three most common approaches to solve this problem under different points of view. The first approach consists of analyzing similitude among different partitions produced through utilization of similarity metrics among partitions. The second uses hyper-graphs to represent relationship among the objects and applies hyper-graph partitioning algorithms on them to find the clusters. In the third approach the elements of input set are labeled and, then, labels are combined to present a final result, normally through some voting system.
Strehl & Ghosh (2002) introduce the problem of combining multiple partitions of a set of objects into a single partition consolidated from obtained partial labels. In short, the objective of this approach is obtaining a set of labels which correspond to the result of each partition and, considering only partial results, combining them in order to obtain a consensual result, not taking into consideration previous characteristics about the objects which determined the partitions. In fact, this is the most popular way of result fusion among the three presented by Strehl (2002) for data clustering tasks and difficulties associated to its utilization have been investigated in several other works which approach this issue (Dimitriadou et al., 2001; Frossyniotis et al., 2004; Zhou & Tang, 2006; Tumer & Agogino, 2008).

Some SOM ensemble-based works introduce specific result combining techniques through map fusion techniques. In the approach proposed by Vrusias et al. (2007), a great self-organizing map is divided into small sub-maps and sent to the several units of a computational grid, to be trained in parallel. Each unit trains its own sub-map with a subset of different data. In this case, result fusion is done with base on means of individually trained maps. The mean values of the neurons are obtained through the arithmetic mean, in each dimension, for each SOM instance in the ensemble.
This calculation is made after a prefixed number of interactions in training stage. Once that each neural network is trained with its base on its respective dataset, this process tends to decrease as to accuracy in comparison to training a single network with the all data available, however more efficacy is obtained as to time spent in training. On the other hand the ensemble has a potential to generate better results than a single neural network once that a greater amount of training can be performed in the same time interval.
In another proposal, Georgakis et al. (2005) suggest a SOM ensemble simultaneously trained with slightly different data subsets and used to organize and recover documents. In this case, result fusion is also performed through an arithmetical mean of its synaptic weights, but combining the most similar neurons of each map in order to compose a new neuron of the final map. The difficulty of this proposal is maintaining the topology of partial maps in the final map. The same strategy is used in a later work for image recovery based on contents (Georgakis & Li, 2006).
Hore et al. (2006) describe some ways of results fusion based on label combining and show that these methods are not suitable for application on very large databases. Which is why, they present a proposal of cluster ensembles which extracts a set of centroids, labels these centroids and combines results to identify the clusters of the original dataset. Besides, the work includes an additional process to eliminate malformed clusters due to initialization or data distribution failures or to existing noises.

2.4 Security and privacy preserving data mining

Data security and privacy-preserving are among the primal factors which motivate creation and maintenance of distributed database (Chak-Man et al., 2004). Many organizations, then, maintain their databases geographically distributed, as a way to increase the security of their information; for if, by chance, one of their security policies fails, the intruders has access to only a part of the existing information.
The need for assure information confidentiality during a knowledge extraction process in databases is a very current area of research in scientific society (Kapoor et al., 2007). Researches involving data security and privacy-preserving in databases had an unexpected increase in the last years, caused by growing preoccupation of individuals in sharing their personal information via Internet, as well as the worry of business in assuring security of this information (Verykios et al., 2004).
It is known that combining several sources of data during a KDD process increases analysis process, even though it jeopardizes security and privacy-preserving of data involved in the process (Oliveira & Zaïane, 2007). Wherefore, data mining algorithms which operate in distributed way must take into consideration not only the way data is distributed among the units, in order to avoid unnecessary transferences, rather they must also ensure that transferred data is protected against occasional attempts of undue appropriation attempts. Inasmuch as digital repositories have become more and more susceptible to attacks and business and organizations all over the world have frequently been held responsible for

abuses, once that governments have been adopting more and more rigorous legislations pertaining to collected data privacy-preserving, these worries have been demanding new advances in the area of distributed data mining (Kapoor et al., 2006).

A potentially interesting market to distributed data mining is corporative organizations, composed of a significant number of businesses which work around one principal activity, such as local productive arrangements, business agglomerations, corporative networks, cooperatives and franchises. Simultaneous application of data mining algorithms on databases owned by several companies which act on the same branch allows obtaining more complete information and more accurate knowledge on this segment, augmenting the knowledge of the group about that area of business (Thomazi, 2006). Nonetheless, in spite of the obvious advantages of this approach, most of businesses participating in corporative organizations decide for analyzing on their individual databases. Security restrictions hinder sharing information from customers among partner companies in several countries and create a series of problems related to privacy-preserving, preventing companies from adopting this strategy.

Privacy-preserving cluster analysis rises as a solution to this problem, permitting that the parties to cooperate among them in knowledge extraction, preventing obligation of each of them of revealing their individual data to the others. This approach concentrates its efforts in algorithms which assure privacy and security to data involved in the process, mainly in applications in which security has fundamental importance, for instance, in medical and commercial applications (Berkhin, 2006; Silva, 2006).

Verykios et al. (2004) discuss the state of the art in data security and privacy, presenting the most common three techniques: the ones based on heuristic, which seek purposely to alter some database values, avoiding, however, losses in the process; the ones based on cryptography, which codify data in order to avoid access to information from other parties; and the ones based on data rebuilding, which use some technique in order to introduce perturbation in data, keeping existing relations among them. The authors present one more classification of the most common data mining algorithms according to the presented techniques.

The first references to security related problems in KDD problems arose even in the 90s (O'Leary, 1991; Piatetsky-Shapiro, 1995; Clifton & Marks, 1996). Nevertheless, the first researches with concrete results on privacy-preserving data mining area were published by Agrawal & Ramakrishnan (2000) and Lindell & Pinkas (2000). The former, based on a data rebuilding technique known as randomization, which introduces noise along to actual data, avoiding that data may be reconstituted, keeping, however, existing relations among them. The latter, using a cryptography technique named Secure Multi-party Computation (SMC), to classify data on horizontally distributed bases. SMC technique was proposed by Goldreich et al. (1987), from original idea proposed by Yao (1986).

Even though both approaches do not consider the need for data transference reduction among the units, several other works which followed are direct extensions of the these techniques. Agrawal & Aggarwal (2001) made continuity of the first work, adding more privacy to data and including the utilization of EM algorithm during data reconstruction. Following, Evfimievski et al. (2002) adapt the algorithm for association rule extraction on categorical attributes, adding noise to data and measuring the influence of these noises in final result. The technique used in the second work, based on SMC, was investigated in several other works. In spite of its efficacy in guaranteeing mined data security, its application in data mining tasks has ended up being inefficient, due to its complexity

(Clifton et al., 2002; Du & Atallah, 2001). More recently, some variations of this technique have been investigated, in the sense of reducing complexity.

Kantarcioglu & Vaidya (2002) criticize the security of randomization processes and the complexity of SMC algorithms, besides the need of all of these units for being connected during the process. As an alternative, they present an architecture which cheats these limitations in association rule extraction in distributed databases with information about clients. Nevertheless, this architecture requires the database to be entirely transferred to the central unit, which makes it unfeasible in many data mining applications. Vaidya & Clifton (2003) propose distributed implementation of K-means algorithm, based on SMC technique, for cluster analysis on vertically distributed databases. Lin et al. (2005) adapt the same idea for utilization along with EM algorithm. More recently, Vaidya et al. (2006) summarize the techniques most used in privacy-preserving data mining in prediction and description, both on horizontally and vertically partitioned databases.

Statistical techniques have been used to ensure data security and privacy-preserving in clustering tasks. Merugu & Ghosh (2003) present an architecture for distributed data clustering based on a technique named generative models, which causes data perturbation based on a statistical model, in order to guarantee privacy. Klusch et al. (2003) propose a distributed clustering algorithm based on local density estimation. This algorithm works in a distributed way, using an objective function to extract local partition density and combines the partial clusters sending information about clustering nucleus to the central unit. Data privacy and security are kept, once that only information about the clustering nuclei is shared.

Estivill-Castro (2004) proposes a method which combines a protocol of communication between two or more parts based on SMC and the utilization of K-medoids, a more robust variation of K-means, for clustering vertically partitioned data. Another approach based on the usage of K-medoids proposes the use of a cryptography technique denominated homomorphic ciphering to permit data sharing among the parties without jeopardizing security (Zhan, 2007). Later, Zhan (2008) expanded this technique to other data mining tasks. Jha et al. (2005) propose the utilization of K-means through two security protocols, polynomial evaluation and homomorphic evaluation.

The problem which arises when confidential information may be deduced from data made available to non-authorized users is known as the problem of inference in databases (Verykios et al., 2004; Farkas & Jajodia, 2002). Oliveira & Zaïane (2003) introduce a set of methods for data perturbation, based on geometrical transformations (translation, scale alteration) in p-dimensional space. Initially any attributes that may be used for individual identification of objects are eliminated. Then, the method effects several geometrical transformations on data, keeping statistical relations among them, but preventing them to be reconstructed.

Later, Oliveira & Zaïane (2004) propose improvement in the method of transformation based on geometric rotation in order to protect attribute values while these are shared in a clustering process. The main advantage of the proposed method is that it is independent from clustering algorithms. More recently, the authors combine results of previous studies in a new method for privacy-preserving cluster analysis, denominated Dimensionality Reduction-Based Transformation (DRBT), with applications on the commercial area (Oliveira & Zaïane, 2007).

Jagannathan & Wright (2005) introduce the concept of arbitrary data partitioning, which is the generalization of horizontal and vertical partitioning and present a method for data

clustering tasks with K-means algorithm on arbitrarily partitioned data bases. This method utilizes a cryptography-based protocol to guarantee data privacy. Jagannathan et al. (2006) suggest a safer variant of K-means algorithm previously proposed, however for clustering on horizontally distributed databases.

İnan et al. (2006) and İnan et al. (2007) approach privacy-preserving clustering analysis through an algorithm which permits to build a dissimilarity matrix among objects on horizontally distributed databases, through SMC to ensure security. The algorithm works suitably with numerical and categorical attributes and the built dissimilarity matrix may be applied to other data mining tasks. Kapoor et al. (2007) present an algorithm named PRIPSEP (PRIvacy Preserving SEquential Patterns), based on SMC technique, which permits mining sequential patterns on distributed database, while it maintains the privacy-preserving of the individual.

In some more recent works, Vaidya (2008) presents and discusses several data mining methods which operate in a distributed way on vertically partitioned databases, while Kantarcioglu (2008) does the same to methods which operate in a distributed way on horizontally partitioned databases. Fung et al. (2008) propose an architecture for data clustering analysis which convert a cluster analysis process in a classification activity. The proposed algorithm carried out data clustering and associates data in a set of classes. Then, it codifies actual data through labels and transmits codified data as well as respective classes to other units, thus preserving privacy of data involved in the process.

3. The partSOM architecture clustering process

This section presents a cluster ensemble methodology for privacy preserving clustering in distributed databases, using traditional and well known algorithms, such as self-organizing maps and K-means. The proposed methodology combines a clustering architecture, the partSOM architecture (Gorgônio & Costa, 2010), with principles of vector quantization, building a cluster ensemble model that can be used to cluster analysis in distributed environments composed by a set of partner companies involved in this process, avoiding jeopardizing the privacy of their customers.

The main idea of this process is focused on omission of real information about customers, changing a set of real individuals for one (or more) representative (and fictional) individual with similar statistical characteristics of the real individuals. This strategy, based on vector quantization principles, enables that a group of individuals with similar characteristics to be able to be represented by a single individual (vector) corresponding to that group. As illustrated in Figure 2, the vectors $\{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_7, \mathbf{x}_8\}$ can be represented by $\mathbf{w}_1$ vector and $\{\mathbf{x}_2, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_9\}$ can be represented by $\mathbf{w}_2$ vector. This strategy is used to reduce the amount of space required to store or transmit a dataset and has been widely used by clustering tasks and data compression of signals, particularly voice and image.

The partSOM architecture presents a strategy to carry out cluster analysis over distributed databases using self-organizing maps and K-means algorithms. This process is separated in two stages: initially, data are analyzed locally, in each distributed unit. In a second stage, a central unit receives partial results and combines them into an overall result.

The partSOM algorithm, embedded in partSOM architecture, consists of six steps and is presented as it follows. An overview of the complete architecture is showed in Figure 3.

1. A traditional clustering algorithm is applied in each local unit, obtaining a reference vector, known as the codebook, from each local data subset;

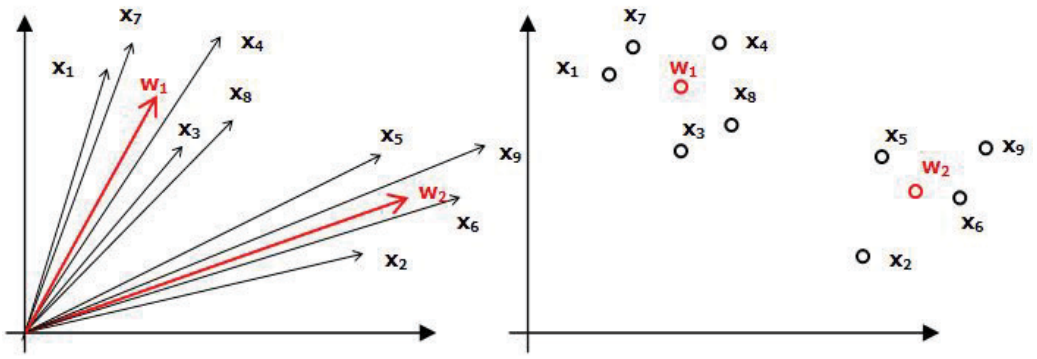


Fig. 2. Example of a vector quantization process in a bidimensional plan

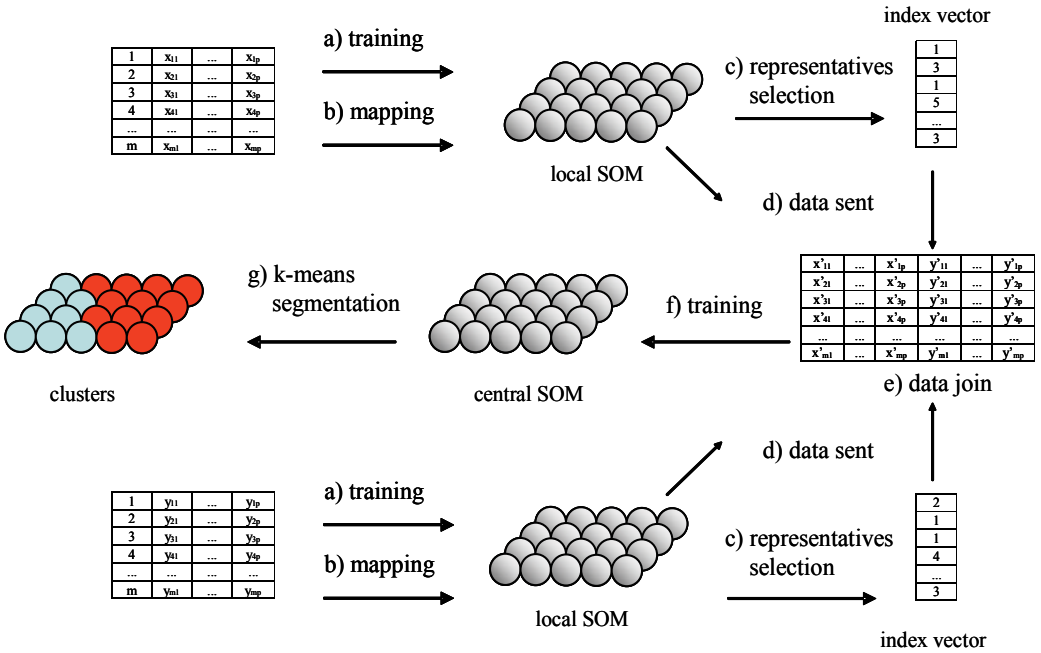


Fig. 3. An overview of the partSOM architecture with SOM and K-means algorithms

2. Each input data is compared with codebook issues and the index corresponding to the most similar vector present in the codebook is stored in an index vector. So, a data index vector is created based on representative objects instead of original objects;
3. Each remote unit sends the codebook and the index vector to the central unit, which will conduct the unification of all partial results;
4. The central unit is responsible for receiving index vectors and codebooks from each local unit and combining partial results and building a whole database. In this process, index vector issues are substituted by the similar issues in the codebook;
5. The clustering algorithm is applied on the whole database obtained in previous step, to identify existing clusters in the collective database;

6. A segmentation algorithm is applied on results obtained after the final cluster process, in order to improve the quality of the visualization results.

Despite the difference between the original and the remounted database, which are slightly different, the topology and statistical characteristics from original data is maintained, because representative objects in the index vector are very similar to the original data, as shown in several experiments (Gorgônio & Costa, 2008; Gorgônio & Costa, 2010). As a matter of fact, this is an important characteristic of the partSOM architecture, since results obtained with this architecture can be generalized as being equivalent to the clustering process of the entire original databases.

The architecture presented was developed focusing on geographically distributed databases, independently of criteria used in partitioning. Wherefore, a solution which is stable in any form of partitioning has been required, whether it is horizontal vertical or arbitrary, even though additional techniques may be used to better its performance in specific domains.

4. Some contributions to partSOM clustering process

This section presents some contributions to increase security and privacy preserving in a clustering process using the partSOM architecture. First of all, it is proposed a data pre-processing stage, which are removed all information that could be used to identify an individual. Following, it is proposed a pruning algorithm to reduce the amount of data transferred between the local and central units. Finally, it is proposes the use of a covariance matrix from each local data unit to reduce losses during the process of vector quantization.

4.1 The pre-processing stage

In real world applications, raw data usually are named *dirty data*, because they can contain errors, missing values, redundant information or are incomplete and inconsistent. So, most of data mining process needs a pre-processing stage that objectives to carry out tasks such as data cleaning, data integration and transformation, data reduction, although this important step is sometimes neglected in data mining process.

Conventionality, a relational database is a set of two-dimensional tables interrelated by one or more attributes. Each table is a two-dimensional structure of rows and columns where each row represents a record from the database and each column represents an attribute associated with that record. Figure 4 suggests a sample of a typical table in a database.

After pre-processing stage, data are usually arranged in single table known as data matrix, which must satisfy the requirements of the chosen algorithm. The data matrix **D** is formed by a set of n vectors, where each vector represents an element of the input set. Each vector has p components, which correspond to the set of attributes that identify it. A data matrix example, related to the previous presented table in Figure 4, is shown in Figure 5.

In this example, some attributes were removed, others were transformed and the whole dataset was normalized. As discussed in literature (Hore et al., 2006), this stage contributes to privacy and security maintenance of data and information stored in database, because real data are replaced by a set of representatives with same statistical distribution of original data. Thus, since only codebook and index vector are sent to the central unit and no real information is transferred, the security is maintained.

#	Name	Sex	Age	Wage	Civil State	Children	...	State
1	A. Araújo	M	39	2.300,00	Married	3	...	RN
2	Q. Queiroz	F	82	1.350,80	Widowed	2	...	PB
3	W. Wang	M	21	720,50	Single	1	...	CE
4	E. Eudes	F	18	1.420,00	Single	0	...	SP
5	S. Silva	M	16	450,00	Single	0	...	RN
6	G. Gomes	M	42	32.827,52	Married	2	...	DF
7	K. Key	F	38	410,50	Divorced	1	...	SE
...	...	...	...	...	...	...	...	...
N	M. Mendes	M	21	3.500,00	Married	4	...	BA

Fig. 4. Sample of a typical table in a database

0,72457	-0,72457	0,20077	-0,27575	-0,72457	...	-0,35355
-1,20760	1,20760	2,17410	-0,36094	-0,72457	...	-0,35355
0,72457	-0,72457	-0,62526	-0,41751	1,20760	...	-0,35355
-1,20760	1,20760	-0,76294	-0,35473	1,20760	...	2,47490
...	...	...	...	...	...	...
0,72457	-0,72457	-0,62526	-0,16805	-0,72457	...	-0,35355

Fig. 5. Data matrix sample obtained after pre-processing stage

4.2 The pruning algorithm

In terms of partSOM architecture, the most suitable algorithm during the initial codification stage in the local units is the self-organizing maps (Kohonen, 2001). In this case, the codebook may contain a few entries with little or no representation in the input set, known as dead neurons. These elements occur with some frequency in clustering processes using the SOM, what has been cited in the literature (Kamimura, 2003). Although inactive neurons can help to maintain the input data topology when they are projected on the map, these units can be discarded without impairment in a process of vector quantization using SOM, because such elements are not referenced in data reassembly stage.

In terms of K-means algorithm, codebook elements with little representation may correspond to outliers or noise in the input data and, eventually, these elements can be discarded from representatives set without great impairment to the maintenance of the statistical distribution of data. So, in both cases, it is possible to include a pruning algorithm in a stage before the transfer of data to the central unit, to reduce the size of the codebook and avoiding moving items that are not used (or are not relevant) in data reconstruction. The procedure for reducing the codebook is performed by a pruning algorithm (Figure 6), which will be detailed below.

The pruning algorithm receives the input dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, the trained codebook $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_k\}$, the set of representatives $\mathbf{R}$ and an integer value θ, which corresponds to the representation threshold required for each element. Then, the algorithm searches for elements whose representation is less than or equal to the threshold and eliminates them from the codebook. Finally, the representative choice algorithm is called again to reselect the representatives of each input dataset.

Importantly, the pruning algorithm is an optional step, whose objective is to reduce the amount of data transferred between the remote units and central unit. In the particular case

in which the threshold value is zero, only the inactive neurons are eliminated without any change in the outcome.

```
                              Pruning Algorithm

Input:  input dataset (X); original codebook (W);
        representatives set (R); threshold (θ)
Output: modified codebook (W');
        modified representatives set (R')

procedure pruning(X,W,R,θ)
for each w_j ∈ W
        cont = 0
        for each r_i ∈ R
                if (R[i] = j)
                        cont = cont + 1
                endif
        endfor
        if (cont <= θ)
                W' = remove(W,j)
        endif
endfor
R' = choose_representative(X,W')
return(W',R')
```

Fig. 6. The pruning procedure algorithm

4.3 The covariance matrix

The first step in partSOM architecture uses a vector quantization process to effect a compression in the input data and thus reduce the amount of data transferred to the central unit. As in any process of data compression, there are losses associated with vector quantization and, possibly some of the information existing in the input data is discarded during the first stage of the algorithm.

However, as described in Gorgônio (2010), a vector quantization process approximates a probability density function of the input set by a finite set of reference vectors. Thus, if the set of reference vectors chosen to represent the input data is representative enough to capture the statistical distribution of data in the input space, the close relations between the input elements will be maintained. Thus, even if the vector quantization process holds losses, these losses tend to be minimized with proper choice of a good set of representatives.

An alternative to minimize the losses occurring in the process of vector quantization is the use of additional statistical information contained in the original sample, so that the reconstructed data are as similar as possible to the input data. The covariance matrix of a set of data allows extracting the variance and correlation between the samples, and an efficient solution to create random samples containing the same statistical characteristics of the original sample.

Thus, if the covariance matrices of each cluster are drawn in remote units and sent along with the codebooks, so that each centroid can carry information about the variance of the data that it represents, and this information could be used to generate samples with a statistical distribution even more similar to the original dataset, helping to reduce losses associated with the process of vector quantization.

5. Conclusion

This chapter discussed the utilization of cluster ensembles in data clustering and classification tasks. Matters related to existence of geographically distributed databases and mechanisms used for data partitioning were analyzed. It was also presented a wide review on algorithms and strategies used in data mining, mainly in clustering tasks. Following, matters related to distributed data clustering security and privacy were addressed. Eventually, some information fusion techniques used to combine results come from multiple clustering solutions were cited in reviewed works.

The partSOM architecture was presented as a proposal for performing cluster analysis on geographically distributed databases, such as discussed in previous works. However, this study focused specifically on issues related to security and privacy preserving in distributed databases clustering. The main contribution of this work was a bibliographic review about the theme and a discussion about some techniques that can be used in a privacy preserving distributed databases clustering process, including:

i. A data pre-processing stage, which objectives to remove all information that could be used to identify an individual;
ii. A pruning algorithm to reduce the amount of data transferred between the local and central units;
iii. The use of a covariance matrix from each local data unit to reduce losses during the process of vector quantization.

Future research directions will be focused on extent the partSOM architecture, including use of others privacy-preserving strategies. Furthermore, it is necessary to apply and to evaluate this model in real world applications.

6. Acknowledgment

This work was supported by Federal University of Rio Grande do Norte. Flavius Gorgônio (flavius@ufrnet.br) works at Laboratory of Business Applied Computational Intelligence, Department of Exact and Applied Sciences, Caicó, RN, Brazil. José Alfredo F. Costa (alfredo@dee.ufrn.br) works at Laboratory of Adapting Systems, Department of Electrical Engineering, Natal, RN, Brazil.

7. References

Agrawal, R. & Srikant, R. (2000). Privacy-preserving data mining, *ACM SIGMOD Record*, ACM Press, Vol.29, No.2, (June, 2000), pp. 439-450

Agrawal, D. & Aggarwal, C. (2001), On the design and quantification of privacy preserving data mining algorithms, *Proceedings of the Symposium on Principles of Database Systems*, pp. 247-255, Santa Barbara, May, 2001

Arroyave, G.; Lobo, O. & Marín, A. (2002). A parallel implementation of the SOM algorithm for visualizing textual documents in a 2D plane, *Encuentro de Investigación sobre Tecnologías de Información Aplicadas a la Solución de Problemas*, Medellín, Colombia

Berkhin, P. (2006). A survey of clustering data mining techniques, In: *Grouping multidimensional data: recent advances in clustering*, J. Kogan; M. Teboulle & C. Nicholas (Eds.), pp. 25–72, Springer-Verlag, Heidelberg

Bhaduri, K.; Das, K.; Liu, K. & Kargupta, H. (November 2010) Privacy Preserving Distributed Data Mining Bibliography, In: *Distributed Data Mining Bibliography*, 03.11.2010, Available from http://www.cs.umbc.edu/~hillol/DDMBIB

Breiman, L. (1996). Bagging predictors, *Machine Learning*, Vol.24, No.2, pp. 123-140

Calvert, D. & Guan, J. (2005). Distributed artificial neural network architectures, *Proceedings of the 19th Int. Symposium on High Performance Computing Systems and Applications*, pp. 2-10

Chak-Man, L.; Xiao-Feng, Z. & Cheung, W. (2004). Mining local data sources for learning global cluster models, *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, Vol.20, No.24, pp. 748-751, September, 2004

Clifton, C. & Marks, D. (1996). Security and privacy implications of data mining, *Proceedings of the ACM SIGMOD Workshop on Data Mining and Knowledge Discovery*, pp.15-19, Montreal, Canada, June, 1996

Clifton, C.; Kantarcioglu, M.; Vaidya, J.; Lin, X. & Zhu, M. (2002). Tools for privacy preserving distributed data mining, *SIGKDD Explorations*, Vol.4, No.2, (December, 2002), pp. 28-34

DeWitt, D. & Gray, J. (1992). Parallel database systems: the future of high performance database processing. *Communications of the ACM*, Vol.36, No.6, (June, 1992), pp. 85-98

Dimitriadou, E.; Weingessel, A. & Hornik, K. (2001). Voting-merging: an ensemble method for clustering, *Proceedings of the Int. Conf. on Artificial Neural Networks*, LNCS, Vol.2130, pp. 217-224, London: Springer-Verlag

Du, W. & Atallah, M. (2001). Secure multi-party computation problems and their applications: A review and open problems, *New Security Paradigms Workshop*, pp. 11-20, Cloudcroft, New Mexico, September, 2001

Evfimievski, A.; Srikant, R.; Agrawal, R. & Gehrke, J. (2002). Privacy preserving mining of association rules, *Proceedings of the 8th International Conference on Knowledge Discovery in Databases and Data Mining*, Canada, pp. 217-228, July, 2002

Estivill-Castro, V. (2004). Private representative-based clustering for vertically partitioned data, *Proceedings of the Fifth Mexican International Conference in Computer Science*, (September, 2004), pp. 160-167

Faceli, K. (2006). *Um framework para análise de agrupamento baseado na combinação multi-objetivo de algoritmos de agrupamento*, PhD Thesis, Instituto de Ciências Matemáticas e de Computação (ICMC), Universidade de São Paulo, São Paulo, Brazil

Farkas, C. & Jajodia, S. (2002). The inference problem: a survey, *ACM SIGKDD Explorations Newsletters*, Vol.4, No.2, (December, 2002), pp. 6-11

Fern, X. & Lin, W. (2008). Cluster Ensemble Selection, *Proceedings of the 2008 SIAM Int. Conf. on Data Mining*, Atlanta, Georgia, April 24-26, 2008

Forman, G. & Zhang, B. (2000). Distributed data clustering can be efficient and exact. *ACM SIGKDD Explorations Newsletter*, Vol.2, No.2, (December, 2000), pp. 34-38

Freud, Y. & Schapire, R. (1999). A short introduction to boosting, *Journal of Japanese Society for AI*, Vol.14, No.5, pp. 771-780

Frossyniotis, D.; Likas, A. & Stafylopatis, A. (2004). A clustering method based on boosting, *Pattern Recognition Letters*, Vol.25, pp. 641-654

Fung, B.; Wang, K.; Wang, L. & Debbabi, M. (2008). A framework for privacy-preserving cluster analysis, *Proceedings of the IEEE International Conference on Intelligence and Security Informatics*, (June, 2008), pp. 46-51

Georgakis, A.; Li, H. & Gordan, M. (2005). An ensemble of SOM networks for document organization and retrieval, *Proceedings of the Int. Conf. on Adaptive Knowledge Representation and Reasoning*, pp. 141-147, (June, 2005), Espoo, Finland

Georgakis, A. and Li, H. (2006). Content based image retrieval using a bootstrapped SOM network, LNCS, Vol.3972, pp. 595-601, London: Springer-Verlag

Goldreich, O.; Micali, S. & Wigderson, A. (1987). How to play any mental game – a completeness theorem for protocols with honest majority, *Proceedings of the 19th ACM Symposium on the Theory of Computing*, pp. 218-222

Gorgônio, F. & Costa, J. (2008) Parallel self-organizing maps with application in clustering distributed data. *Proceedings of the International Joint Conference on Neural Networks*, Vol.1, (June, 2008), pp. 420, Hong-Kong

Gorgônio, F. & Costa, J. (2010) PartSOM: PartSOM: A Framework for Distributed Data Clustering Using SOM and K-Means. In: Matsopoulos, G. (ed.), *Self-Organizing Maps*, InTech Education and Publishing, Vienna, Austria

Hämäläinen, T. (2002). Parallel implementation of self-organizing maps, In: *Self-Organizing Neural Networks: Recent Advances and Applications*, U. Seiffert & L. Jain (Eds.), Vol.78, pp. 245-278, New York: Springer-Verlag

Haykin, S. (2001). *Redes neurais: princípios e prática*, 2ª ed., Porto Alegre: Bookman

He, Z.; Xu, X. & Deng, S. (2005), Clustering mixed numeric and categorical data: a cluster ensemble approach, Technical report, 07.06.2010, Available from http://aps.arxiv.org/ftp/cs/papers/0509/0509011.pdf

Ho, T. (2000). Complexity of classification problems and comparative advantages of combined classifiers. *Proceedings of the 1st International Workshop on Multiple Classifier Systems*, LNCS, Vol.1857, pp. 97-106, London: Springer-Verlag

Hore, P.; Hall, L. and Goldgof, D. (2006). A cluster ensemble framework for large data sets, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vol.4, pp. 3342-3347, October, 2006

İnan, A.; Saygın, Y.; Savaş, E.; Hintoğlu, A. & Levi, A. (2006). Privacy preserving clustering on horizontally partitioned data, *Proceedings of the 22nd Int. Conf. on Data Engineering Workshops*, pp. 95-103

İnan, A.; Kaya, S.; Saygın, Y.; Savaş, E.; Hintoğlu, A. & Levi, A. (2007). Privacy preserving clustering on horizontally partitioned data, *Data & Knowledge Engineering*, Vol.63, No.3, (December, 2007), pp. 646-666

Jagannathan, G. & Wright, R. (2005). Privacy-preserving distributed k-means clustering over arbitrarily partitioned data, *Proceedings of the 11th ACM SIGKDD Int. Conf. on Knowledge Discovery in Data Mining*, pp. 593-599

Jagannathan, G.; Pillaipakkamnatt, K. & Wright, R. (2006). A new privacy-preserving distributed k-clustering algorithm, *Proceedings of the 2006 SIAM International Conference on Data Mining*, pp. 492-496

Jha, S.; Kruger, L. & McDaniel, P. (2005). Privacy Preserving Clustering, *Proceedings of the 10th European Symposium on Research in Computer Security*, pp. 397-417

Jiang, Y. & Zhou, Z. (2004). SOM ensemble-based image segmentation, *Neural Processing Letters*, Vol.20, No.3, (November, 2004), pp. 171-178

Kantarcioglu, M. & Vaidya, J. (2002). An architecture for privacy-preserving mining of client information, In: *ACM International Conference Proceeding Series*, C. Clifton & V. Estivill-Castro (Eds), Vol.144, pp. 37-42, Australian Computer Society, Darlinghurst

Kantarcioglu, M. (2008). A survey of privacy-preserving methods across horizontally partitioned data, In: *Privacy-preserving data mining*, C. Aggarwal & P. Yu, pp. 313-336, Springer

Kantardzic, M. (2002). *Data mining: concepts, models, methods, and algorithms*, Wiley-IEEE Press

Kapoor, V.; Poncelet, P.; Trousset, F. & Teisseire, M. (2006). Privacy preserving sequential pattern mining in distributed databases, *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, New York, NY, pp. 758-767

Kapoor, V.; Poncelet, P.; Trousset, F. & Teisseire, M. (2007). Préservation de la vie privée: recherche de motifs séquentiels dans des bases de données distribuées. *Revue Ingénierie des Systèmes d'Information*, Vol.12, No.1, (Décembre, 2007), pp. 85-107

Kargupta, H.; Huang, W.; Sivakumar, K. & Johnson, E. (2001). Distributed clustering using collective principal component analysis, *Knowledge and Information Systems*, Vol.3, No.4, pp. 422-448

Kamimura, R. (2003). Competitive learning by information maximization: eliminating dead neurons in competitive learning, *Proceedings of the Joint International Conference ICANN/ICONIP*, LNCS, Vol.2714, pp. 99-106, Springer, Berlin, German

Klusch, M.; Lodi, S. & Moro, G. (2003). Distributed clustering based on sampling local density estimates, *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pp. 485-490

Kohonen, T. (2001). *Self-organizing maps*, 3rd edition, Berlin: Springer

Koikkalainen, P. (1994). Progress with the tree-structured self-organizing map, *Proceedings of the 11th European Conference on Artificial Intelligence*, New York: Wiley

Kuncheva, L. (2003). That elusive diversity in classifier ensembles. *Proceedings of the 1st Iberian. Conference on Pattern Recognition and Image Analysis*, LNCS, Vol.2652, pp. 1126-1138, London: Springer-Verlag

Kuncheva, L. (2004). *Combining pattern classifiers: methods and algorithms*, New Jersey: John Wiley & Sons

Laaksonen, J.; Koskela, M. & Oja, E. (1999). PicSOM: self-organizing maps for content-based image retrieval, *Proceedings of the 1999 Int. Joint Conf. on Neural Networks*, Vol.4, pp. 2470-2473

Laaksonen, J.; Koskela, M.; Laakso, S. & Oja, E. (2000). PicSOM – content-based image retrieval with self-organizing maps, *Pattern Recognition Letters*, Vol.21, No.13-14, (December, 2000), pp. 1199-1207

Laaksonen, J.; Koskela, M. & Oja, E. (2002). PicSOM – Self-organizing image retrieval with MPEG-7 content descriptors, *IEEE Transactions on Neural Networks*, Vol.13, No.4, (July, 2002), pp. 841-853

Leisch, F. (1998). *Ensemble methods for neural clustering and classification*. PhD Thesis, Institut für Statistik, Wahrscheinlichkeitstheorie und Versicherungsmathematik, Teschnische Universität Wien, Austria

Lin, X.; Clifton, C. & Zhu, M. (2005). Privacy-preserving clustering with distributed EM mixture modeling, *Knowledge Information Systems*, Vol.8, No.1, (July, 2005), pp. 68-81

Lindell, Y. & Pinkas, B. (2000). Privacy preserving data mining, *Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology*, pp. 36-54, August, 2000

Luo, H-L.; Xie, X-B. & Li, K-S. (2007). A new method for constructing clustering ensembles, *Proceedings of the Int. Conf. on Wavelet Analysis and Pattern Recognition*, Vol.2, pp.874-878, November 2-4, 2007

Merugu, S. & Ghosh, J. (2003). Privacy-preserving distributed clustering using generative models, *Proceedings of the 3rd IEEE International Conference on Data Mining*, pp. 211-218

Neagoe, V-E. & Ropot, A-D. (2001). Concurrent self-organizing maps for automatic face recognition, *Proceedings of the 29th International Conference of the Romanian Technical Military Academy*, pp. 35-40, Bucharest, Romania, November, 2001

Neagoe, V-E. & Ropot, A-D. (2002). Concurrent Self-organizing maps for pattern classification, *Proceedings of 1st IEEE Int. Conf. on Cognitive Informatics*, pp. 304

Neagoe, V-E. & Ropot, A-D. (2004). Concurrent self-organizing maps – a powerful artificial neural tool for biometric technology, *Proceedings of IEEE World Automation Congress*, Vol.17, Seville

O'Leary, D. (1991). Knowledge discovery as a threat to database security, In: *Knowledge discovery in databases*, G. Piatetsky-Shapiro & W. Frawley (Eds.), pp. 507-516, AAAI/MIT Press, Menlo Park

Oliveira, S. & Zaïane, O. (2003), Privacy preserving clustering by data transformation, *Proceedings of the 18th Brazilian Symposium on Databases*, pp. 304-318, Manaus, Brasil

Oliveira, S. & Zaïane, O. (2004). Privacy preservation when sharing data for clustering, *Proceedings of the Int. Workshop on Secure Data Management in a Connected World*, Vol.1, pp. 67-82, Toronto, Canada

Oliveira, S. & Zaïane, O. (2007). A privacy-preserving clustering approach toward secure and effective data analysis for business collaboration. *Computers & Security*, Vol.26, pp. 81-93

Oza, N. & Tumer, K. (2008). Classifier ensembles: select real-world applications, *Information Fusion*, Vol.9, No.1, (January, 2008), pp. 4-20

Piatetsky-Shapiro, G. (1995). Knowledge discovery in personal data vs. privacy: a mini-symposium, *IEEE Expert: Intelligent Systems and Their Applications*, Vol.10, No.2, (April, 1995), pp. 46-47

Roli, F.; Giacinto, G. & Vernazza, G. (2001). Methods for designing multiple classifier systems, *Proceedings of the 2nd Int. Workshop on Multiple Classifier Systems*, LNCS, Vol.2096, pp. 78-87, London: Springer-Verlag

Rosario, G.; Rundensteiner, E.; Brown, D. & Ward, M. (2004). Mapping nominal values to numbers for effective visualization, *Information Visualization*, Vol.3, No.2, (June, 2004) pp. 80-95

Silva, S. (2006). *Comitês de agrupamento aplicados a dados de expressão gênica*, Master Thesis, Universidade Federal do Rio Grande do Norte, Natal, Brazil

Son, J. & Kim, M. (2004). An adaptable vertical partitioning method in distributed systems, *Journal of Systems and Software*, Vol.73, No.3, pp. 551-561

Sousa, M. (1998). *Mineração de dados: uma implementação fortemente acoplada a um sistema gerenciador de banco de dados paralelo*. Master Thesis, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil

Strehl, A. (2002). *Relationship-based clustering and cluster ensembles for high-dimensional data mining*, PhD Thesis, The University of Texas at Austin, Austin, Texas, USA

Strehl, A. & Ghosh, J. (2002). Cluster ensembles: a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, Vol.3, (March, 2002), pp. 583-617

Thomazi, S. (2006). *Cluster de turismo: introdução ao estudo de arranjo produtivo local*, Aleph, São Paulo, Brasil

Tumer, K. & Agogino, A. (2008). Ensemble clustering with voting active clusters. *Pattern Recognition Letters*, Vol.29, No.14, (October, 2008), pp. 1947-1953

Vaidya, J. & Clifton, C. (2003). Privacy-preserving k-means clustering over vertically partitioned data. *Proceedings of the Ninth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, New York, pp. 206-215

Vaidya, J.; Clifton, C. & Zhu, Y. (2006). *Privacy preserving data mining*, Springer, New York

Vaidya, J. (2008). A survey of privacy-preserving methods across vertically partitioned data, In: *Privacy-preserving data mining*, C. Aggarwal & P. Yu, pp. 337-358, Springer

Verykios, V.; Bertino, E.; Fovino, I.; Provenza, L.; Saygin, Y. & Theodoridis, Y. (2004). State-of-the-art in privacy preserving data mining. *ACM SIGMOD Records*, Vol.33, No.1, (March, 2004), pp. 50-57

Vin, T.; Seng, M.; Kuan, N. & Haron, F. (2005). A framework for grid-based neural networks, *Proceedings of the 1st Int. Conf. on Distributed Frameworks for Multimedia Applications*, pp. 246-253

Vrusias, B.; Vomvoridis, L. & Gillam, L. (2007). Distributing SOM ensemble training using grid middleware, *Proceedings of the 2007 Int. Joint Conf. on Neural Networks*, pp. 2712-2717

Yang, M-H. & Ahuja, N. (1999). A data partition method for parallel self-organizing map, *Proceedings of the 1999 International Joint Conference on Neural Networks*, Vol.3, pp. 1929-1933

Yao, A. (1986). How to generate and exchange secrets, *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, pp. 162-167

Yu, Z.; Zhang, S.; Wong, H-S. & Zhang, J. (2007). Image segmentation based on cluster ensemble, *Proceedings of the 4th Int. Symposium on Neural Networks*, pp. 894-903, China, June, 2007

Zhan, J. (2007). Privacy preserving K-medoids clustering, *IEEE International Conference on Systems, Man and Cybernetics*, (October, 2007), pp. 3600-3603

Zhan, J. (2008). Privacy-preserving collaborative data mining, *IEEE Computational Intelligent Magazine*, (May, 2008), pp. 31-41

Zhang, S.; Wu, X. & Zhang, C. (2003). Multi-database mining, *IEEE Computational Intelligence Bulletin*, Vol.2, No.1, (June, 2003), pp. 5-13

Zhao, Y.; Gao, J. & Yang, X. (2005). A survey of neural network ensembles, *Proceedings of the Int. Conf. on Neural Networks and Brain*, Vol.1, pp. 438-442, October, 2005

Zhou, Z-H. & Tang, W. (2006). Clusterer ensemble, *Knowledge-Based Systems*, Vol.19, No.1, (March, 2006), pp. 77-83

3

A Method for Project Member Role Assignment in Open Source Software Development using Self-Organizing Maps

Shingo Kawamura, Minoru Uehara, and Hideki Mori
Department of Open Information Systems,
Graduate School of Engineering,
Toyo University,
Japan

1. Introduction

We propose the development of open source software (OSS) by a web community. At present, SourceForge.net (http://sourceforge.net/) is a typical community for OSS development, consisting of CVS/SVN repositories, mailing lists, bug tracking systems, task management systems, BBS, open discussion forums, and so on. Although many people are involved in the open discussion forums of SourceForge.net, a project leader is required to expedite and manage the process, which is a great responsibility. Membership is also restricted to expert software developers. In the method proposed in this paper, we envisage not only expert users being able to participate, but also those that use the software but do not have programming skills. Moreover, when the development groups are organized, a supervisor is no longer necessary. The community is managed automatically through the aggregation and distribution of the participating members. The basic concept is exactly the model used by existing SNSs and Wikis, and the development process is based on the spiral model. This paper aims to construct a Web2.0 environment that supports such development and enables the proposal of topics by users. Our method allows the selection of suitable project members from a human resource database (DB) using a self-organizing map (SOM), that is, reinforcement learning. In other words, we propose a web community based on the wisdom of crowds, which is distributed and aggregated. To achieve this, we propose the use of a SOM. Section 2 explains the SOM and the concept of the wisdom of the crowd. The proposed system is described in Section 3. Experiments on feature analysis of the members using a SOM are presented in Section 4, with the results and a discussion thereof given in Section 5 and our conclusions in Section 6.

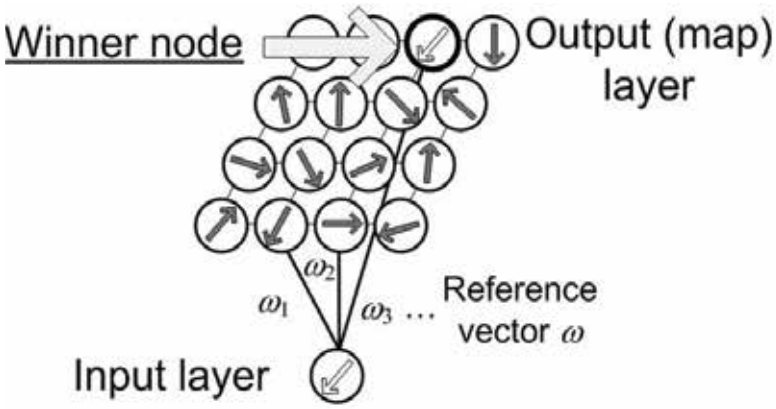
2. Related work

2.1 SOM

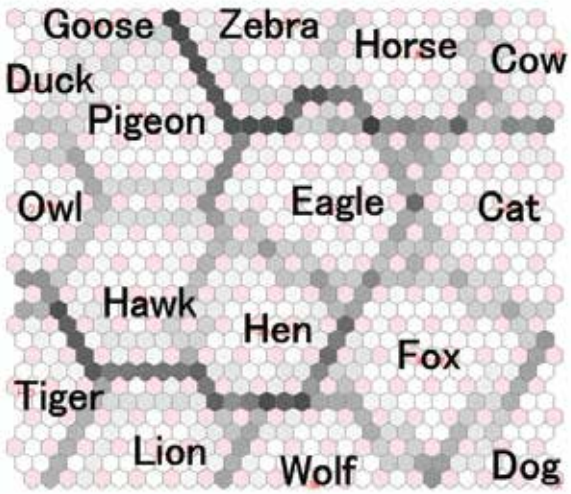
The SOM was designed by Kohonen (1995) at Helsinki University. The neural network is modeled by the visual area in the human brain, and consists of two layers, an input layer and an output (map) layer.

$$d_1 = \sqrt{\sum_{j=1}^{n} \left(x_j - \omega_{ij} \right)^2} \tag{1}$$

x: input vector, ω: reference vector



(a) Architecture of 2 dimensional SOM



(b) Feature analysis

Fig. 1. Example of a SOM

In a SOM, the most common similarity measure is the Euclidean distance, which is defined as the distance of the input vector from the weight vector of a neural node. The nodes are arranged in a two dimensional layout as shown in Figure 1(a), where x is the input vector and ω the reference vector.

The weights of each node are initialized randomly in the initial state of learning. Then, the node, whose weight vector is the closest to the input vector, is selected as the winner node

and the weights of the winner node and its neighbors are updated by the following neighborhood function.

$$\text{if } i \notin N_c \text{, then } h_{ci} = \alpha_0(1 - t/T) \tag{2}$$

$$\text{else } h_{ci}(t) = 0 \quad N_c(t) = N_c(0)(1 - t/T)\text{,} \tag{3}$$

where $\alpha(t)$ is the coefficient of learning, T is the learning time, and N_c is the neighborhood region.

Each weight of a node represents a feature of the object. As an example, animal feature data analyzed by a SOM are shown in Figure 1(b). SOMs are also using in data mining and cluster analysis. In this paper, we use a SOM to ensure diversity of users in a web community. We analyze the characteristics of the available human resources using a SOM. Since the required skill set differs at each stage in a project, we envisage using not only skill, but also personality and aptitude in our analysis, as there may not be any experts participating and aggregation is incorporated in the method.

2.2 Wisdom of crowds

It is said that the service known as the Web2.0 has the characteristics of the "wisdom of crowds", that is, diversity of opinion, independence, decentralization, and aggregation (Fig. 2). Recently, much attention has been focused on crowd sourcing, which is a technique for acquiring productive capacity at an appropriate level in subcontracting development to many unspecified members. Previously, development of OSS was done in a similar manner by many volunteers. In addition, not only OSS development, but also knowledge sharing has been performed spontaneously by networks of individuals with the necessary knowledge (wikis) and experiences (blogs) in recent years. These have the characteristics of the "wisdom of crowds". An advantage of this is that diverse opinions can be reflected, although, on the other hand noisy information tends to be exaggerated.

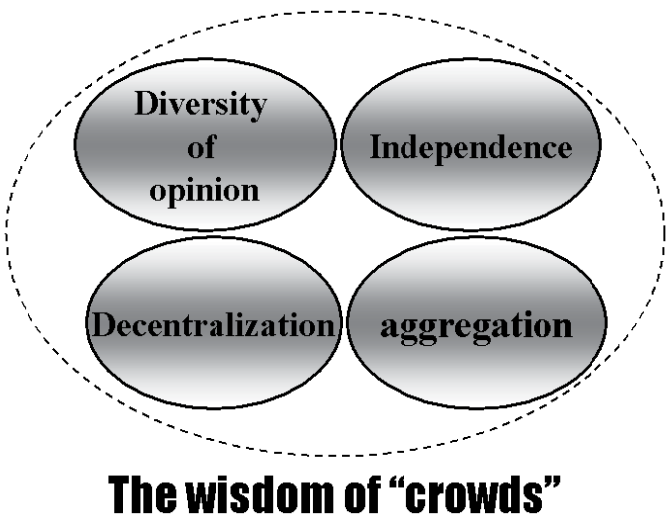


Fig. 2. Wisdom of crowds

3. OSS project

3.1 About C-DOS

OSS development often uses the agile development model that embodies short-term iteration. We propose a web community for OSS development using a SOM implementing the agile model for analyzing developers and their automatic assignment.

C-DOS (Community to Develop OSS using the Spiral model) is the name of the web community dedicated to the development of OSS. C-DOS actually consists of two communities; one to propose OSS projects and the other to promote these projects. Both communities are implemented using OpenPNE (http://docs.openpne.jp), which is an open source SNS environment.

3.2 User participation

Anyone can register and participate in the C-DOS community. When users register, they themselves must evaluate their IT development skills using SWEBOK (SoftWare Engineering Body Of Knowledge) (http://www.swebok.org/) and provide personal preferences to the community.

3.3 Proposal of a new OSS project

Once user registration has been completed, the user has the right to join the community, and is registered in the human resources DB. If users have suggestions for new software development or improvements to existing software, they can provide the requirements of the project. Alternatively, they may put in a request to be included in the development team. All requests are sent to the human resources DB. When the number of applicants exceeds the pre-defined minimum, the suggestion is approved by the community and the project starts. It should be noted that a "group leader" has not traditionally been an integral part of this type of network of members ("suggests" included). Figures 3 and 4 illustrate the process flows from registering a new user to starting a new project.

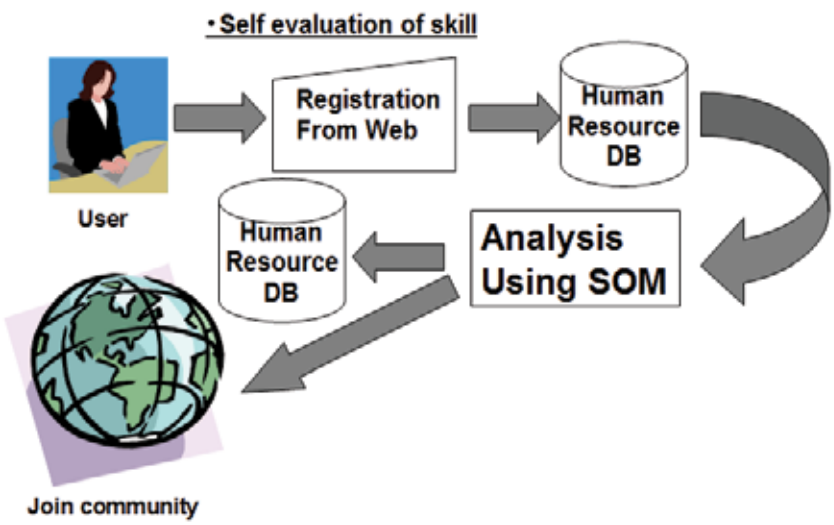


Fig. 3. Process flow for user registration

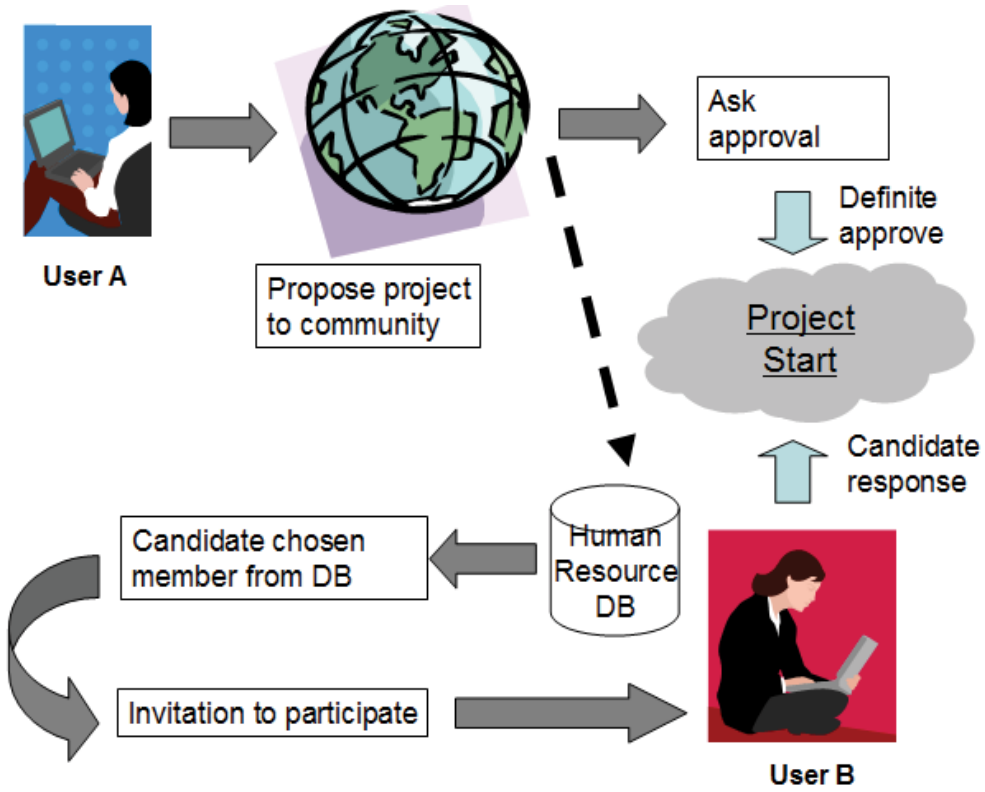


Fig. 4. Process flow for project proposal

3.4 Model for development process

We propose a new development model based on the spiral model (Boehm, 1988). OSS often uses the agile development model, which is flexible enough to be able to change OSS definitions and requirements. But, it is limited to use by expert programmers only. Thus, we have designed a development model for C-DOS that corresponds to the development process. An outline of this model is shown in Figure 5.

Each project is broken down into 4 stages or layers covering the entire process, that is, requirements, designing, building, and testing. After a project has been approved, the project members are assigned to a particular layer by the SOM. Members assigned to each layer can read the working data of the other layers, but cannot alter this data.

When members finish a series of tasks in the current layer, the next group of members begin working on the corresponding next layer.

3.5 Development of an OSS project

Included in the community developing an OSS project are a Wiki and BBS. Members engage in online discussions using the BBS and the results of each layer are written to the Wiki, which can be updated by any of the project members. Updates are finalized when the members in the layer approve the content. Moreover, updated results in the content of the

Wiki are evaluated by all the members of other layers. If agreement is reached through decisions of the majority of members, the task in the layer is considered completed. Importantly, we have designed "bridge layers" between the main layers, where members in a bridge layer have the right to participate in neighboring layers. The bridge layer has the effect of design consistency. Members in the bridge layer are also selected by the SOM, with the number of members being flexible.

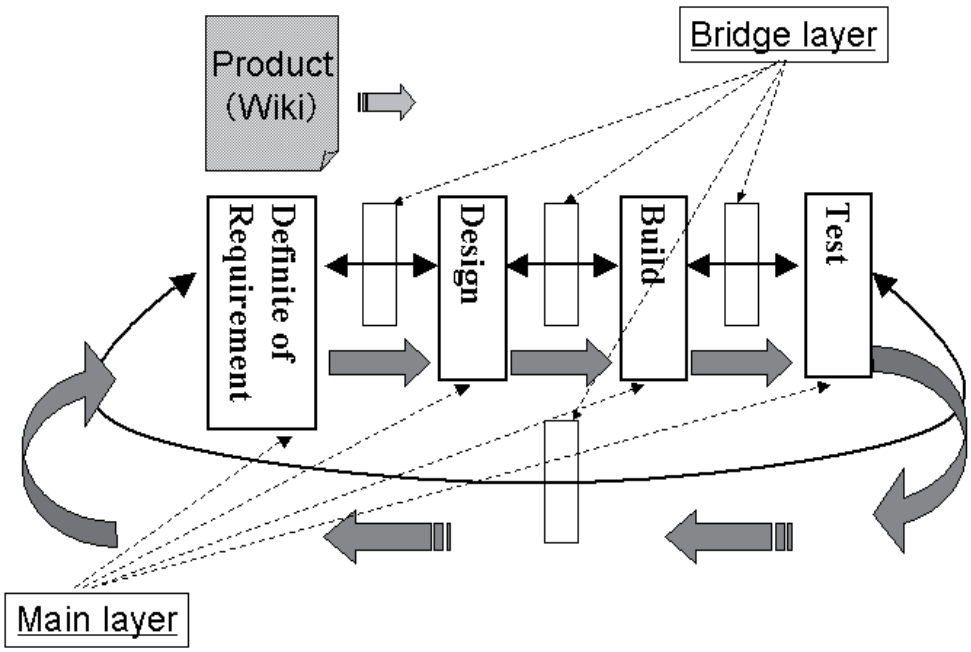


Fig. 5. Model for development of C-DOS

4. Experiment

Processing to assign members to each layer is necessary in C-DOS. This can be achieved by applying the cluster analysis described in this section, in which we conduct a SOM analysis on the information provided on the registration forms and observe the results.

4.1 Skills evaluation

Member self evaluate their knowledge and skills for software development. The resulting input data is based on a five-level rating scale using SWEBOK.

4.2 Analysis of candidate developers using SOM

In this section, we perform a SOM analysis, integrating our knowledge of candidates' interests, hobbies, and their software development skills in relation to the community. Using SWEBOK, it is possible to evaluate a member's knowledge about software development.

SWEBOK, established by a group led by the IEEE in 2001, is a system containing the minimum knowledge that a professional should have acquired after working for four years after graduation from university, The SOM analysis uses SOM_PAK, which is a tool provided by Prof. Kohonen's lab (Kate, 2002).
We show the detailed experimental conditions in Tables 1 and 2. In both evaluations, we follow the taxonomy of Bloom (1956), and users self evaluate their skills in ten phases.

Input Data	21 dimension
Map phase	12 x 8 hexagonal
Learning times	1000

Table 1. Outline of the analysis using a SOM

I.	**Definition of requirement**
i)	Basis of software requirement
ii)	Process of requirement
iii)	Extract of require
iv)	Analyses of requirement
v)	Require to specification
vi)	Confirm appropriateness of requirement
vii)	Consideration matters in practice
II.	**Designing skill**
i)	Basis of software design
ii)	Main problems of software design
iii)	Structure and architecture
iv)	Analyses and evaluation of software designing quality
v)	A notational system of software design
vi)	Tactics and method
III.	**Programming skill**
i)	Basis of building software
ii)	Management of building software
iii)	Consideration matters in practice
IV.	**Testing skill**
i)	Basis of software testing
ii)	Test level
iii)	Techniques of test
iv)	Measure of test
v)	Test process

Table 2. Input data format (knowledge and skill for software development)

Name	I							II						III			IV				
	i	ii	iii	iv	v	vi	vii	i	ii	iii	iv	v	vi	i	ii	iii	i	ii	iii	iv	v
A	10.0	9.0	8.0	8.0	8.0	8.0	8.0	9.0	8.0	9.0	7.0	8.0	8.0	10.0	8.0	8.0	9.0	8.0	8.0	7.0	8.0
B	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	9.0	9.0	9.0	9.0	9.0
C	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.0	9.0	9.0	9.0	8.0	8.0	8.0	8.0	8.0
D	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	7.0	7.0	7.0	7.0	7.0
E	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	7.0	7.0	7.0	7.0	7.0
F	6.0	6.0	6.0	6.0	6.0	6.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0
G	6.0	6.0	6.0	6.0	6.0	6.0	6.0	7.0	7.0	7.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
H	7.0	7.0	7.0	7.0	7.0	7.0	7.0	8.0	8.0	8.0	8.0	8.0	8.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0
I	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	5.0	5.0	5.0	6.0	6.0	6.0	6.0	6.0
J	7.0	7.0	7.0	7.0	6.0	7.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0
K	7.0	6.0	6.0	6.0	6.0	7.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0
L	9.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
M	7.0	7.0	7.0	7.0	7.0	7.0	7.0	8.0	8.0	8.0	8.0	8.0	8.0	9.0	9.0	9.0	7.0	7.0	7.0	7.0	7.0
N	8.3	6.2	6.1	8.9	6.0	6.7	6.8	2.4	3.1	3.7	3.0	2.5	3.1	2.3	3.0	2.9	2.6	2.4	2.4	3.1	3.4
O	7.4	8.8	6.9	8.1	6.7	9.2	7.8	3.6	2.7	3.0	2.5	3.5	2.3	4.0	3.8	3.0	2.0	3.8	3.4	3.4	3.6
P	3.8	3.1	3.1	2.7	3.3	3.1	2.1	7.0	7.8	9.4	8.3	9.9	6.7	3.7	3.0	2.5	4.0	3.5	3.8	3.1	2.7
Q	2.9	3.4	3.7	3.5	3.7	3.7	2.4	8.9	8.6	8.6	8.5	9.7	8.4	2.7	3.1	3.4	2.6	3.9	2.7	3.4	3.8
R	3.3	4.0	2.7	3.8	3.9	2.1	3.5	3.2	3.2	2.8	2.4	3.9	3.4	7.5	8.9	9.6	2.0	3.8	3.5	2.5	2.1
S	3.8	3.8	2.7	3.1	2.6	2.4	2.7	3.1	2.2	3.6	2.1	3.4	3.6	7.6	6.2	7.5	3.5	3.8	2.3	3.8	2.1
T	2.3	3.3	3.6	3.2	2.9	2.2	3.8	2.6	3.4	3.1	3.6	3.7	3.3	2.3	3.6	2.1	7.4	9.9	7.7	9.0	6.4
U	2.9	2.8	3.8	3.5	2.3	3.3	3.9	3.6	3.4	2.2	3.1	3.6	3.9	3.6	3.5	2.8	7.6	8.4	7.4	8.3	9.6
V	2.9	3.8	2.9	3.8	2.2	3.6	2.1	3.2	2.8	3.1	2.1	2.1	3.2	2.9	2.6	3.7	2.7	3.2	2.2	2.5	2.3
W	9.1	9.8	7.1	7.5	7.3	8.1	8.4	6.5	9.9	8.9	7.2	6.6	9.5	6.1	9.9	6.5	6.4	7.1	9.8	9.9	6.9

Table 3. Experimental input data

5. Results

We defined three reference models, namely, a specialist model, in which project members have excellent scores for a specific skill phase, a generalist model, in which members have good scores, above average in all phases, and an ordinary model, in which members generally have immature skill scores. Scores in the 3 reference models are shown as (N-U), W, and V, respectively.
SOM mapping is performed using the sampled score questionnaire from our OSS project members including the reference models' scores. Then, the distance is determined between the sampled location and the reference model locations on the SOM map.
The data used for mapping is shown in Table 3. For instance, N and O specialize in a phase I skill. A two-dimensional SOM is applied to 12x8 arrays. The SOM results are illustrated in Figure 6, while the reference vector after the experiment is given in Table 4. Mappings of the specialist reference model are located on the right of the map and form their own clusters. Member G, because the skill value in phase II is the highest, is mapped close to cluster (P, Q), which belongs to phase II of the specialist reference model. Member H, on the other hand, despite the phase II skill value being as high as that of G, is mapped far from P and Q, because, contrary to the previous case, the other skills are also high. In the generalist case with a high skill value, the mapping is located on the upper left of the map. Ordinary cases with no particular features are scattered around the map and do not belong to any cluster.

6. Conclusion

We proposed a method using a SOM for the assignment of project members in the development of OSS. In the method, member skills are expressed numerically for four software development phases. Then a SOM analysis is carried out on the data, resulting in developers being mapped and located on the map according to their skills. In the experiment, we introduced reference models featuring varying skill levels for particular phases of the development. Using a sample of selected members, we conducted a SOM analysis on the data from the developer questionnaires, and then performed a comparison of the distances between the reference models and the sample. For samples with a high speciality in a particular skill, the mapping shows their location on the map close to the corresponding reference models. According to the results, generalist members with overall good skills in many development phases, are concentrated on the upper left of the map. Members with average skills for each phase, on the other hand, are located randomly on the map. In future research, using the data from previous successful OSS projects, we intend to analyze the relationship between the proposed reference models and actual skills data in order to refine the reference models.

7. Acknowledgment

We would like to thank Mr. N. Oguchi, S. Touji, and K. Matsumoto for their helpful advice and encouragement during our research.

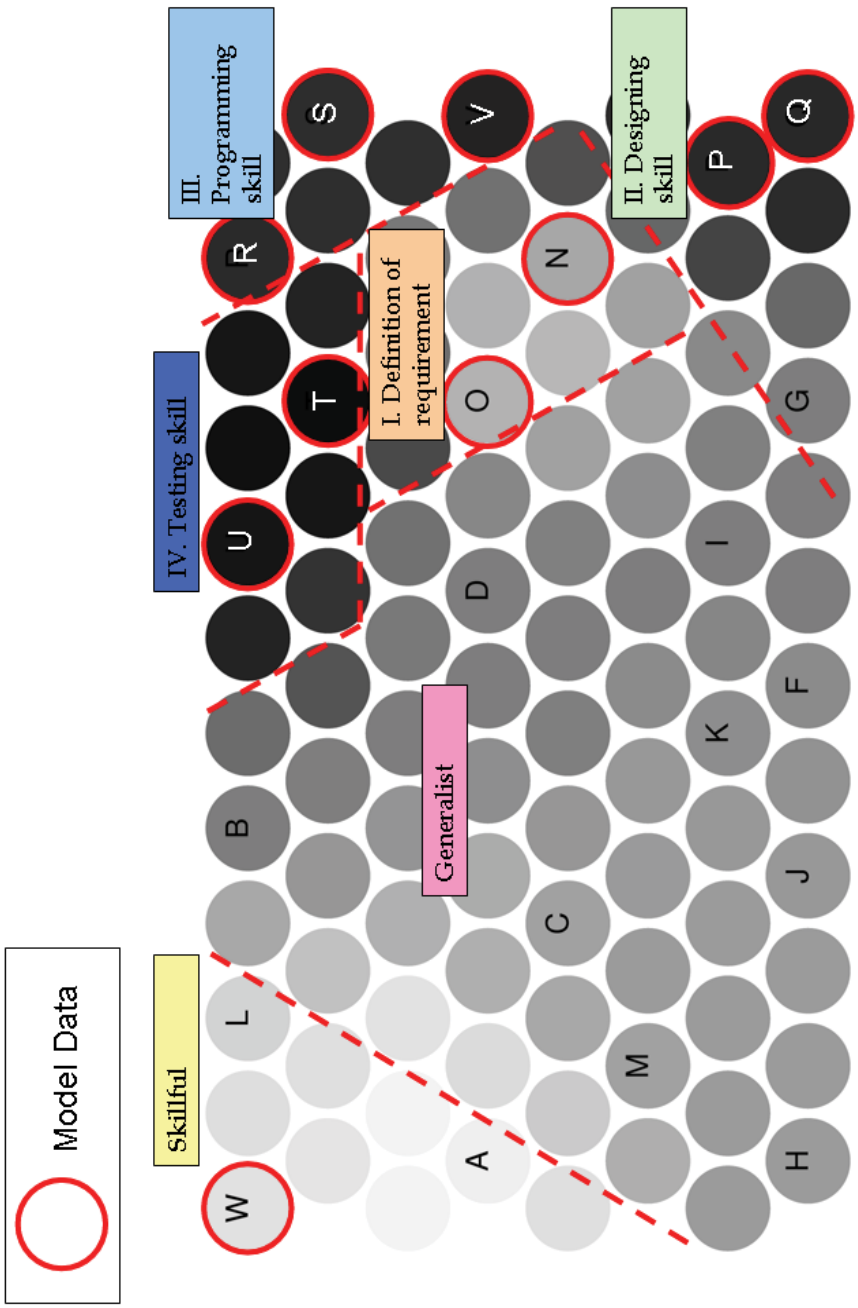


Fig. 6. Experimental results

x dim	y dim	a 1	a 2	a 3	a 4	a 5	a 6	a 7	a 8	a 9	a 10	a 11	a 12	a 13	a 14	a 15	a 16	a 17	a 18	a 19	a 20	a 21	Name
1	1	9.1	9.7	7.2	7.5	7.3	8.1	8.4	6.6	9.8	8.9	7.2	6.7	9.4	6.3	9.8	6.6	6.6	7.2	9.7	9.8	7.0	W
2	1	**9.1**	**9.0**	**7.5**	**7.7**	**7.6**	**8.1**	**8.2**	**7.2**	**9.0**	**8.5**	**7.6**	**7.2**	**8.8**	**7.9**	**9.9**	**8.1**	**8.1**	**8.4**	**9.9**	**9.9**	**8.3**	
3	1	8.7	8.0	7.7	7.7	7.7	7.8	7.8	7.6	8.0	7.9	7.7	7.6	7.9	9.3	9.8	9.4	9.5	9.6	9.9	9.9	9.5	L
4	1	**7.4**	**6.9**	**6.9**	**6.9**	**6.9**	**6.9**	**6.9**	**6.9**	**6.9**	**6.9**	**6.9**	**6.9**	**6.9**	**8.9**	**8.9**	**8.9**	**9.5**	**9.5**	**9.5**	**9.5**	**9.5**	
5	1	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	9.0	9.0	9.0	9.0	9.0	B
6	1	**5.5**	**5.5**	**5.6**	**5.6**	**5.4**	**5.6**	**5.7**	**5.6**	**5.6**	**5.4**	**5.5**	**5.6**	**5.7**	**7.3**	**7.3**	**7.1**	**8.8**	**8.9**	**8.7**	**8.9**	**9.1**	
7	1	**3.2**	**3.2**	**4.0**	**3.8**	**2.8**	**3.5**	**4.1**	**3.8**	**3.7**	**2.7**	**3.5**	**3.9**	**4.1**	**4.0**	**4.0**	**3.3**	**7.7**	**8.6**	**7.6**	**8.5**	**9.2**	
8	1	2.7	2.9	3.7	3.4	2.5	3.0	3.9	3.3	3.4	2.5	3.2	3.6	3.7	3.2	3.5	2.6	7.5	8.8	7.5	8.5	8.7	U
9	1	**2.6**	**3.0**	**3.7**	**3.4**	**2.6**	**2.8**	**3.8**	**3.1**	**3.4**	**2.6**	**3.3**	**3.6**	**3.6**	**3.0**	**3.6**	**2.5**	**7.5**	**9.1**	**7.5**	**8.6**	**8.0**	
10	1	**2.8**	**3.5**	**3.3**	**3.3**	**3.0**	**2.3**	**3.5**	**2.8**	**3.1**	**3.1**	**3.1**	**3.7**	**3.4**	**4.2**	**4.9**	**4.4**	**5.8**	**7.7**	**6.0**	**6.9**	**5.0**	
11	1	3.4	3.9	2.8	3.6	3.4	2.2	3.3	3.1	2.9	3.1	2.4	3.7	3.5	7.2	7.8	8.5	2.8	4.1	3.4	3.3	2.3	R
12	1	**3.5**	**3.9**	**2.7**	**3.5**	**3.2**	**2.3**	**3.1**	**3.2**	**2.7**	**3.2**	**2.2**	**3.6**	**3.5**	**7.5**	**7.5**	**8.5**	**2.8**	**3.8**	**2.9**	**3.1**	**2.1**	
1	2	**9.2**	**9.3**	**7.4**	**7.7**	**7.6**	**8.1**	**8.3**	**7.2**	**9.2**	**8.8**	**7.3**	**7.1**	**8.9**	**7.6**	**9.6**	**7.4**	**7.6**	**7.8**	**9.5**	**9.4**	**7.7**	
2	2	**9.1**	**8.5**	**7.8**	**7.9**	**7.8**	**8.0**	**8.1**	**7.7**	**8.5**	**8.3**	**7.7**	**7.7**	**8.4**	**9.0**	**9.8**	**9.0**	**9.1**	**9.2**	**9.8**	**9.8**	**9.1**	
3	2	**8.2**	**7.4**	**7.4**	**7.4**	**7.4**	**7.4**	**7.4**	**7.4**	**7.4**	**7.4**	**7.4**	**7.4**	**7.4**	**9.4**	**9.4**	**9.4**	**9.7**	**9.7**	**9.7**	**9.7**	**9.7**	
4	2	**6.8**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**8.6**	**8.6**	**8.6**	**9.2**	**9.2**	**9.2**	**9.2**	**9.2**	
5	2	**6.1**	**6.1**	**6.1**	**6.1**	**6.1**	**6.1**	**6.1**	**6.1**	**6.1**	**6.1**	**6.1**	**6.1**	**6.1**	**8.1**	**8.1**	**8.1**	**8.8**	**8.8**	**8.8**	**8.8**	**8.8**	
6	2	**4.7**	**4.7**	**5.1**	**5.0**	**4.5**	**4.9**	**5.1**	**5.0**	**4.9**	**4.5**	**4.8**	**5.0**	**5.1**	**6.2**	**6.2**	**5.9**	**8.1**	**8.4**	**8.0**	**8.3**	**8.9**	
7	2	**3.7**	**3.8**	**4.4**	**4.2**	**3.4**	**3.9**	**4.5**	**4.1**	**4.1**	**3.4**	**4.0**	**4.3**	**4.4**	**4.7**	**4.8**	**4.2**	**7.4**	**8.2**	**7.3**	**8.0**	**8.4**	
8	2	**2.8**	**3.2**	**3.8**	**3.5**	**2.8**	**2.9**	**4.0**	**3.3**	**3.5**	**2.8**	**3.5**	**3.8**	**3.7**	**3.2**	**3.8**	**2.8**	**7.5**	**9.0**	**7.5**	**8.6**	**7.9**	
9	2	2.5	3.2	3.7	3.3	2.7	2.5	3.8	2.9	3.4	2.9	3.5	3.7	3.5	2.7	3.6	2.3	7.4	9.5	7.6	8.8	7.3	T
10	2	**3.2**	**3.7**	**3.4**	**3.7**	**3.3**	**2.5**	**3.7**	**2.8**	**3.1**	**3.2**	**3.0**	**3.6**	**3.4**	**4.6**	**5.3**	**5.0**	**5.1**	**6.8**	**5.3**	**6.1**	**4.4**	
11	2	**3.5**	**3.9**	**2.7**	**3.5**	**3.2**	**2.3**	**3.1**	**3.2**	**2.7**	**3.2**	**2.2**	**3.6**	**3.5**	**7.4**	**7.4**	**8.4**	**2.8**	**3.8**	**2.9**	**3.1**	**2.1**	
12	2	3.5	3.9	2.7	3.4	3.0	2.4	2.9	3.1	2.6	3.2	2.2	3.4	3.5	7.0	6.8	7.8	2.8	3.7	2.7	3.2	2.1	S
1	3	**9.7**	**9.1**	**7.8**	**7.9**	**7.8**	**8.0**	**8.1**	**8.3**	**8.5**	**8.9**	**7.1**	**7.7**	**8.4**	**9.0**	**8.6**	**7.8**	**8.4**	**7.9**	**8.6**	**7.9**	**7.9**	
2	3	**9.7**	**8.8**	**7.9**	**8.0**	**7.9**	**8.0**	**8.0**	**8.6**	**8.1**	**8.8**	**7.2**	**7.9**	**8.1**	**9.7**	**8.6**	**8.3**	**9.0**	**8.4**	**8.6**	**7.9**	**8.3**	
3	3	**9.2**	**8.3**	**7.9**	**7.9**	**7.9**	**7.9**	**7.9**	**8.3**	**7.9**	**8.3**	**7.4**	**7.9**	**7.9**	**9.9**	**9.0**	**9.0**	**9.3**	**8.9**	**8.9**	**8.4**	**8.9**	
4	3	**7.6**	**7.2**	**7.2**	**7.2**	**7.2**	**7.2**	**7.2**	**7.2**	**7.2**	**7.2**	**7.2**	**7.2**	**7.2**	**9.2**	**9.2**	**9.2**	**8.9**	**8.9**	**8.9**	**8.9**	**8.9**	
5	3	**6.8**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**6.6**	**8.6**	**8.6**	**8.6**	**9.0**	**9.0**	**9.0**	**9.0**	**9.0**	

x_{dim}	y_{dim}	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	Name
6	3	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	7.2	7.2	7.2	7.2	7.2	
7	3	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	5.9	7.8	7.8	7.8	7.0	7.1	7.0	7.0	7.1	
8	3	5.6	5.6	5.7	5.7	5.6	5.7	5.8	5.7	5.7	5.6	5.7	5.7	5.8	7.5	7.5	7.4	7.1	7.2	7.1	7.2	7.2	
9	3	4.4	4.9	4.8	5.0	4.4	4.4	5.1	3.6	4.0	3.9	4.0	4.2	3.9	4.0	4.7	3.8	6.2	7.8	6.5	7.3	6.0	
10	3	5.5	5.5	5.1	6.2	4.7	5.2	5.6	2.8	3.1	3.3	3.0	3.2	3.0	3.1	3.6	3.0	4.3	5.6	4.6	5.4	4.5	
11	3	5.8	5.1	4.5	6.1	4.3	4.9	4.7	2.8	2.8	3.5	2.5	2.8	3.2	4.1	4.1	4.6	2.8	3.0	2.4	3.2	2.8	
12	3	3.6	3.9	2.9	3.7	2.7	3.1	2.7	3.1	2.6	3.3	2.2	2.8	3.4	5.3	4.7	5.8	3.0	3.5	2.4	3.1	2.2	
1	4	9.6	8.6	7.9	7.9	7.9	7.9	7.9	8.7	8.0	8.7	7.3	8.0	8.0	9.9	8.5	8.5	9.1	8.4	8.4	7.6	8.4	A
2	4	9.0	8.3	7.7	7.7	7.7	7.7	7.7	8.4	7.8	8.4	7.3	7.8	7.8	9.7	8.5	8.5	8.6	8.0	8.0	7.5	8.0	
3	4	7.6	7.3	7.3	7.3	7.3	7.3	7.3	7.4	7.3	7.4	7.2	7.3	7.3	9.3	9.1	9.1	8.4	8.4	8.4	8.3	8.4	
4	4	7.5	7.2	7.2	7.2	7.2	7.2	7.2	7.2	7.2	7.2	7.2	7.2	7.2	9.2	9.2	9.2	8.5	8.5	8.5	8.5	8.5	
5	4	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	6.6	8.6	8.6	8.6	7.6	7.6	7.6	7.6	7.6	
6	4	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	7.0	7.0	7.0	7.0	7.0	
7	4	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	7.0	7.0	7.0	7.0	7.0	D
8	4	6.3	6.5	6.2	6.5	6.1	6.6	6.4	5.4	5.3	5.4	5.3	5.4	5.2	7.1	7.1	6.9	5.9	6.3	6.2	6.2	6.3	
9	4	7.7	7.6	6.5	8.3	6.4	8.0	7.3	3.3	3.1	3.5	2.9	3.3	2.9	3.6	3.7	3.3	2.6	3.4	3.2	3.5	3.7	O
10	4	7.7	7.4	6.4	8.3	6.2	7.8	7.1	3.0	2.9	3.3	2.7	3.0	2.7	3.1	3.4	3.0	2.3	3.1	2.9	3.2	3.5	
11	4	5.6	5.2	4.6	6.3	4.2	5.3	4.6	2.9	2.9	3.4	2.5	2.5	3.1	3.0	3.1	3.5	2.7	2.9	2.4	2.9	2.9	
12	4	3.1	3.9	3.0	4.0	2.4	3.7	2.3	3.2	2.8	3.1	2.1	2.1	3.2	2.9	2.7	3.7	2.7	3.2	2.2	2.5	2.3	V
1	5	9.1	8.4	7.7	7.7	7.7	7.7	7.7	8.7	8.0	8.7	7.3	8.0	8.0	9.6	8.2	8.2	8.4	7.7	7.7	7.0	7.7	
2	5	8.5	8.0	7.5	7.5	7.5	7.5	7.5	8.4	8.0	8.4	7.5	8.0	8.0	9.5	8.5	8.5	8.0	7.6	7.6	7.1	7.6	
3	5	7.4	7.2	7.1	7.1	7.1	7.1	7.1	7.7	7.6	7.7	7.5	7.6	7.6	9.1	9.0	9.0	7.8	7.7	7.7	7.6	7.7	
4	5	7.1	7.1	7.1	7.1	7.1	7.1	7.1	7.3	7.3	7.3	7.3	7.3	7.3	9.0	9.0	9.0	7.9	7.9	7.9	7.9	7.9	C
5	5	6.8	6.8	6.8	6.8	6.8	6.8	6.8	7.0	7.0	7.0	6.8	6.8	6.8	8.6	8.6	8.6	7.6	7.6	7.6	7.6	7.6	
6	5	6.1	6.1	6.1	6.1	6.1	6.1	6.1	6.2	6.2	6.2	6.1	6.1	6.1	7.9	7.9	7.9	6.9	6.9	6.9	6.9	6.9	
7	5	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	8.0	8.0	8.0	7.0	7.0	7.0	7.0	7.0	
8	5	6.1	6.1	6.0	6.1	6.0	6.1	6.1	5.8	5.8	5.9	5.8	5.8	5.8	7.7	7.7	7.7	6.7	6.8	6.8	6.8	6.8	
9	5	7.1	7.7	6.5	7.6	6.4	8.0	7.2	4.1	3.7	3.9	3.5	4.1	3.4	4.9	4.9	4.4	3.5	4.5	4.3	4.4	4.5	
10	5	7.8	7.5	6.5	8.5	6.4	8.0	7.3	3.0	2.9	3.3	2.7	3.0	2.7	3.2	3.4	3.0	2.3	3.1	2.9	3.3	3.5	
11	5	7.3	6.9	6.0	8.0	5.8	7.3	6.6	3.0	2.9	3.4	2.7	2.8	2.8	3.0	3.3	3.0	2.4	3.0	2.8	3.1	3.3	N

x_{dim}	y_{dim}	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	Name
12	5	4.6	4.6	3.9	5.1	3.5	4.6	3.5	3.5	3.5	4.0	3.1	3.2	3.6	2.9	2.8	3.3	2.8	3.1	2.5	2.8	2.7	
1	6	7.6	7.4	7.2	7.2	7.2	7.2	7.2	7.2	8.2	8.0	8.2	7.8	8.0	8.8	8.4	8.4	7.2	7.0	7.0	6.8	7.0	
2	6	7.1	7.1	7.0	7.0	7.0	7.0	7.0	7.8	7.8	7.8	7.7	7.8	7.8	8.9	8.8	8.8	7.3	7.2	7.2	7.2	7.2	M
3	6	7.0	7.0	7.0	7.0	7.0	7.0	7.0	7.5	7.5	7.5	7.4	7.4	7.4	8.9	8.9	8.9	7.4	7.4	7.4	7.4	7.4	
4	6	6.9	6.8	6.8	6.8	6.6	6.9	6.6	7.0	7.0	7.0	6.7	6.7	6.7	8.3	8.3	8.3	7.3	7.3	7.3	7.3	7.3	
5	6	6.9	6.3	6.3	6.3	6.3	6.9	6.3	7.0	7.0	7.0	6.3	6.3	6.3	7.5	7.5	7.5	6.5	6.5	6.5	6.5	6.5	
6	6	6.4	6.0	6.0	6.0	6.0	6.4	6.0	6.8	6.8	6.8	6.0	6.0	6.0	7.2	7.2	7.2	6.2	6.2	6.2	6.2	6.2	
7	6	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.1	6.1	6.1	6.0	6.0	6.0	6.6	6.6	6.6	6.5	6.5	6.5	6.5	6.5	
8	6	6.5	6.6	6.2	6.8	6.1	6.8	6.5	6.5	5.1	5.0	5.1	4.9	5.1	4.7	4.7	4.5	4.8	5.2	5.1	5.2	5.2	
9	6	7.1	7.7	6.6	7.6	6.4	8.0	7.2	4.3	3.9	4.1	3.3	3.8	3.2	4.1	4.0	3.5	2.9	4.0	3.8	3.8	4.0	
10	6	7.1	7.1	6.1	7.5	6.0	7.5	6.7	3.8	3.7	4.2	3.5	4.1	3.3	3.5	3.5	3.0	2.6	3.4	3.2	3.3	3.5	
11	6	5.3	4.8	4.4	5.0	4.4	5.0	4.1	5.4	5.8	6.9	6.0	7.0	5.1	3.4	3.1	2.7	3.3	3.3	3.4	3.1	3.0	
12	6	3.5	3.3	3.2	3.1	3.2	3.3	2.2	6.6	7.0	8.1	7.2	8.4	6.4	3.4	2.9	2.9	3.5	3.5	3.3	3.0	2.8	
1	7	7.0	7.0	7.0	7.0	7.0	7.0	7.0	8.0	8.0	8.0	8.0	8.0	8.0	7.6	7.6	7.6	6.3	6.3	6.3	6.3	6.3	
2	7	7.0	7.0	7.0	7.0	7.0	7.0	7.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	6.5	6.5	6.5	6.5	6.5	
3	7	7.0	6.9	6.9	6.9	6.8	7.0	6.8	7.7	7.7	7.7	7.5	7.5	7.5	8.2	8.2	8.2	6.6	6.6	6.6	6.6	6.6	
4	7	7.0	6.9	6.9	6.9	6.3	7.0	6.3	7.1	7.1	7.1	6.4	6.4	6.4	7.5	7.5	7.5	6.4	6.4	6.4	6.4	6.4	
5	7	6.9	6.5	6.5	6.5	6.0	6.9	6.0	7.0	7.0	7.0	6.0	6.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0	
6	7	6.6	6.1	6.1	6.1	6.0	6.6	6.0	7.0	7.0	7.0	6.0	6.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0	K
7	7	6.3	6.0	6.0	6.0	6.0	6.3	6.0	6.7	6.7	6.7	6.0	6.0	6.0	6.3	6.3	6.3	6.0	6.0	6.0	6.0	6.0	
8	7	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.4	6.4	6.4	5.8	5.8	5.8	5.3	5.3	5.3	5.8	5.8	5.8	5.8	5.8	I
9	7	6.1	6.1	6.0	6.1	6.0	6.1	6.1	6.3	6.3	6.3	5.3	5.4	5.3	4.9	4.9	4.9	5.3	5.4	5.4	5.4	5.4	
10	7	6.4	6.5	6.2	6.6	6.1	6.6	6.4	5.9	5.8	5.9	4.6	4.7	4.6	4.6	4.7	4.5	4.5	4.8	4.7	4.8	4.8	
11	7	4.2	3.9	3.9	3.7	4.0	3.9	3.1	7.1	7.6	8.6	7.5	8.7	6.5	3.7	3.4	3.1	3.9	3.8	3.8	3.5	3.3	
12	7	3.4	3.3	3.4	3.1	3.5	3.4	2.3	7.9	8.2	9.0	8.4	9.8	7.5	3.2	3.1	3.0	3.3	3.7	3.2	3.3	3.3	P
1	8	7.0	7.0	7.0	7.0	7.0	7.0	7.0	8.0	8.0	8.0	8.0	8.0	8.0	7.3	7.3	7.3	6.2	6.2	6.2	6.2	6.2	H
2	8	7.0	6.9	6.9	6.9	6.9	7.0	6.9	7.9	7.9	7.9	7.8	7.8	7.8	7.5	7.5	7.5	6.2	6.2	6.2	6.2	6.2	
3	8	7.0	6.8	6.8	6.8	6.1	7.0	6.1	7.1	7.1	7.1	6.2	6.2	6.2	7.1	7.1	7.1	6.0	6.0	6.0	6.0	6.0	
4	8	6.9	6.7	6.7	6.7	6.0	6.9	6.0	7.0	7.0	7.0	6.0	6.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0	J
5	8	6.7	6.3	6.3	6.3	6.0	6.7	6.0	7.0	7.0	7.0	6.0	6.0	6.0	7.0	7.0	7.0	6.0	6.0	6.0	6.0	6.0	
6	8	6.5	6.0	6.0	6.0	6.0	6.5	6.0	7.0	7.0	7.0	6.0	6.0	6.0	6.9	6.9	6.9	6.0	6.0	6.0	6.0	6.0	F
7	8	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.5	6.5	6.5	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	
8	8	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.5	6.5	6.5	5.5	5.5	5.5	5.0	5.0	5.0	5.5	5.5	5.5	5.5	5.5	
9	8	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.7	6.7	6.7	5.3	5.3	5.3	5.0	5.0	5.0	5.3	5.3	5.3	5.3	5.3	G
10	8	5.4	5.2	5.3	5.2	5.3	5.3	5.0	7.0	7.2	7.5	6.0	6.4	5.6	4.6	4.5	4.4	4.7	4.7	4.7	4.6	4.5	
11	8	3.4	3.4	3.5	3.2	3.6	3.5	2.4	7.9	8.2	8.9	8.3	9.6	7.5	3.3	3.1	3.0	3.4	3.7	3.3	3.3	3.3	
12	8	3.3	3.3	3.4	3.1	3.5	3.4	2.3	8.0	8.2	9.0	8.4	9.8	7.6	3.2	3.1	3.0	3.3	3.7	3.2	3.3	3.3	Q

Table 4. Reference vector

8. References

S. Bloom. (Ed.) (1956). Taxonomy of Educational Objectives: The Classification of Educational Goals; *Handbook* I, Cognitive Domain, New York

W. B. Boehm. (1988). Spiral Model of Software Development and Enhancement, *IEEE Computer* 21(5): pp. 61-72

M. Furst. & R. A. DeMillo. (2006). Creating Symphonic-Thinking Computer Science Graduates for an Increasingly Competitive Global Environment p.3. (http://www.cc.gatech.edu/images/pdfs/threads_whitepaper.pdf)

A.S. Kate (Ed.) (2002). *Neural Networks in Business: Techniques and Applications*, IGI, Hershey

T. Kohonen. (1995). *Self-Organizing Maps*, *Springer*, New York

J. Surowiecki. (2005). *The Wisdom of Crowds, Anchor*, New York

Part 2

Data Envelopment Analysis

4

Modelling with Self-Organising Maps and Data Envelopment Analysis: A Case Study in Educational Evaluation

Lidia Angulo Meza, Luiz Biondi Neto, Luana Carneiro Brandão, Fernando do Valle Silva Andrade, João Carlos Correia Baptista Soares de Mello and Pedro Henrique Gouvêa Coelho
Universidade Federal Fluminense and Universidade do Estado do Rio de Janeiro
Brazil

1. Introduction

In this chapter we deal with a problem of educational evaluation. We deal with an organization for distance education in the State of Rio de Janeiro, Brazil. This organization is the centre for distance undergraduate education in the Rio de Janeiro State (CEDERJ for the name in Portuguese). Although CEDERJ provides a wide set of undergraduate courses we focus ourselves on the Mathematics undergraduate course. The choice of this course is due to the fact that it exists since the very beginning of the CEDERJ.

We do not intend to evaluate distance undergraduate education itself. That is, we will not compare results from distance undergraduate education with results from *in situ* undergraduate education. Instead, we will compare distance education with itself, thus meaning we will evaluate some thirteen centres of distance education, all of them belonging to the CEDERJ. We want to determine the best managerial practices and the most favourable regions to inaugurate new CEDERJ centres.

The comparison hereabove mentioned takes into account how many students finish the course in each centre, how many students have began the course and the proxy for the resources employed in each centre. In the present chapter, we only consider graduates as outputs because graduating students is the main target of CEDERJ, while producing researches have low priority.

In order to perform this evaluation, we will use a non parametric technique known as Data Envelopment Analysis – DEA. Initially developed by Charnes et al (1978), this technique deals with productive units, called Decision Making Units (DMUs). The DMUs use the same inputs to produce the same outputs and the DMUs set must be homogenous, i.e. they must work in similar environmental conditions. It is important to notice that these DMUs are not necessarily units involved in a productive or manufacture process, but they can be entity using resources (inputs) to generate some kind of products (outputs).

In our case, the homogenous conditions are not verified since CEDERJ centres are located in different regions of the Rio de Janeiro State with different socio economical conditions that cannot be considered in the evaluation. So, in order to perform a DEA evaluation, we need

to separate the centres in homogenous clusters according to their environmental conditions. To do that, we use the Kohonen self-organizing maps to cluster the centres. This is done taking into account some environmental variables.
After the clustering of the centres, we perform a DEA evaluation inside each cluster and overall DEA evaluation using an handicap index to compare the heterogeneous DMUs. We also identify the efficient centre and the benchmarks for the inefficient ones.
As mentioned above, this chapter deals with Data Envelopment Analysis and Kohonen Self Organizing Maps. The self-organising maps are a special case of neural networks. There are already some papers dealing with the use of Neural Networks and Data Envelopment Analysis altogether. For instance, Samoilenko and Osei-Bryson (2010) use Neural Networks and DEA to determine if the differences among efficiency scores are due to environmental variables or the management process. The use of Neural Network for clustering and benchmarking container terminals was done by Sharma and Yu (2009). Also Churilov and Flitman (2006) used Kohonen self-organizing maps to cluster countries participating of the Olympics and then using DEA for producing a new ranking of participating teams. Emrouznejad and Shale (2009) and Biondi Neto et al. (2004) used the back propagation neural network algorithm to accelerate computations in DEA. Çelebi and Bayraktar (2008) used Neural Networks to estimate missing information for suppliers evaluation using DEA.
This chapter is organized as follows; in the next two sections we briefly present the fundamentals of Data Envelopment Analysis (DEA) and Kohonen Neural Networks. In each of these sections we also present a brief bibliographical review of each one in the area of interest in this chapter, educational evaluation. In section 4, we present our case study, the CEDERJ distance undergraduate centres. Kohonen maps are used to cluster and DEA to evaluate the CEDERJ centres. Finally we present some conclusions, our acknowledgments and the references.

2. The fundamentals of data envelopment analysis

Data Envelopment Analysis – DEA was initially developed by Charnes et al. (1978) for school evaluation. This is a linear programming method to compute Decision Making Units – DMUs comparative efficiencies whenever financial matters are neither the only ones to take into consideration nor even the dominant ones. A DMU relative efficiency is defined as the ratio of the weighted sum of its outputs to the weighted sum of its inputs.
Contrary to traditional multi-criteria decision aid models there is no arbitrary decision-maker that chooses the weights to be assigned to each weighing coefficient. These obtain instead from the very mathematical model. To do so, a fractional programming problem is solved to assign to each DMU the weights that maximize its efficiency. The weights are thus different for each unit and they are the most advantageous for the unit. So the DEA approach avoids the criticism from unit managers whose evaluation was not good that the weights were biased.
DEA models can take into account different scales of operation. When that happens the model is called BCC (Banker et al., 1984). When efficiency is measured taking no account of scale effects, the model is called CCR (Charnes et al., 1978). The formulation for the previously linearized fractional programming problem is shown in (1) for the DEA CCR (Cooper et al., 2000, Seiford, 1996).
For model (1) with n DMUs, m inputs and s outputs, let h_o be the efficiency of DMU o being studied; let x_{ik} be i input of DMU k, let y_{jk} be j output of DMU k; let v_i be the weight assigned

to i input; let u_j be the weight assigned to j output. This model must be solved for each DMU.

$$\begin{aligned} &\min \ \sum_{i=1}^{m} v_i x_{io} \\ &\text{st} \\ &\sum_{j=1}^{s} u_j y_{jo} = 1 \\ &\sum_{j=1}^{s} u_j y_{jk} - \sum_{i=1}^{m} v_i x_{ik} \le 0 \ , \quad k = 1,...,n \\ &u_j, v_i \ge 0 \quad \forall x, y \end{aligned} \tag{1}$$

Evaluating governmental institutions, such as CEDERJ and other educational institutions, is difficult mainly because of the price regulation and subventions, what generally leads to distortion (Abbott & Doucouliagos, 2003). However, DEA does not require pricing, and this is why it is broadly used for this type of evaluations.

DEA has been widely used in educational evaluation. For instance, Abbott & Doucouliagos (2003) measured technical efficiency in the Australian university system. They considered as outputs many variables referring to research and teaching. Abramo et al (2008) evaluated Italian universities, concerning basically scientific production.

The first authors went through analysis using various combinations of inputs and outputs, because the choice of the variables can greatly influence how DMUs are ranked, which is similar to what is done the process of variable selection in the present paper. The seconds also verify the importance of choosing the right variables, by comparing the final results with analysis of sensitivity, and observing how different they are.

Abbott & Doucouliagos (2003) introduce the concept of benchmarking as one of DEA strengths, though neither of the articles actually calculates it. Finding benchmarks and anti-benchmarks is important for the study's applicability, since it is the first step to improving the inefficient DMUs. These authors also propose clustering the universities, according to the aspects of tradition and location (urban or not), which in their work, does not significantly affect results.

A more comprehensive review of DEA in education can be found in Soares de Mello et al (2006).

3. Fundamentals of Kohonen maps

The human brain organizes information in a logic way. The cortex has billions of neurons with billions of synaptic connections among them involving nearly all brain. The brain is orderly divided in subsections including: motor cortex, somatosensory cortex, visual cortex, auditory cortex. The sensory inputs are orderly mapped to those cortex areas (Kohonen, 2001, Haykin, 1999, Bishop, 1995).

It seems that some of these cells are trained in a supervised way and others in a supervised and self-organized way.

A paramount aspect of the self-organized networks is motivated by the organization of the human brain in regions in such a way that the sensory inputs are represented by

topologically organized maps. The Kohonen self-organizing map emulates that unsupervised learning in a simple and elegant way and also taking into account the neuron neighbourhood (Mitra et al., 2002).

The topographic map development principle according to Kohonen (2001) is as follows:

"The space location of an output neuron in a topographic map corresponds to a particular domain or feature of data drawn from the input space"

From that principle came up two feature mapping models: the Willshaw (1969) and Willshaw and Von der Malsburg (1976) model, having strong neurobiological motivations, and the Kohonen (2001) model, not as close to neurobiology as the previous one but enabling a simple computing treatment stressing the essential characteristics of the brain maps. Moreover, the Kohonen model depicted in Figure 1 yields a low input dimension.

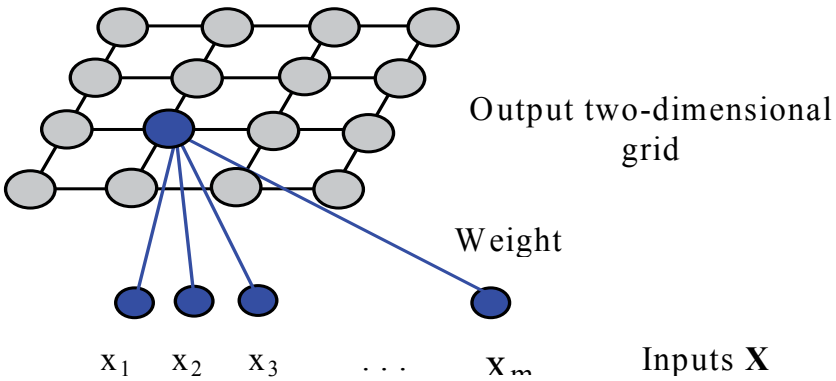


Fig. 1. Kohonen Self-Organizing Map

Another way to characterize a SOM (self-organizing maps) is shown in Figure 2. In that case, it is easily seen that each neuron receives identical input set information.

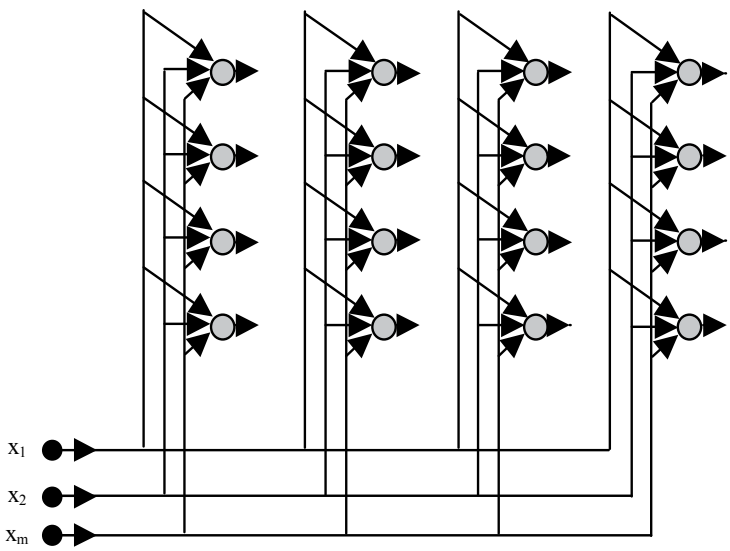


Fig. 2. Another way to represent Kohonen maps

The SOMs are Artificial Neural Networks (ANN) special structures in a grid form that work in a similar way of the human brain as far as the information organization is concerned, and are based on competitive learning. The most used SOM is the topologically interconnected two-dimensional, where the neurons are represented by rectangular, hexagonal and random grid knots of neighbour neurons. Higher dimensional maps can also be modelled. In Fig ure 3 one can see the neuron position in a (8X8) hexagonal representation.

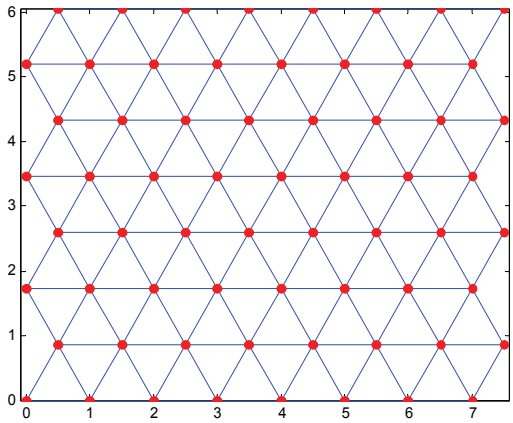


Fig. 3. Hexagonal neuron positions

In order to analyze the competitive process, let us suppose that the input space is m-dimensional and that X represent a random input pattern (Haykin, 1999) such that one can write

$$\boldsymbol{X} = [x_1\ x_2\ x_3\ \ldots\ x_m]^t \tag{2}$$

Assuming the weight vector W of each neuron has the same dimension as that of the input space, for a given neuron j of a total of l neurons, the weight vector can be written as

$$\boldsymbol{W}_j = [w_{j1}\ w_{j2}\ w_{j3}\ \ldots\ w_{jm}]^t, \quad j = 1\ 2\ 3\ \ldots, l \tag{3}$$

For each input vector, the scalar product is evaluated in order to find the X vector which is closest to the weight vector W. By comparison, the maximum scalar product as defined in (4) is chosen, representing the location in which the topological neighbourhood of excited neurons should be centred,

$$\max\ (\boldsymbol{W}_j^t.\boldsymbol{X}), \quad j = 1\ 2\ 3\ \ldots\ ,l \tag{4}$$

Maximizing the scalar product in (4) is equivalent to minimize the Euclidian distance between X and W. Figure 4 shows that the less the Euclidian distance the more approximation between X and W.

Other metrics such as Minkowski, Manhatten, Hamming, Hausdorf, Tanimoto coefficients and angle between vectors could also be used (Kohonen, 2001, Haykin, 1999, Michie et al., 1994).

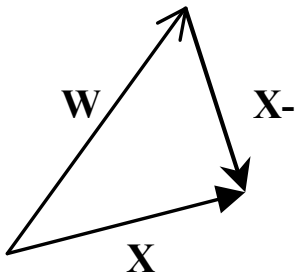


Fig. 4. Minimization of Euclidian Distance

The closest neuron to the input vector X, given by (5), is called the winner neuron whose index is V(X), where

$$V(X) = \min \left\| X - W_j \right\|, \quad j = 1\ 2\ 3\ ...,l \tag{5}$$

By means of a competitive process, a continuous input space pattern can be mapped into a discrete output space of neurons.

In the cooperative process, the winner neuron locates the centre of a topological neighbourhood of cooperating neurons, which is biologically defined by the existence of interactive lateral connections in a cluster of biological neural cells. So the active winner, the winner one, tends to strongly stimulate its closest neighbour neurons and weakly the farthest ones. It is apparent that the topological neighbourhood concerned to the winner neuron decreases with increasing lateral distance.

It is essential to find a topological neighbourhood function $sN_{j,V(X)}$, that be independent from the winner neuron location written in (5). That neighbourhood function should represent the topological neighbourhood centred in the winner neuron, indexed by V, having as closest lateral neighbours, a group of excited neurons and cooperative ones from which a representative can be chosen which is denominated j neuron. The lateral distance, $D_{j,V}$, between the winner neuron indexed, by V, and the excited neuron, indexed by j can be written as in (6) (Haykin, 1999).

$$N_{j,V(X)} = \exp\left(-\frac{D_{j,V}^2}{2\sigma^2} \right) \tag{6}$$

where σ is the neighbourhood width.

The topological neighbourhood function A $N_{j,V(\mathbf{X})}$ shown in Fig. 5 should have the following properties (Mitra et al., 2002, Haykin, 1999):

- Be symmetric relative to the point of maximum, characterized by the winner neuron, indexed by V(**X**), for which $D_{j,V} = 0$.
- When $D_{j,V}$ goes to ± ∞, the magnitude of the topological neighbourhood function monotonically decreases, tending towards zero.

The more dependent the lateral distance $D_{j,V}$ be, the greater will be the cooperation among the neighbourhood neurons. So, for a two-dimensional output grid, the lateral distance can be defined as in (7), for which the discrete vector $\wp_j$ represents the position of the excited neuron, and $\wp_V$ the position of the neuron that won the competition.

$$D_{j,V} = \sqrt{\left\| \wp_j - \wp_V \right\|^2} \tag{7}$$

Another point to be considered is that the topological neighbourhood should decrease with discrete time n. In order to accomplish that, the width σ, of the topological neighbourhood $N_{j,V(X)}$ should decrease in time. That could be achieved if the width of the topological neighbourhood decreases in time. The width could be written as in (8) where σ_0 represents the initial value of the neighbourhood width and τ_1 a time constant. Usually σ_0 is adjusted to have the same value as the grid ratio, i.e. $\tau_1=1000/\log \sigma_0$.

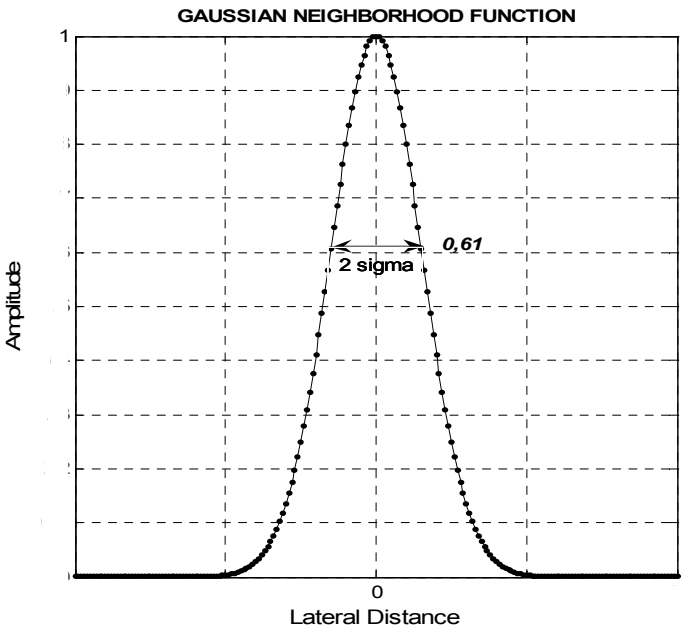


Fig. 5. Gaussian neighbourhood function

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1}\right), \quad n = 0, 1, 2, 3,... \tag{8}$$

The expression of the topological neighbourhood in time can be written as

$$N_{j,V(X)}(n) = \exp\left(-\frac{D_{j,V}^2}{2\sigma^2(n)}\right), \quad n = 0, 1, 2, 3, ... \tag{9}$$

The adaptive process is the last phase of the self- organizing map procedure and during this phase the adjustment of the connection weights of the neurons are carried out. In order the network succeed in the self-organization task, it is necessary the weights $\mathbf{W}_j$ of the excited **j** neuron be updated relatively to the input vector **X**.

Due to the connection changes that happen in one direction, the Hebb rule can not be used in the same way as in the supervised learning that would lead the weights to saturation. For

that, a slight change is done in the Hebb rule, including a new term $g(y_j)\,\mathbf{W}_j$ called forgetting term, in which $\mathbf{W}_j$ is the vector weight of the excited **j** neuron and $g(y_j)$ is a positive scalar function of the output y_j of neuron **j**. The only requirement imposed on the function $g(y_j)$ is that the constant term in the Taylor series expansion of $g(y_j)$ be zero, so that $g(y_j) = 0$ for $y_j = 0$. Given such a function, the change to the weight vector of the excited neuron **j** in the grid can be written as in (9) where $\boldsymbol{\eta}$ is the learning rate parameter.
The first term in equation (10) is the Hebbian term and the second the forgetting (Kohonen, 2001, Haykin, 1999, Bishop, 1995).

$$\Delta W_j = \eta y_j X - g(y_j) W_j \tag{10}$$

In order to satisfy the requirement, a linear function for $g(y_j)$ is chosen as

$$g(y_j) = \eta y_j \tag{11}$$

Using $y_j = N_{j,V(X)}$, equation (10) can be written as (12) as

$$\Delta W_j = \eta N_{j,V(X)} (X - W_j) \tag{12}$$

Using discrete-time notation a weight updating equation can be written which applies to all neurons that are within the topographic neighbourhood equation of the winner neuron (Kohonen, 2001, Haykin, 1999),

$$W_j(n+1) = W_j(n) + \eta(n) N_{j,V(X)}(n)(X - W_j(n)) \tag{13}$$

In (13) the learning rate parameter changes each iteration, with an initial value around 0.1 and decreasing with increasing discrete-time n up to values above 0.01 (Mitra et al., 2002). To that end, equation (14) is written in which η decays exponentially and τ_2 is another time-constant of the SOM algorithm. For the fulfilment of the requirements one could choose for instance, $\eta_0 = 0.1$ and $\tau_2 = 1000$.

$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right), \quad n = 0, 1, 2, 3, \ldots \tag{14}$$

Self-organizing maps have been widely used in many fields. For instance, regarding the subject of the present chapter, Kohonen networks have been used in education for peer identification process in business schools (re)accreditation process (Kiang et al., 2009) and to determine students' specific preferences for school websites (Cooper & Burns, 2007).
In the Brazilian Rio de Janeiro state self-organized maps were used to cluster cities according to characteristics of electrical consumption (Biondi Neto et al., 2007).
Then the self-organizing maps will be used to cluster CEDERJ distance education centres, in order to perform a DEA evaluation.

4. Distance learning in Rio de Janeiro: The CEDERJ

One of CEDERJ's main target is to contribute with the geographic expansion of undergraduate public education. This is also one of the targets of public universities in general. A second main target is to grant access to undergraduate education for those who

are not able to study in regular hours, usually because of work. Finally, developing the state's high school teachers and offering vacancies in graduate courses are also targets to be achieved. We can notice that many of these are similar to UAB's aims, which is a consequence of the fact that CEDERJ is part of the UAB system.

In CEDERJ, students have direct contact with tutors, who are of great importance (Soares de Mello, 2003) for they are responsible for helping students with their subjects as well as their motivation. Its pedagogical program is based on advances in the area of information and communication technologies, but also offers practical classes in laboratories. Students receive printed and digital material, which includes videos, animations, interactivity with tutors, teachers, other students and guests. This whole environment helps creating knowledge.

Its expansion in terms of number of local centres and types of courses brings up the need to evaluate CEDERJ globally, since the system consumes public resources, and also locally, in order to reduce eventual differences.

Gomes Junior *et al* (2008) evaluated CEDERJ courses using the so called elementary multi-criteria evaluation, Condorcet, Copeland and Borda (Roy & Bouyssou, 1993). The authors point out that there is an apparent relation between regions wealth and its position in the final ranking; and a reverse relation between the number of regular universities and the local centre's position. In the present study, these variables should be considered when clustering the local centres.

Menezes (2007) made a scientific investigation on distance education, focusing on CEDERJ, analysing how new information and communication technologies impact on time and space organization.

There are many other studies on CEDERJ, yet they are mostly qualitative. Qualitative literature allows different interpretations, and it might become clearer with measurable facts. Our goal is with this quantitative approach to complement the existent qualitative literature, with no intention to replace it.

5. Evaluation of CEDERJ with DEA and Kohonen maps

The DMUs being evaluated in the present research are the local centres that offer Mathematics undergraduate course, therefore each of the following variables are related to the Math course in each local centre.

AI – Number of students enrolled in the course in a certain semester (*input*)

NT – Number of tutors in the first semester of 2009 (*input*) proxy for the resources used in the centre.

AF – Number of students that graduated in the first semester of 2009 (*output*)

There are other professionals, besides tutors, involved in the CEDERJ system, such as those responsible for preparing the material. However, the Math material is the same in every local centre, so these professionals should be attributed to each course, not to each local centre.

Seeking the semester that should be used for the first input, a process of variable selection is carried out because, according to Thanassoulis (1996), the group of variables used in the analysis can have great impact on its result. Therefore, in this evaluation process, variables are selected in a way that inputs better explain outputs and that less DMUs have maximum efficiency.

This process has been performed on the work of Andrade et al (2009) and it aimed to obtain a set of values for the AI variable, considering 1st and 2nd semesters of 2005 (1/2005 and 2/2005, respectively) and 1st semester of 2006 (1/2006). Since the graduation semester is the 1st one of 2009 (1/2009) and that the Math course has eight semesters of duration, it would be normal to use the number of enrolled students in 2/2005. Nevertheless, students may anticipate or postpone their graduation and therefore another semester might be chosen as the one that better explains the outputs. If 1/2005 is chosen, for example, it means that the majority of students postpone their graduation.

Although 24 local centres offer the Math course, only 13 have had graduates in 1/2009. Therefore only these 13 centres can be considered in the model, otherwise, results might be distorted because of the zero output. Besides the 24 centres, other four centres offer math tutorials – not the whole course, only tutorials. These, however, are not considered in this work.

According to the process of variable selection demonstrated in Andrade *et al* (2009), the semester chosen for the number of students enrolled in the course in a certain semester (AI) is 2/2005.

Another point to be considered is that local centres are subjected to different social, environmental and structural realities (Gomes Junior et al., 2008). This is important because in order to use DEA and compare DMUs, we should guarantee that they are homogeneous. The CEDERJ centres are located in regions with socio-economic characteristics very different among them. So, the DMUs are clearly non homogenous. If we try to use DEA with the complete set of centres we will have a DEA model with non homogenous DMU. This is a well-know pitfall in DEA (Dyson et al., 2001). So, we must be divided into clusters with homogeneous characteristics before using DEA. Afterwards, a homogenisation process will be carried out to perform an overall evaluation.

5.1 Clustering the DMUs

For the clustering of the CEDERJ centres we used the Kohonen self-organizing maps. The variables used were:

- The number of vacancies as a proxy to the size of the centre.
- The ratio of the candidates per vacancies for the Maths undergraduate course as a proxy to the cognitive level of the students enrolled in the course.
- The city's Human Development Index (HDI) as a proxy for the socio economical characteristics of the city.

The number of semesters since the opening of the centre as a proxy to the maturity of the centre.

Different configurations for the Kohonen Maps were tested. We used grids with the (6x6), (4x4), (3x4), (3x3), (3x2) representations. Of all the clusters obtained we selected the one that did not let a centre isolated, which allows a better condition to perform an efficiency analysis using Data Envelopment Analysis. This was achieved using a grid with the (3x3) and the (3x2) representations, with the same clustering. The final clustering is shown in Table 1.

We obtained four clusters, with the mentioned representation, that contain centres with similar characteristics regarding size, students level, centre's maturity and socio-economical characteristics as explain previously.

Cluster	Centre
1st Cluster	Volta Redonda
	Paracambi
	Petrópolis
2nd Cluster	Angra dos Reis
	São Pedro da Aldeia
	Saquarema
	Três Rios
	Campo Grande
3rd Cluster	Macaé
	Piraí
	São Fidelis
4th Cluster	Cantagalo
	Itaperuna

Table 1. Centres Clustering

5.2 Evaluation in each cluster

Once the clustering process is finished we performed the evaluation inside each cluster. We use the CCR output oriented model shown in section 2. The data, for the three variables considered, and the results for each one of the four clusters can be found in Tables 2, 3, 4 and 5.

Centre	Inputs		Output	Efficiency Index (%)
	AI	NT	AF	
Volta Redonda	99	10	10	80.80
Paracambi	72	7	9	100.00
Petrópolis	79	8	1	10.00

Table 2. Efficiency Index for the Centres in cluster 1

Centre	Inputs		Output	Efficiency Index (%)
	AI	NT	AF	
Angra dos Reis	60	6	8	100.00
São Pedro da Aldeia	20	5	1	37.50
Saquarema	62	6	4	50.00
Três Rios	60	8	3	37.50
Campo Grande	62	6	1	12.50

Table 3. Efficiency Index for the Centres in cluster 2

Centre	Inputs		Output	Efficiency Index (%)
	AI	NT	AF	
Macaé	29	6	3	50.00
Piraí	23	6	6	100.00
São Fidelis	61	6	2	33.33

Table 4. Efficiency Index for the Centres in cluster 3

Center	Inputs		Output	Efficiency Index (%)
	AI	NT	AF	
Cantagalo	40	7	2	50.00
Itaperuna	36	7	4	100.00

Table 5. Efficiency Index for the Centers in cluster 4

In these tables we can see that we obtained exactly one efficient centre in each cluster. This shows that despite having few DMUs in each cluster, DEA had success in obtaining a ranking in each cluster.

We can also observe that there are notorious differences among the efficiency indexes in the same cluster. A large proportion of centres are less than 50% efficient. This is not usual in DEA.

5.3 Clusters evaluation

In performing the clustering and DEA evaluation in each cluster we take into account the differences in the environmental conditions of the centres. Now we are going to perform a DEA evaluation with the efficient centres of each cluster. Such centres are representative of the best managerial practices for each environmental condition. As was done previously, we used the CCR output oriented DEA model. The results of the evaluation of the four centres can be found in Table 6.

As observed in this Table, two centres were efficient, Angra dos Reis and Piraí. The least efficient of the four was Itaperuna.

Centre	Efficiency index (%)
Paracambi	96.43
Angra dos Reis	100.00
Piraí	100.00
Itaperuna	53.37

Table 6. Evaluation of the efficient centres

We can say that the efficient centres, thus, efficient clusters, are so because of them being regions with accelerated development based of tourism, oil and industry in general. The students in these clusters have no other options for undergraduate courses other than the long distance centres of CEDERJ.

In the city of Itaperuna is from cluster 4, which contains an underdeveloped region of the northwest Rio de Janeiro state. This region has an improving number of high schools but still of poor quality.

The first cluster, represented by Paracambi, is composed by very developed cities. These cities are close to *in situ* centres of high quality undergraduate courses. This condition nullifies the existence of potential good students, because the mostly preferred the *in situ* courses. This fact justifies its efficiency index in the group form by the efficient centres in each cluster.

The efficient centres are located in developed regions with good students but not with significant number of *in situ* courses.

Therefore, we may suppose that the differences among those centres are due only to the environmental aspects, as the centres have the best managerial practices in their clusters. So it is possible to use the efficiency in Table 6 to evaluate the environmental conditions of the cluster represented by each centre. The efficiencies will be used as a handicap factor for each cluster.

5.4 Overall evaluation

Taking into account the differences between clusters, we perform an homogenisation of the centres, to be able to compare all of them in one single cluster. This is done by multiplying the inputs (number of students enrolled in the course in the 2nd semester of 2005 and number of tutors) of each centre times the efficiency obtained by their representative in

Table 6. We consider that the efficiency index obtained by each representative centre in Table 6 acts has a handicap factor. This methodology is inspired by the sports handicapping system for competitions with disabled athletes (Percy & Scarf, 2008, Percy & Warner, 2009). The data used and the efficiency obtained using the CCR output oriented DEA model are shown in Table 7.

In this Table we can observed that, as expected, the efficiency centres in the original clusters are still efficient. We may now compare centres of different clusters. One of the lowest overall efficient is the centre of Campo Grande. This centre is located in a poor region of a reach city, Rio de Janeiro. This may indicate a problem in clustering this centre. Furthermore, there are a lot of *in situ* undergraduate courses surrounding Campo Grande. As explained before those factors are not favourable to a centre. The Petrópolis centre, with the lowest efficiency, is in a rich city and very close, less than one hour driving, of the major campus of the main Brazilian university. Due to the fact that distance education is not yet well know and the nearness of a prestigious university, many students prefer to travel to the *in situ* courses. The city of São Pedro da Aldeia is in a summer vacations region, many people living in Rio de Janeiro have a summer house in this city. Often, it occurs that some students obtain a vacancy in the centre of São Pedro da Aldeia, profiting from the fact of of having a house in the city and later they enrol in a *in situ* course in Rio de Janeiro, abandoning the long distance course in Sao Pedro de Aldeia. This explains the lower efficiency.

Centre	Input		Output	Efficiency Index (%)
	AI	NT	AF	
Volta Redonda	95,4643	9,64286	10	78.09
Paracambi	69,4286	6,75	9	100.00
Petrópolis	76,1786	7,71429	1	9.77
Angra dos Reis	60	6	8	100.00
São Pedro da Aldeia	20	5	1	9.82
Saquarema	62	6	4	50.00
Três Rios	60	8	3	31.30
Campo Grande	62	6	1	12.50
Macaé	29	6	3	47.44
Piraí	23	6	6	100.00
São Fidelis	61	6	2	25.00
Cantagalo	21,36	3,738	2	48.60
Itaperuna	19,224	3,738	4	100.00

Table 7. Homogenized data and overall efficiency index

We also perform an analysis of benchmarks for the inefficient centres. The benchmarks of an inefficient centre give the managerial guidelines to achieve the efficient levels in inputs or outputs. These are depicted in Table 8.
In this Table we may observed that the three cities originally in cluster 1, Volta Redonda, Paracambi and Petropolis, have benchmarks outside their own cluster. In the original cluster 3, Sao Fidelis is the only centre that has not at least one benchmark inside its own cluster. All the efficient centres except Paracambi, are their own benchmarks. This fact vindicates that Paracambi is a weakly efficient centre. This means that in an overall evaluation that the number of students that graduated in the first semester of 2009 can be improved in comparison to other efficient centres.

DMU	Benchmarks
Volta Redonda	Angra dos Reis; Piraí
Paracambi	Angra dos Reis
Petrópolis	Angra dos Reis; Piraí
Angra dos Reis	Angra dos Reis
São Pedro da Aldeia	Angra dos Reis; Piraí
Saquarema	Angra dos Reis
Três Rios	Angra dos Reis; Piraí
Campo Grande	Angra dos Reis
Macaé	Angra dos Reis; Piraí
Piraí	Piraí
São Fidelis	Angra dos Reis
Cantagalo	Angra dos Reis; Piraí
Itaperuna	Itaperuna

Table 8. Benchmarks in the overall efficiency evaluation

6. Final comments

The main objective of this chapter was to perform the evaluation of the centres of distance undergraduate Math courses of the CEDERJ. This evaluation was carried out using Data Envelopment Analysis. A total of thirteen centres were evaluated, these having environmental differences among them. They were divided in four cluster using Kohonen self-organized maps according to the size of the centres, level of the centres, socio economical characteristics and maturity of the centres proxies. In each cluster, we performed a DEA analysis obtaining exactly one efficient centre for each cluster. Comparing the clusters we conclude that centres in very poor or very rich regions will probably have low efficiency.

We also performed an homogenisation of the centres in order to obtain and overall evaluation and a benchmark analysis. We observed that the majority of the centres have benchmarks outside their own cluster. The fact that a large number of centres have very little efficiency may indicate that we must refine the clustering process. A variable that seems to be important and may be used in future works is the distance of the centre to major *campi* of *in situ* courses universities. The distance between two distance centres may also be considered for the clustering process in future works.
It may also be useful to perform a time window analysis of the centres.
It worth noting that São Fidelis and Campo Grande are each one in a single cluster for almost all the Kohonen maps configurations. It only in the configuration used they are clustered with other centres. This may indicated that São Fidelis and Campo Grande have been under evaluated in this study. In the future we may study a new process to perform a fair evaluation to those two centres.

7. Acknowledgments

We would like to thanks FAPERj and CNPq for their financial support.

8. References

Abbott, M. & Doucouliagos, C. (2003). The efficiency of Australian universities: A data envelopment analysis. *Economics of Education Review,* Vol. 22**,** No. 1**,** pp. 89-97, 0272-7757.

Abramo, G.; D'Angelo, C. A. & Pugini, F. (2008). The measurement of italian universities' research productivity by a non parametric-bibliometric methodology. *Scientometrics,* Vol. 76**,** No. 2**,** pp. 225-244, 0138-9130.

Andrade, F. V. S.; Brandão, L. C. & Soares de Mello, J. C. C. B. (2009). Avaliação de um curso de matemática à distância com modelos DEA e seleção de variáveis. *Relatórios de Pesquisa em Engenharia de Produção da UFF,* Vol. 9**,** pp. 10, 1678-2399.

Banker, R. D.; Charnes, A. & Cooper, W. W. (1984). Some models for estimating technical scale inefficiencies in data envelopment analysis. *Management Science,* Vol. 30**,** No. 9**,** pp. 1078-1092, 0025-1909

Biondi Neto, L.; Coelho, P. H. G.; Soares De Mello, J. C. C. B. & Angulo Meza, L. (2007). Self-organizing maps for classification of the rio de janeiro state cities based on electrical energy consumption. *ICEIS 2007 - 9th International Conference on Enterprise Information Systems, Proceedings,* pp. 447-450, june, 2007, Funchal.

Biondi Neto, L.; Lins, M. P. E.; Gomes, E. G.; Soares de Mello, J. C. C. B. & Oliveira, F. S. (2004). Neural data envelopment analysis: a simulation. *International Journal of Industrial Engineering,* Vol. 11**,** pp. 14-24, 1072-4761.

Bishop, C. M. (1995). *Neural networks for pattern recognition,* Oxford University Press, 0-19-853864-2, New York.

Çelebi, D. & Bayraktar, D. (2008). An integrated neural network and data envelopment analysis for supplier evaluation under incomplete information. *Expert Systems with Applications,* Vol. 35**,** No. 4**,** pp. 1698-1710, 0957-4174.

Charnes, A.; Cooper, W. W. & Rhodes, E. (1978). Measuring the efficiency of decision-making units. *European Journal of Operational Research,* Vol. 2**,** pp. 429-444, 0377-2217.

Churilov, L. & Flitman, A. (2006). Towards fair ranking of olympics achievements: The case of Sydney 2000. *Computers and Operations Research,* Vol. 33**,** No. 7**,** pp. 2057-2082, 0305-0548.

Cooper, C. & Burns, A. (2007). Kohonen self-organizing feature maps as a means to benchmark college and university websites. *Journal of Science Education and Technology,* Vol. 16**,** No. 3**,** pp. 203-211, 1059-0145 (print version), 1573-1839 (electronic version).

Cooper, W. W.; Seiford, L. & Tone, K. (2000). Data envelopment analysis: A comprehensive text with models, applications, references and DEA-solver software, Kluwer, 0387452818, Boston.

Dyson, R. G.; Allen, R.; Camanho, A. S.; Podinovski, V. V.; Sarrico, C. S. & Shale, E. A. (2001). Pitfalls and protocols in DEA. *European Journal of Operational Research,* Vol. 132**,** No. 2**,** pp. 245-259, 0377-2217.

Emrouznejad, A. & Shale, E. (2009). A combined neural network and DEA for measuring efficiency of large scale datasets. *Computers and Industrial Engineering,* Vol. 56**,** No. 1**,** pp. 249-254, 0360-8352.

Gomes Junior, S. F.; Soares de Mello, J. C. C. B. & Soares de Mello, M. H. C. (2008). Utilização do método de Copeland para avaliação dos pólos regionais do CEDERJ. *Rio's international journal on sciences of industrial and systems engineering and management,* Vol. 2**,** No. 4**,** pp. 87-98, 1982-6443.

Haykin, S. (1999). *Neural networks: a comprehensive foundation,* Prentice Hall, 0132733501, New Jersey.

Kiang, M. Y.; Fisher, D. M.; Chen, J. C. V.; Fisher, S. A. & Chi, R. T. (2009). The application of SOM as a decision support tool to identify AACSB peer schools. *Decision Support Systems,* Vol. 47**,** No. 1**,** pp. 51-59, 0167-9236.

Kohonen, T. (2001). *Self-organizing maps,* Springer-Verlag, 3540679219, Berlin.

Menezes, E. P. (2007). A espacialidade e a temporalidade da educação a distância: O caso do CEDERJ/CECIERJ. *13º Congresso Internacional de Educação a Distância*, september, 2007, Curitiba.

Michie, D.; Spiegelhalter, D. J. & Taylor, C. C. (1994). *Machine learning, neural and statistical classification,* Ellis Horwood, 013106360X, Chichester.

Mitra, P.; Murthy, C. A. & Pal, S. K. (2002). Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 24**,** No. 3**,** pp. 301-312

Percy, D. F. & Scarf, P. A. (2008). On the development of decision rules for bar quiz handicapping. *Journal of the Operational Research Society,* Vol. 59**,** No. 10**,** pp. 1406-1414, 0160-5682.

Percy, D. F. & Warner, D. B. (2009). Evaluating relative performances in disabled sports competitions. *IMA Journal Management Mathematics,* Vol. 20**,** No. 2**,** pp. 185-199, 1471-6798 (on line), 1471-678X (print).

Roy, B. & Bouyssou, D. (1993). *Aide multicritère à la décision: méthods et cas,* Economica, Paris.

Samoilenko, S. & Osei-Bryson, K. M. (2010). Determining sources of relative inefficiency in heterogeneous samples: Methodology using Cluster Analysis, DEA and Neural Networks. *European Journal of Operational Research,* Vol. 206**,** No. 2**,** pp. 479-487, 0377-2217.

Seiford, L. M. (1996). Data envelopment analysis: The evolution of the state of the art (1978-1995). *Journal of Productivity Analysis,* Vol. 7**,** No. 2-3**,** pp. 99-137

Sharma, M. J. & Yu, S. J. (2009). Performance based stratification and clustering for benchmarking of container terminals. *Expert Systems with Applications,* Vol. 36**,** No. 3 PART 1**,** pp. 5016-5022, 0957-4174.

Soares de Mello, J. C. C. B.; Gomes, E. G.; Angulo-Meza, L.; Soares de Mello, M. H. C. & Soares de Mello, A. J. R. (2006). Engineering Post-Graduate Programmes: A Quality and Productivity Analysis. *Studies in Educational Evaluation,* Vol. 32**,** pp. 136-152, 0191-491X.

Soares de Mello, M. H. C. (2003). Uma experiência presencial em EAD: o caso CEDERJ. . *XXVI CNMAC*, september, 2003, São José do Rio Preto.

Thanassoulis, E. (1996). Assessing the efficiency of schools with pupils of different ability using Data Envelopment Analysis. *Journal of the Operational Research Society,* Vol. 47**,** No. 1**,** pp. 84-97, 0160-5682.

Willshaw, D. J.; Buneman, O. P. & Longuet-Higgins, H. C. (1969). Non-holographic associative memory. *Nature,* Vol. 222**,** pp. 960-962, 0028-0836.

Willshaw, D. J. & Von der Malsburg, C. (1976). How patterned neural connections can be set up by self-organization. *Proceedings of the Royal Society of London Series B,* Vol. 194**,** pp. 431-445, 1471-2954.

5

Self-Organizing Maps Infusion with Data Envelopment Analysis

Mithun J. Sharma[1] and Yu Song Jin[2]
[1]*Dibrugarh University*
[2]*Korea Maritime University*
[1]*India*
[2]*Republic of Korea*

1. Introduction

This chapter presents work on the use of an artificial intelligence technique to cluster stratified samples of container terminals derived from Data Envelopment Analysis (DEA). This technique is Kohonen's self-organizing map (SOM; (Kohonen, 1995)). Data envelopment analysis measures the relative efficiency of comparable entities called Decision Making Units (DMUs) essentially performing the same task using similar multiple inputs to produce similar multiple outputs ((Charnes et al., 1978)). The purpose of DEA is to empirically estimate the so-called efficient frontier based on the set of available DMUs. DEA provides the user with information about the efficient and inefficient units, as well as the efficiency scores and reference sets for inefficient units. The results of the DEA analysis, especially the efficiency scores, are used in practical applications as performance indicators.

There are many problems associated with applying the DEA in some applications. One problem is that the improvement projection for inefficient units in DEA analysis is concrete relative to its efficiency score. This means, in DEA, relative performance of any DMU can be contrasted only to the efficient DMUs that register unit efficiency score. There is no influence on the performance of efficient DMUs by presence or absence of inefficient DMUs. Therefore, the classical DEA does not actually provide a direct means to rank DMUs based on their relative degrees of efficiency or inefficiency ((Sharma & Yu, 2010)).

The second problem is that the DEA models assume that all DMUs are homogenous and identical in their operations ((Seiford, 1994)). Since various applications have heterogeneous DMUs and there is a high request to evaluate these applications under the DEA due to its acceptance as a performance measurement in different kind of business, we have to modify the DEA to work with these applications. If the heterogeneous DMUs are assessed by DEA without any modifications, the DEA yields a biased performance scores and inaccurate analyses. For example, the resources (land, equipment, and labor) of container terminals varies across the world, which requires to be evaluated in term of its common input characterstics. An essential requirement in analyzing these container terminals is to build a fair referencing system for each container terminal to manage and provide a solid plan that improves all inefficient terminals and supports all efficient terminals. This system can not be assessed under the standard DEA due to the non-homogenous nature of these container terminals in terms of their operations, different standards of equipments, infrastructure, and variety in quay length and area size. These factors will yield unfair benchmarking evaluation

if we apply the standard DEA. In order to conform to the homogeneity assumption, we use learning network clustering (SOM) procedure to minimize total dissimilarity. Nor is the SOM approach the only neural network approach which could be used, however, we find the Kohonen approach to be highly effective method.

2. Algorithm

The DEA tools which are necessary to appreciate the method are described in detail in ((Cooper et al., 2004);(Sharma & Yu, 2010)). There it is demonstrated how one can use these tools to measure efficiency and stratify samples for relative attractiveness to identify competitors with level wise target improvement. Here in this chapter, a new algorithm is provided to perform the DEA computation in non-homogenous DMUs by introducing the SOM-based DEA technique. In the proposed method, some of the non-homogenous DMUs are classified into seperate groups for appropriate benchmarking. Clustering stratified DMUs obtained from DEA using SOM is divided into two steps. The first step is to train the SOM against the DMUs as a training data set. The second one is to map input DMUs to output DMU clusters. The algorithm in the subsequent subsection achieves this objective.

2.1 SOM-based DEA

Assume there are n DMUs, each with m inputs and s outputs. We define the set of all DMUs as $J^1, J^1 = DMU_j, j = 1,...,n$ and the set of efficient DMUs in J^1 as E^1. Then the sequences of J^1 and $E^1$are defined interactively as $J^{l+1} = J^l - E^l$ where $E^l = DMU_p \in J^l | \phi_p^l = l$, and ϕ_p^l is the optimal value to the following linear programming problem:

$$max_{\lambda_i,\phi} \phi_p^l = \phi \tag{1}$$

$$s.t. \sum_{i \in F(J^l)} \lambda_i x_{ji} - x_{jp} \leq 0 \forall j \tag{2}$$

$$\sum_{i \in F(J^l)} \lambda_i y_{ki} - \phi y_{kp} \geq 0 \forall k \tag{3}$$

$$\lambda_i \geq 0, i \in F(J^l) \tag{4}$$

where $k = 1$ to s, $j = 1$ to m, $i = 1$ to n, y_{ki} = amount of output k produced by DMU_{i^*}; x_{jp} = input vector of DMU_p, x_{ji} = amount of input j utilized by DMU_i; y_{kp} = output vector of DMU_p. $i \in F(J^l)$ in other words $DMU_i \in J^l$, i.e. $F(.)$ represents the correspondence from a DMU set to the corresponding subscript index set.

The following algorithm accomplishes subsequent stratum.

Step 1: Set $l = 1$. Evaluate the entire set of DMUs, J^l, to obtain the set, E^1, of first-level frontier DMUs (which is equivalent to classical CCR DEA model), i.e. when $l = 1$, the procedure runs a complete envelopment model on all n DMUs and E^1 consists of all of the DMUs on the resulting overall best-practice efficient frontier.

Step 2: Exclude the frontier DMUs from future DEA runs and set $J^{l+1} = J^l - E^l$

Step 3: If $J^{l+1} = 3E^{l+l}$, then stop. Otherwise, evaluate the remaining subset of inefficient DMUs, J^{l+1}, to obtain the new best-practice frontier E^{l+1}.

Stopping Rule: The algorithm stops when $J^{l+1} = 3E^{l+l}$.

The training data is a set of all DMUs, without output variables but includes the class each DMU belongs to, $J = \{DMU_1, DUM_2, ..., DMU_n\}$ of already classified samples. Each sample

$DMUj = \{x_1, x_2, ..., x_m\}$ is a vector where $x_1, x_2, ..., x_m$ represent input variables of the sample. The training data is augmented with a vector $C = \{E_1, E_2, ..., E_{l+1}\}$ where $E_1, E_2, ..., E_{l+1}$ represent the class each sample belongs to. The SOM uses a set of neurons, often arranged in a 2D rectangular or hexagonal grid, to form a discrete topological mapping of an input space, $X \in \Re^n$. At the start of the learning, all the weights $\{w_{r1}, w_{r2}, ..., w_{rm}\}$ are initialised to small random numbers. w_{ri} is the weight vector associated to neuron i and is a vector of the same dimension, n, of the input. m is the total number of neurons. $\mathbf{r}i$ is the location vector of neuron i on the grid. Then the algorithm repeats the following steps.

– At each time t, present an input, $x(t)$, select the winner,

$$v(t) = \arg\min_{k \in \Omega} \|X(t) - W_k(t)\| \tag{5}$$

– Updating the weight of winners and its neighbours,

$$\Delta W_k(t) = \alpha(t\eta(v,k,t))\,[X(t) - W_v(t)] \tag{6}$$

– Repeat until the map converges,

where $\eta(v,k,t)$ is the neighborhood function and Ω is the set of neuron indexes.

3. Application

The proposed SOM-based DEA algorithm was applied to container port industry. Data was collected for 70 container terminals from relevant data sources like *Containerization International Year Book*, *The Drewry Annual Container Market Review and Forecast* and specific field studies of container port. The input and output data were selected from a comprehensive set of literature review ((Hayuth & Roll, 1993);(Park & De, 1993); (Tongzon, 2001);(Barros & Athanassiou, 2004)). The statistics related to the sample are shown in table 1.

Inputs and outputs used in DEA must be measurable, but they need not be measured in the same units. The number of variables used in the DEA formula has direct effect on any particular piece of data. (Szezepura et al., 1992) argue that the number of variables should be limited to the maximum extent. In general, the number of test units should be considerably greater than the total number of variables.

A number of container terminals constitute a container port. For one to one comparison (Song & Cullinane, 2003) we have investigated container terminals selected from underdeveloped to transition economies to developed economies that include large, medium and small container terminals. The following features/measures are chosen as inputs: (1) quay length (meters); (2) terminal area (sq. meters); (3) quay cranes (number); (4) transfer cranes (number); (5) reach stackers (number) and (6) straddle carriers (number). On the other hand, container throughput (TEU)[1] is the most appropriate and analytically tractable indicator of the effectiveness of the production of a port. Almost all previous studies treat it as an output variable, because it closely relates to the need for cargo-related facilities and services and is the primary basis upon which container ports are compared, especially in assessing their relative size, or activity levels. Therefore, throughput is chosen as an output variable.

[1]TEU is the abbreviation for Twenty feet Equivalent Unit, referring to the most standard size for a container of 20 ft in length.

	Throughput	QC	TC	SC	RSC	QL	TA
Mean	882143.414	9	14.185	12.985	80.51	1105.042	517876.1
Std. error	98748.9083	0.666	2.084	2.951	0.852	82.505	48321.5
Median	573,049	8	9	0	7	927.5	350,000
Mode	N/A	6	0	0	2	600	300,000
Std. deviation	826192.642	5.579	17.442	24.692	7.172	690.286	404286.9
Sample variance	6.082*1011	31.130	304.24	609.72	51.441	476495.52	1.63*10
Kurtosis	4.269	1.304	5.114	2.222	2.426	4.064	0.931
Skewness	1.960	1.249	2.004	1.846	1.314	1.800	1.265
Range	3,901,632	24	90	94	36	3646	1,648,000
Minimum	98,368	2	0	0	0	300	20,000
Maximum	4,000,000	26	90	94	36	3946	1,668,000
Sum	61,750,039	630	993	909	596	77,353	36,251,334
Count	70	70	70	70	70	70	70
Confidence level (95%)	196998.679	1.330	4.159	5.887	1.710	164.59	96398.82

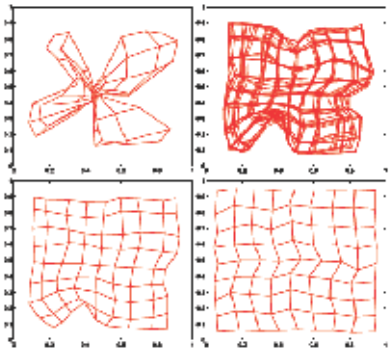
QC:Quay Cranes;TC:Transfer Cranes; SC:Straddle Carrier;RSC:Reach Stacker;QL:Quay Length;TA:Terminal Area

Table 1. Descriptive statistics of container terminal data

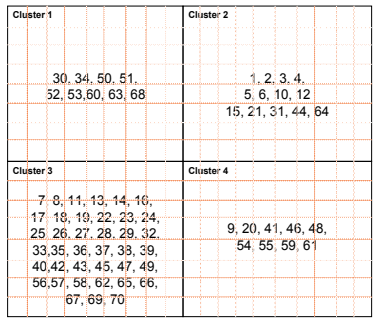
The input variable quay crane is a vital piece of equipment in the production process where it transfers the cargo from a container shipping line across a quay to the shore. This production process fundamentally decides the efficiency of a port, and is vital to its competitive position. As a storage area, the container terminalarea acts as a buffer between sea and inland transportation or transshipment. The size of a ship is very frequently thousands of times the size of the land vehicles that carry the cargo to and from the port. As such, the use of such storage space is normally inevitable. The main pieces of equipment used within a terminal area are the transfer cranes, reach stackers and straddle carriers. (Dowd & Leschine, 1978) argue that the production of a container terminal depends on the efficient use of labor, land and equipment. The measurement of terminal production, therefore, is a means of quantifying efficiency in the utilization of these three resources. Given the characteristics of container port production, the total quay length and the terminal area are the most suitable proxies for the land factor input and the number of quay cranes, the number of transfer cranes, the number of reach stackers and the number of straddle carriers are the most suitable proxies for the equipment factor input. Measures of these variables should be incorporated into the models as input variables

Usually traditional DEA method if employed to obtain efficiency measure for 70 container terminals, would give 18 efficient container terminals with unit score and 52 inefficient terminals with scores < 1. All these inefficient terminals are projected to the 18 efficient terminals for reference irrespective of their differences in efficiency scores. For example, a container terminal with a score of 0.07 is projected to the frontier. It is difficult for this inefficient container terminal to improve its performance by benchmarking a container terminal on the frontier due to huge performance gap along with the differences in their input characteristics. Therefore, it is important to have attainable benchmark target for improvement keeping in view the homogeneity assumption. The partitioning analysis is useful to provide an appropriate benchmark target for poor performers. By using the SOM-based DEA algorithm described in sub-section 3.1, we obtained five levels of efficient frontiers and four clusters. The efficient frontiers are as follows:

$E^1 = \left\{ DMU_j | j = 19;20;29;34;36;39;41;42;45;46;53;54;57;59;60;63;67;69 \right\}$



(a) Two-dimensional Kohonen network



(b) Final clustering of container terminal data

Fig. 1. (a) shows the flattening of a two-dimensional Kohonen network in a quadratic input space. The four diagrams display the state of the network after 100, 1000, 5000, and 10000 iterations.(b) Four clusters of container terminal data at the final state of 1000 iterations.

$$E^2 = \left\{ DMU_j | j = 4;5;8;12;13;15;16;18;26;31;32;35;40;44;47;48;51;58;61;65;66;70 \right\}$$
$$E^3 = \left\{ DMU_j | j = 1;2;7;10;11;17;28;30;37;38;52 \right\}$$
$$E^4 = \left\{ DMU_j | j = 3;9;23;27;43;50;55;62 \right\}$$
$$E^5 = \left\{ DMU_j | j = 6;14;21;22;24;25;33;49;56;64;68 \right\}$$

The proposed SOM-based DEA algorithm produced five stratum of DMUs based on their efficiency level and four clusters as shown in figure 1(b) based on their input traits. Figure 1(a) shows the flattening of a two-dimensional Kohonen network in a quadratic input space. The four diagrams display the state of the network after 100, 1000, 5000, and 10000 iterations.After organizing the DMUs based on our proposed procedure, the projection of inefficient terminals were determined. The inefficient DMUs in the lowest stratum i.e. E^5 benchmarks their immediate upper stratum with similar input features. Same is the case with the DMUs in E^4, E^3, and E^2 belonging to separate clusters.

The application of the model reveals some interesting insight for improving poorly performing terminals. For example, let us consider DMUs 6, 21, 14, and 22 of E^5. DMUs 6 and 21 falls in cluster 2 whereas DMUs 14, and 22 are in cluster 3. Traditional DEA techinique will refer these inefficient terminals to the efficient frontier of 1 irrespective of difference in the efficiency scores. E.g., DMU 6 gets an efficiency score of 0.07 and for improvement, this particular DMU is referred to DMU 9 with efficiency score 1. However, DMU 6 varies from DMU 9 in various aspects of resource mix. Referring to DMU 9 for improvement is unrealistic due to the presence of heterogeneity in their input traits. Using SOM-based DEA technique, DMU 6 is referred to DMU 3 in efficient stratum E^4 with similar input mix (they belong to cluster 2). In the same way DMU 21 in E^5 is also referred to DMU 3 in E^4 as it falls in cluster 2. Whereas DMUs 14 and 22 of E^5 is referred to DMUs 23, 27, 43, and 62 of E^4 as they belong to cluster 3. Thus SOM-based DEA algorithm significantly enhances the capability of traditional DEA tool in prescribing realistic reference points for inefficient DMUs which otherwise is not possible with traditional DEA alone.

4. Conclusion

The benchmarking and improvement projection using the conventional DEA procedure is not desirable because the inefficient DMUs are projected to the efficient frontier ignoring the differences in the efficiency score and input traits. Upon analysis it was found that the efficiency score of DMUs ranged from 4.75% to 100% out of which 18 container terminals are found to be efficient with a score of 1. The 57 inefficient terminals had to refer these limited efficient terminals for improvement. In general, the benchmarking is done to improve the performance of DMUs. But a DMU with low score of 4.75% cannot make direct improvement projections to DMU of score 100%, it needs a DMU with reasonably equivalent characteristics and capacity for benchmarking and improvement.
In this chapter we remedied this limitation of conventional DEA by our proposed SOM-based DEA algorithm. Using the proposed algorithm we could organize the inefficient units into multiple efficient stratum ($E^2 = 22$, $E^3 = 11$, $E^4 = 8$, $E^5 = 11$) thereby providing a level-wise improvement path for poor performing DMUs and also context for evaluation with 4 clusters that conform to the homogeneity assumption, thereby minimizing total dissimilarity in the benchmarking procedure.

5. References

Barros, C.P., Athanassiou, M. (2004). Efficiency in European seaports with DEA: Evidence from Greece and Portugal. *Maritime Economics and Logistics*, Vol. 6, 122 140.

Cooper, W. W., Seiford, L. M., & Zhu, Joe (2004). Name of paper, In: *Handbook on data envelopment analysis*, Kluwer.

Charnes, A., Cooper, W.W., Rhodes, E. (1978). Measuring the efficiency of decision making unit. *European Journal of Operational Research*, Vol. 2, 429 444.

Dowd, T.J., Leschine, T.M. (1990). Container terminal productivity: A perspective. Maritime Policy and Management 17, 107 111.

Hayuth, Y., & Roll, Y.(1993). Port performance comparison applying data envelopment analysis (DEA). *Maritime Policy and Management*, Vol. 20, 153 161.

Kohonen T., 1995. *Self Organizing Maps*, Springer-Verlag, Berlin.

Park, R-K.,& De, P.(2004). An alternative approach to efficiency measurement of seaports. *Maritime Economics and Logistics*, Vol. 6, 53 69.

Seiford, L. (1994). A DEA bibliography in data envelopment analysis (19781992). DEA bibliography. Merest: University of Massachusetts.

Song, D., Cullinane, K. (2003). A stochastic frontier model of the productive efficiency of Korean container terminals. *Applied Economics*, Vol. 35, 251 267.

Sharma, M.J., Yu, S.J. (2009). Performance based stratification and clustering for benchmarking of container terminals. *Expert Systems with Applications*, Vol. 36, 5016 5022.

Sharma, M.J., Yu, S.J. (2010). Benchmark optimization and attribute identification for improvement of container terminals. *European Journal of Operational Research*, Vol. 201, 568 580.

Szezepura, A., Davis, C., Fletcher, J., Bousoffiane, A.(1992). Applied data envelopment analysis. In: Health Care: The Relative Efficiency of NHS General Practices. Warwick Business School Research Bureau, Coventry.

Tongzon, J.(2001). Efficiency measurement of selected Australian and other international ports using data envelopment analysis. *Transportation Research Part A*, Vol. 35, 113 128.

Part 3

The Study of Multi-media and Web-based Contents

6

A Speech Recognition System for Embedded Applications Using the SOM and TS-SOM Networks

Amauri H. Souza Júnior[1], Guilherme A. Barreto[1] and Antonio T. Varela[2]
[1]*Federal University of Ceará, Department of Teleinformatics Engineering*
[2]*Federal Institute of Education, Science and Technology of Ceará*
[1,2]*Brazil*

1. Introduction

The self-organizing map (SOM) (Kohonen, 1982) is one of the most important neural network architecture. Since its invention it has been applied to so many areas of Science and Engineering that it is virtually impossible to list all the applications available to date (van Hulle, 2010; Yin, 2008). In most of these applications, such as image compression (Amerijckx et al., 1998), time series prediction (Guillén et al., 2010; Lendasse et al., 2002), control systems (Cho et al., 2006; Barreto & Araújo, 2004), novelty detection (Frota et al., 2007), speech recognition and modeling (Gas et al., 2005), robotics (Barreto et al., 2003) and bioinformatics (Martin et al., 2008), the SOM is designed to be used by systems whose computational resources (e.g. memory space and CPU speed) are fully available. However, in applications where such resources are limited (e.g. embedded software systems, such as mobile phones), the SOM is rarely used, especially due to the cost of the best-matching unit (BMU) search (Sagheer et al., 2006).

Essentially, the process of developing automatic speech recognition (ASR) systems is a challenging tasks due to many factors, such as variability of speaker accents, level of background noise, and large quantity of phonemes or words to deal with, voice coding and parameterization, among others. Concerning the development of ASR applications to mobile phones, to all the aforementioned problems, others are added, such as battery consumption requirements and low microphone quality.

Despite those difficulties, with the significant growth of the information processing capacity of mobile phones, they are being used to perform tasks previously carried out only on personal computers. However, the standard user interface still limits their usability, since conventional keyboards are becoming smaller and smaller. A natural way to handle this new demand of embedded applications is through speech/voice commands. Since the neural phonetic typewriter (Kohonen, 1988), the SOM has been used in a standalone fashion for speech coding and recognition (see Kohonen, 2001, pp. 360-362). Hybrid architectures, such as SOM with MultiLayer Perceptrons (SOM-MLP) and SOM with Hidden Markov Models (SOM-HMM), have also been proposed (Gas et al., 2005; Somervuo, 2000). More specifically, studies involving speech recognition in mobile devices systems include those by Olsen et al. (2008); Alhonen et al. (2007) and Varga & Kiss (2008).

It is worth noticing that Portuguese is the eighth, perhaps, the seventh most spoken language worldwide and the third among the Western countries, after English and Spanish. Despite

that, few automatic speech recognition (ASR) systems, specially commercially available ones, have been developed and it is available worldwide for the Portuguese language. This scenario is particularly true for the Brazilian variant of the Portuguese language, due its large amount of accent variation within the country. Scanzio et al. (2010), for example, report experiments with a neural network based speech recognition system and include tests with the Brazilian Portuguese language. Their work is focused on a hardware-oriented implementation of the MLP network.

In this context, the current paper addresses the application of self-organizing maps to the Brazilian Portuguese isolated spoken word recognition in embedded systems. For this purpose, we are particularly interested in evaluating several software strategies to speedup SOM computations in order to foster its use in real-time applications. The end-user application is a speaker-independent voice-driven software calculator which is embedded in the Nokia N95 smartphone.

The remainder of the paper is organized as follows. In Section 2 the SOM architecture and its learning process are described. In Section 3, the techniques used for reducing the SOM's computational cost are introduced. Section 4 presents the evaluation of SOM-based methods and other classical algorithms for speech recognition. The paper is concluded in Section 5.

2. Evaluated architectures

2.1 The basic SOM algorithm

The *Self-Organizing Map* (SOM) (Kohonen, 1982) is a well-known unsupervised competitive learning algorithm that learns, from examples, a projection from a high-dimensional continuous input space $\mathcal{X}$ onto a low-dimensional discrete space (lattice) $\mathcal{A}$ of M neurons which are arranged in fixed topological forms, e.g., as a 2-dimensional rectangular array. The map $i^*(\mathbf{x}) : \mathcal{X} \to \mathcal{A}$, defined by the weight matrix $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_q), \mathbf{w}_i \in \mathcal{X}$, assigns to each input vector $\mathbf{x} \in \mathcal{X}$ a winning neuron $i^* \in \mathcal{A}$. Using Euclidean distance, one of the simplest strategies to find the winning neuron, $i^*(t)$, is given by

$$i^*(t) = \arg\min_{\forall i} \|\mathbf{x}(t) - \mathbf{w}_i(t)\|_2, \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^p$ denotes the current input vector, $\mathbf{w}_i(t) \in \mathbb{R}^p$ is the weight vector of neuron i, and t symbolizes the time steps associated with the iterations of the algorithm.

Adjustment of the weight vectors of the winning neuron (also called best matching unit - BMU) and of those neurons belonging to its neighborhood:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t) h(i^*, i; t)[\mathbf{x}(t) - \mathbf{w}_i(t)] \tag{2}$$

where $0 < \alpha(t) < 1$ is the learning rate and $h(i^*, i; t)$ is usually the Gaussian neighborhood function:

$$h(i^*, i; t) = \exp\left(-\frac{\|\mathbf{r}_i(t) - \mathbf{r}_{i^*}(t)\|^2}{2\sigma^2(t)}\right) \tag{3}$$

where $\mathbf{r}_i(t)$ and $\mathbf{r}_{i^*}(t)$ are the coordinates of neurons i and i^* in a predefined output array respectively, and $\sigma(t) > 0$ defines the radius of the neighborhood function at time t.

The variables $\alpha(t)$ and $\sigma(t)$ should both decay with time to guarantee convergence of the weight vectors to stable steady states. The operations defined in Eqs. (1) and (2) are repeated for a certain numbers of iterations until a steady state of global ordering of the weight vectors has been achieved.

	BMU search	updating phase	param. adaptation	total
multi.	$NnM\epsilon$	$NM\epsilon(4+k+n)$	2ϵ	$\epsilon[2+NM(2n+k+4)]$
divi.	-	$NM\epsilon$	ϵ	$\epsilon(NM+1)$
adi.	$NM\epsilon(n-1)$	$NM\epsilon(k-1+n)$	-	$NM\epsilon(2n+k-2)$
sub.	$NnM\epsilon$	$NM\epsilon(k+n)$	-	$NM\epsilon(2n+k)$
comp.	$(M-1)N\epsilon$	-	-	$N\epsilon(M-1)$
exp.	-	$NM\epsilon$	2ϵ	$\epsilon(2+NM)$
total	$N\epsilon(3Mn-1)$	$MN\epsilon(3n+3k+5)$	5ϵ	

Table 1. Number of operations in the basic SOM algorithm.

As mentioned in the introduction, the SOM has been widely used in engineering and data analysis but rarely used in real-time applications. The main reason is the computational cost of the SOM, especially with respect to the search for the BMU and the weight updating phase. Table 1 reports the number of atomic operations (multiplication, division, subtraction, comparison and exponentiation) in the basic training SOM algorithm with a Gaussian neighborhood function. In this table, N denotes the number of input vectors, M is the number of neurons, n is the dimension of input vectors, k represents the dimension of the map, and ϵ is the number of iterations.
If the exhaustive BMU search is performed, the algorithm requires the computation of the distances between the input vector and the weight vector for all neurons in the map (see Eq. (1)). The updating phase is the most computationally demanding phase when a Gaussian function is used (for being an asymptotic function, all neurons have their weights updated). Thus, the SOM algorithm has linear complexity, $O(M)$, in numbers of neurons.

2.2 Tree structured SOM

The tree structured self-organizing map (TS-SOM) was proposed by Koikkalainen & Oja (1990) as a fast alternative to the SOM training/testing procedures. The tree search reduces the time complexity of the BMU search from $O(M)$ to $O(\log M)$. The TS-SOM is comprised of several layers of SOM networks with different resolutions, where the neurons in the same layer are laterally connected. In addition, there are hierarchical connections among neurons of different layers. Usually, each neuron is related to 2^D neurons in the next layer, where D is the dimension of the grid. In the training procedure the neurons are adapted layer by layer.
The TS-SOM feature that differs the most from the basic SOM algorithm is the BMU search. In the conventional tree search, the search starts from the root node and proceeds to look for the most similar node to the input pattern in the next layer. In the TS-SOM search, the winner of the next layer is selected from a larger set of nodes: the BMU is the best matching child node among the parent nodes in the immediate neighborhood. The inclusion of the children nodes of the nearest neighbors is computationally light because it does not depend on the number of neurons in the layer. The updating step of the TS-SOM algorithm is similar to the basic algorithm: only the winning neuron and its immediate neighborhood are updated.

3. Reducing the computational costs of the SOM

Recently, several hardware-oriented fast implementations of neural network algorithms, including the SOM, using graphics processing units (GPU) have been proposed (Xiao et al., 2010; Scanzio et al., 2010; Oh & Jung, 2004), but we are particularly interested in speeding up

the SOM computations by simple and efficient software methods, to be described in the next sections.

Use Squared Euclidean Distances - By definition, textbook SOM algorithm requires the execution of the square root function in order to *correctly* compute euclidean distances. However, since we are not interested in the absolute values of the distances, but rather in their relative magnitudes, the square root function does not need to be executed. In this case, the squared euclidean distance can be computed, for example, through a dot product:

$$\|\mathbf{x}(t) - \mathbf{w}_i(t)\|^2 = (\mathbf{x}(t) - \mathbf{w}_i(t))^T(\mathbf{x}(t) - \mathbf{w}_i(t)). \quad (4)$$

Partial Distance Search - The Partial Distance Search (PDS) (Bei & Gray, 1985) is a heuristic method that involves a simple modification in the way distances are calculated. A pseudocode for the PDS technique is given below. During the computation of the accumulated distance sum, d, if the squared partial distance exceeds the smallest distance, d_{min}, to the nearest neighbor found so far, the computation is stopped. The additional time required by the evaluation of the stopping rule $(IF\ (d > d_{min}),\ THEN...$ is, on average, shorter than the time used in exhaustive search (Niskanen et al., 2002). It is reported that PDS reduces the computational cost of the BMU search by half or more (Bei & Gray, 1985).

Algorithm 3.1: PDS($\mathbf{x}$)

$i^* \leftarrow 1$
$d_{min} \leftarrow$ SQUAREDEUCLIDEANDISTANCE$(\mathbf{x}, \mathbf{w}_1)$
for $i \leftarrow 2$ **to** M
do
- $discard \leftarrow$ **false**
- $d \leftarrow 0$
- **for** $j \leftarrow 1$ **to** n
  - **do**
    - $aux \leftarrow \mathbf{x}(j) - \mathbf{w}_i(j)$
    - $d \leftarrow d + aux * aux$
    - **if** $(d > d_{min})$
      - **then**
        - $discard \leftarrow$ **true**
        - **exit**
- **if** $(discard =$ **false** $)$
  - **then**
    - $i^* \leftarrow i$
    - $d_{min} \leftarrow d$

return (i^*)

Shortcut Winner Search - The Shortcut Winner Search (SWS) was proposed by Kohonen (1997). This method requires a partially ordered map. Thus, the probability of the winning neuron for an input vector $\mathbf{x}$ being in the neighborhood of last winning neuron is high. Thereby, Kohonen (1997) recommends storing a pointer relating a vector $\mathbf{x}$ to the winner neuron $i^*(t)$, in the last iteration t. Then, in the iteration $t + 1$ the BMU search can be performed in the immediate neighborhood of the neuron $i^*(t)$, and only if a neuron most similar to $\mathbf{x}$ is found, the search continues in the new neighborhood and so on, until the winner is in the search center.

/um/ (1)	/dois/ (2)	/tres/ (3)
/quatro/ (4)	/cinco/ (5)	/seis/ (6)
/sete/ (7)	/oito/ (8)	/nove/ (9)
/zero/ (0)	/mais/ (plus)	/menos/ (minus)
/vezes/ (times)	/dividido/ (divided by)	/limpar/ (clear)
/voltar/ (back)	/resultado/ (result)	–

Table 2. Words recorded (in Brazilian Portuguese).

Neighborhood Functions - Rather than using the Gaussian neighborhood function, one can use the rectangular function since it does not require the computationally expensive exponential function. Let $R(t)$ be the winner's neighborhood radius in iteration t and $N_{i^*}(t)$ the set of neurons such that $\| \mathbf{r}_{i^*}(t) - \mathbf{r}_i(t) \|^2 < R^2(t)$. Then $h(i^*,i;t) = 1$ is valid for all neurons $i \in N_{i^*}(t)$ and $h(i^*,i;t) = 0$ otherwise. Another neighborhood function is the truncated Gaussian. In this case, values of the Gaussian functions are calculated only for the neurons within a distance range from the winner. The computational advantage is that a smaller number of neurons have their weights updated in comparison to the standard Gaussian neighborhood function.

4. Simulations and discussion

To evaluate the SOM-based algorithms, we recorded Brazilian Portuguese spoken words consisting of the basic mathematic operations and digits needed for the development of an embedded voice-driven software calculator. The data set includes speech data from 14 speakers. For each speaker, 51 utterances were acquired (17 different words repeated 3 times). Table 2 shows the selected words. In this table, the meaning of each word in English is shown within parentheses.

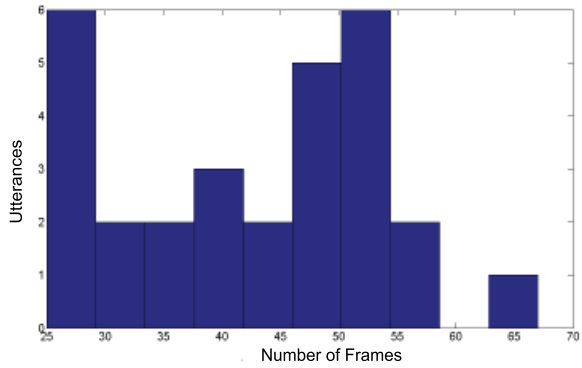
All the words were recorded in an indoor environment corrupted by air-conditioning noise. The involved speakers (11 men and 3 women) spoke freely, maintaining their respective accents. This fact contributed to a more difficult recognition tasks because even the same utterances had different durations after the endpoint detection. For example, Figure 1 shows the histogram of frame numbers of two words: /dividido/ in Figure 1(a) and /resultado/ in Figure 1(b).

In the front-end processing we use a 8 KHz sampling rate and 8 bits for the quantization of the signal amplitudes. Furthermore, we detected the endpoints by speech energy using the algorithm described in Wang et al. (2008).

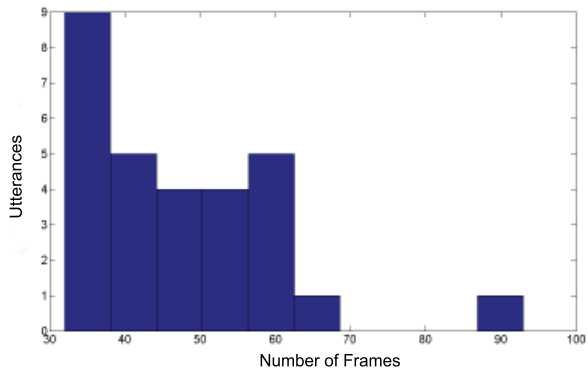
In the feature extraction step, each utterance is segmented into frames of 20 milliseconds. For each frame, a short-term Linear Predictive Coding (LPC) analysis is performed and then 10 cepstral coefficients are obtained from the LPC coefficients (Deller et al., 2000) by means of the following expression:

$$\widehat{c}_h(k) = \begin{cases} \log G, & k = 0, \\ \widehat{a}(k) + \sum_{i=1}^{P-1} (i/k)\widehat{c}_h(i)\widehat{a}(k-i), & k = 1,...,P, \end{cases} \quad (5)$$

where $\widehat{a}(i)$ are the LPC coefficients, $\widehat{c}_h(i)$ are the cepstral coefficients, G is a gain factor (associated with the AR model) estimated as $G = r_s(0) - \sum_{i=1}^{P} \widehat{a}(i) r_s(i)$ (Rabiner & Schafer, 1978) and $r_s(i)$ is the autocorrelation value at lag i, $i = 0,1,...,P$.



(a) Word /dividido/ (divided by).



(b) Word /resultado/ (result).

Fig. 1. Histograms of frame numbers.

We compared the SOM-based neural networks with supervised methods commonly used for speech recognition, such as the MLP network, the sequential K-means algorithm and the *Dynamic Time Warping* (DTW) technique (Sakoe & Chiba, 1978).
The approach used in the simulations of the SOM variants and the K-means algorithm is that described in Rabiner & Juang (1993), where each word category (e.g. /um/) is associated with a vector quantizer that maps all examples from this word category (see Figure 2). When a new utterance is presented, all the vector quantizers are evaluated and the one with the lowest quantization error, $E_q^{(i)}$, $i = 1,...,U = 17$, is the winner.
The SOM and TS-SOM networks used in the simulations had 10 input units and 256 neurons arranged in a 16×16 array. For the sake of comparison, the K-means algorithm also used $K = 256$ prototype vectors. It is worth highlighting that each VQ in the multiple vector quantizer approach described in Figure 2, be it a SOM-based one or the K-means algorithm, use 256 prototype vectors. Hence, the total number of prototypes used is $17 \times 256 = 4352$, demanding considerable computational efforts if the classifier is to be embedded in a mobile phone.
For the MLP network, we used 10 input units, 50 hidden neurons and 17 output neurons. For its training, we used the 1-out-of-N encoding for the target outputs, a learning rate of 0.01 and logistic activation functions for the hidden/output neurons. All the aforementioned

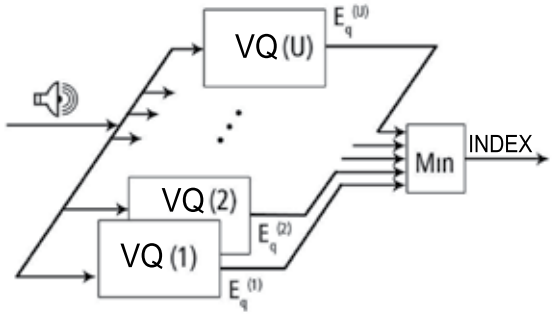


Fig. 2. Multiple vector quantizer approach for speech recognition as described in Rabiner & Juang (1993).

neural networks and the *K*-means algorithm were trained for 500 epochs and evaluated by the hold-out method for 20 independent training/testing runs. For training, it is used 80% of the total number of data vectors, randomly selected at each run, while the remaining 20% data vectors are used for testing purposes.

Table 3 reports the statistics of the performances of all evaluated algorithm, averaged over 20 independent training/testing runs. These simulations correspond to offline tests, executed on a personal computer, not in an embedded device. They were necessary in order to help in the choice of the classification algorithm to be embedded in the mobile phone. The offline tests were performed in a Dell personal computer, processor's clock of 2.33 GHz, 1 GB of RAM memory, in a Windows XP system and Java programming language. In addition, we used a standard (i.e. non-professional) desktop microphone to capture the utterances.

The results presented in Table 3 were gathered after comprehensive and exhaustive tests with different combinations of architectures and input variables, such as the amount of LPC/cepstral coefficients. Analyzing only the average recognition rates, one can see that the worst rate was achieved using the MLP network (81.1 %). With regard to the SOM, the *K*-means and the DTW algorithm, the achieved recognition rates were similar, with the best performances being achieved by the SOM:PDS and the DTW algorithm.

Algorithms	**Recognition rates (%)**				**Time (ms)**	
	mean	*max*	*min*	*std*	*training*	*test*
SOM original	87.7	92.9	82.9	3.22	2,911,873.47	8.14
SOM:SWS+PDS+Rectangular	85.8	92.2	79.4	4.41	2,275,220.35	4.48
SOM:PDS	89.6	92.9	85.1	2.39	2,860,639.27	3.97
SOM:PDS+Rectangular	88.2	92.9	84.4	2.80	2,275,061.98	4.07
SOM:PDS+TruncatedGauss	87.9	93.6	83.7	2.69	2,661,322.49	5.80
TS-SOM	82.5	88.6	75.9	3.69	3,310.01	7.71
MLP	81.1	86.5	72.3	4.09	93,642.47	0.21
KMeans	88.4	92.9	82.3	2.31	154,029.93	8.12
DTW	89.8	93.6	82.9	3.02	0.0012	3.266.43

Table 3. Performances of the evaluated classifiers.

	/0/	/1/	/2/	/3/	/4/	/5/	/6/	/7/	/8/	/9/	/+/	/-/	/*/	/÷/	/r/	/l/	/v/
/0/	**99.3**					0.7											
/1/		**96.5**	0.6		1.2	0.6				1.2							
/2/		0.6	**86.1**	0.6			2.9			1.2			8.7				
/3/			1.4	**83.9**			7.7						7.0				
/4/					**88.1**				1.7	9.0	0.6					0.6	
/5/						**96.2**	0.5						2.2			1.1	
/6/	1.8		1.2	13.9			**80.6**					1.2	1.2				
/7/	2.8		0.7		2.1	2.8	1.4	**87.9**		2.1							
/8/	1.2	6.1	3.0		0.6		1.8		**85.4**				0.6		0.6		0.6
/9/		3.5			0.6					**95.3**						0.6	
/+/							2.9			1.7	**94.9**		0.6				
/-/	0.6	2.5			0.6	3.2	0.6				0.6	**88.6**	3.2				
/*/			5.9	4.9			0.5					5.9	**82.7**				
/÷/	2.7					6.7								**89.9**			0.7
/r/									0.7						**98.7**		0.7
/l/		0.6													4.3	**93.8**	1.2
/v/																0.5	**99.5**

Table 4. Confusion matrix for the DTW algorithm.

Table 4 shows the confusion matrix for the DTW algorithm. On the one hand, the digit /0/ (zero) and the word /v/ (back) presented average successful recognition rates of 99.3% and 99.5%, respectively. On the other hand, most of the errors occurred for the digits /6/ and /3/. This can be partly explained if we note that the utterances of the words /tres/ (3) and /seis/ (6) are very similar in Portuguese, specially their final parts.

Despite the good performance the DTW algorithm, it is important to emphasize that the choice of the best classifier to be embedded in the smartphone should also take into account the required training and testing times. In what concern the online tests, there are two main approaches to embed a speech classifier in a mobile phone. The first performs the training phase off-line in a desktop and, once the classifier is trained, one just uploads the classifier's parameters to the phone. The second approach performs both the classifier training and testing in the phone itself.

From Table 3 one can see that the SOM:PDS does not satisfy the time restrictions to perform the online training phase, but it can be trained offline and their parameters then can be uploaded to the phone. Furthermore, although DTW had achieved good results, its testing time is too long, and then, it cannot be inserted into the embedded device. The option that achieved the best tradeoff between good performance and low processing times relied on the TS-SOM network. We also performed

In order to evaluate the feasibility of embedding the TS-SOM and SOM:PDS networks for real-time speech recognition purposes, we developed a speaker-independent voice-driven calculator on the Nokia N95 smartphone. This phone has a ARM11 processor with clock about 300 MHz. The application was developed using the JME (*Java Micro Edition*) framework. Some snapshots of the graphical interface developed for this embedded application are illustrated in Figure 3.

We defined two metrics to evaluate the TS-SOM and the SOM:PDS classifiers in terms of their processing times when running in the N95 phone, from the user point of view. The first one is the processing time that represents the time actually elapsed to classify an utterance, calculated by the mobile phone itself. The second one is the time perceived by user (response time), calculated by user with a chronometer. The tests were performed with four different words, two of them considered as long words, /*dividido*/ (divided by) and /*resultado*/ (result) and two of them considered short ones, /*um*/ (one) and /*dois*/ (two). The average values for the evaluated metrics are illustrated in Figure 4.

One can see from Figure 4 that while the processing times are all smaller than 1.25 seconds, the response time lasts from 2.10 to 3.25 seconds. The difference between these metrics occurs because a progress bar was created in the application in order to wait for the user utterance.

(a) Speech recording screen. (b) Main menu screen. (c) Calculator screen embedded in Nokia N95 smartphone.

Fig. 3. Snapshots of the voice-driven software calculator.

The pre-processing and recognition algorithms are performed only when this bar reaches its endpoint. Thus, there is fixed time which is wasted while the bar is moving.
Finally, the TS-SOM was evaluated in the online training task. The tests indicated an average time of 14 seconds of training, considering that 34 utterances (17 classes) were recorded on the mobile phone for a single speaker (the owner of the phone!). After this training period, the performance of the TS-SOM classifier was evaluated, reaching values similar to those achieved for the off-line tests (around 78-80%). It is worth noticing, that there is a natural degradation of the recognition performance of the classifier when it is embedded in the smartphone, since the

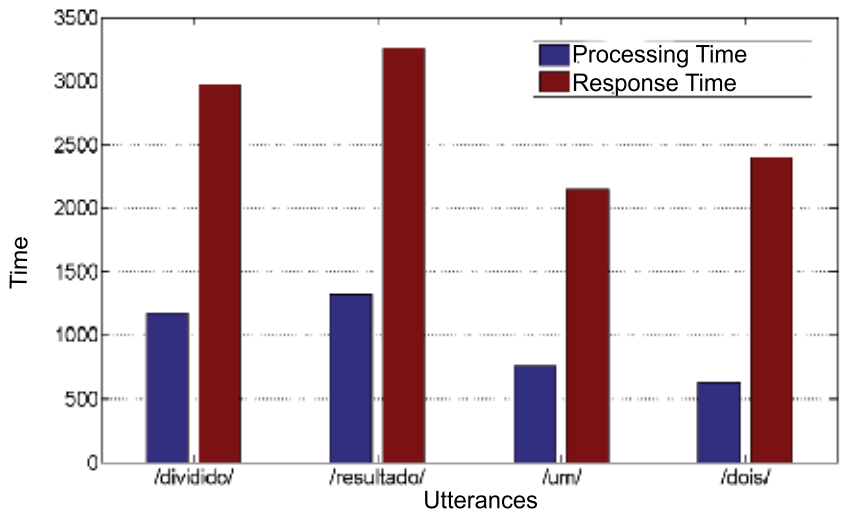


Fig. 4. Processing and response times (in milliseconds) for four words: */dividido/* (divided by), */resultado/* (result), */um/* (one) and */dois/* (two).

acquisition of the speech data is made under conditions which are hard to control in real-world scenarios, for example, the distance of the user from the microphone, noise level and the low-quality of the microphone device.

5. Conclusions

In this paper we analyzed the feasibility of embedding neural network based speech recognition systems into smartphones. We were motivated by the interest in developing a voice-driven software calculator for the Brazilian Portuguese language. For this purpose we first described a number of software techniques to speedup SOM computations, then described the TS-SOM network. The next step was to compare the offline classification performances of the SOM and the TS-SOM networks with those provided by classifiers commonly used for speech recognition purposes, such as the MLP, the *K*-means and the DTW algorithms. From these experiments we were able to select those architectures that can be embedded into a Nokia N95 smartphone for online recognition of words. The choice was based on the tradeoff between high recognition rates and a low processing times. Based on these restrictions, the selected models were the SOM:PDS and TS-SOM networks. From these two classifiers, only the TS-SOM classifier can be trained and tested online, i.e. inside the smartphone itself.

Currently we are optimizing the Java code used to implement the TS-SOM and the SOM:PDS in the embedded application in order to further diminish their processing times. We are also evaluating other speech parameterization techniques, such as wavelets, in order to increase the recognition rates of those two classifiers.

6. References

Alhonen, J., Cao, Y., Ding, G., Liu, Y., Olsen, J., Wang, X. & Yang, X. (2007). Mandarin short message dictation on symbian series 60 mobile phones, *Proceedings of the 4th international conference on mobile technology, applications, and systems (Mobility '07)*, pp. 431–438.

Amerijckx, C., Verleysen, M., Thissen, P. & Legat, J.-D. (1998). Image compression by self-organized Kohonen map, *IEEE Transactions on Neural Networks* 9(3): 503–507.

Barreto, G. A. & Araújo, A. F. R. (2004). Identification and control of dynamical systems using the self-organizing map, *IEEE Transactions on Neural Networks* 15(5): 1244–1259.

Barreto, G. A., Araújo, A. F. R. & Ritter, H. J. (2003). Self-organizing feature maps for modeling and control of robotic manipulators, *Journal of Inteligent and Robotic Systems* 36(4): 407–450.

Bei, C.-D. & Gray, R. (1985). An improvement of the minimum distortion encoding algorithm for vector quantization, *IEEE Transactions on Communications* 33(10): 1132–1133.

Cho, J., Principe, J., Erdogmus, D. & Motter, M. (2006). Modeling and inverse controller design for an unmanned aerial vehicle based on the self-organizing map, *IEEE Transactions on Neural Networks* 17(2): 445–460.

Deller, J. R., Hansen, J. H. L. & Proakis, J. G. (2000). *Discrete-Time Processing of Speech Signals*, John Wiley & Sons.

Frota, R. A., Barreto, G. A. & Mota, J. C. M. (2007). Anomaly detection in mobile communication networks using the self-organizing map, *Journal of Intelligent and Fuzzy Systems* 18(5): 493–500.

Gas, B., Chetouani, M., Zarader, J. & Feiz, F. (2005). The predictive self-organizing map :

application to speech features extraction, *in* M. Cottrell (ed.), *Proceedings of the 5th Workshop on Self-Organizing Maps (WSOM'2005)*, pp. 497–504.

Guillén, A., Herrera, L., Rubio, G., Pomares, H., Lendasse, A. & Rojas, I. (2010). New method for instance or prototype selection using mutual information in time series prediction, *Neurocomputing* 73(10-12): 2030–2038.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps, *Biological Cybernetics* 43: 59–69.

Kohonen, T. (1988). The 'neural' phonetic typewriter, *Computer* 21(3): 11–22.

Kohonen, T. K. (1997). *Self-Organizing Maps*, 2nd extended edn, Springer-Verlag, Berlin, Heidelberg.

Kohonen, T. K. (2001). *Self-Organizing Maps*, 3rd edn, Springer-Verlag, Berlin, Heidelberg.

Koikkalainen, P. & Oja, E. (1990). Self-organizing hierarchical feature maps, *International Joint Conference on Neural Networks (IJCNN'90)*, pp. 279–284 vol.2.

Lendasse, A., Lee, J., Wertz, V. & Verleysen, M. (2002). Forecasting electricity consumption using nonlinear projection and self-organizing maps, *Neurocomputing* 48: 299–311.

Martin, C., Diaz, N. N., Ontrup, J. & Nattkemper, T. W. (2008). Hyperbolic SOM-based clustering of DNA fragment features for taxonomic visualization and classification, *Bioinformatics* 24(14): 1568–1574.

Niskanen, M., Kauppinen, H. & Silvan, O. (2002). Real-time aspects of SOM-based visual surface inspection, *Proceedings SPIE Machine Vision Applications in Industrial Inspection*.

Oh, K.-S. & Jung, K. (2004). Gpu implementation of neural networks, *Pattern Recognition* 37: 1311–1311.

Olsen, J., Cao, Y., Ding, G. & Yang, X. (2008). A decoder for large vocabulary continuous short message dictation on embedded devices, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*, pp. 4337–4340.

Rabiner, L. & Juang, B.-H. (1993). *Fundamentals of speech recognition*, Prentice-Hall International.

Rabiner, L. R. & Schafer, R. W. (1978). *Digital Processing of Speech Signals*, Prenctice-Hall, New Jersey.

Sagheer, A., Tsuruta, N., Maeda, S., Taniguchi, R.-I. & Arita, D. (2006). Fast competition approach using self-organizing map for lip-reading applications, *International Joint Conference on Neural Networks (IJCNN'06)*, pp. 3775–3782.

Sakoe, H. & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition, *IEEE Transactions on Acoustics, Speech and Signal Processing* 26(1): 43–49.

Scanzio, S., Cumani, S., Gemello, R., Mana, F. & Laface, P. (2010). Parallel implementation of artificial neural network training for speech recognition, *Pattern Recognition Letters* 31: 1302–1309.

Somervuo, P. (2000). Competing hidden markov models on the self-organizing map, *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00)*, Vol. 3, pp. 169–174.

van Hulle, M. (2010). Self-organizing maps, *in* G. Rozenberg, T. Baeck & J. Kok (eds), *Handbook of Natural Computing: Theory, Experiments, and Applications*, Springer-Verlag, pp. 1–45.

Varga, I. & Kiss, I. (2008). Speech recognition in mobile phones, *in* Z.-H. Tan & B. Lindberg (eds), *Automatic Speech Recognition on Mobile Devices and over Communication Networks*, Springer, pp. 301–325.

Wang, J.-F., Wang, J.-C., Mo, M.-H., Tu, C.-I. & Lin, S.-C. (2008). The design of a speech

interactivity embedded module and its applications for mobile consumer devices, *IEEE Transactions on Consumer Electronics* 54(2): 870–875.

Xiao, Y., Leung, C. S., Ho, T.-Y. & Lam, P.-M. (2010). A GPU implementation for LBG and SOM training, *Neural Computing and Applications* . DOI: 10.1007/s00521-010-0403-7.

Yin, H. (2008). *Computational Intelligence: A Compendium*, Vol. 115 of *Studies in Computational Intelligence*, Springer-Verlag, chapter The Self-Organizing Maps: Background, Theories, Extensions and Applications, pp. 715–762.

7

Combining SOMs and Ontologies for Effective Web Site Mining

Dimitris Petrilis and Constantin Halatsis
National and Kapodistrian University of Athens
Greece

1. Introduction

The Internet since the late 90s, when it became mainstream, has dramatically changed the way people work, communicate, get educated, socialize and stay informed about current affairs. According to WorldWideWebSize.com (WorldWideWebSize.com, 2010) the Indexed Web contains at least 14.56 billion pages as of August 2010. The estimated minimal size of indexed World Wide Web is based on the estimations of the numbers of pages indexed by Google, Bing, Yahoo Search and Ask. In addition according to Royal Pingdom (Royal Pingdom, 2009) on December 2009 there were 234 million web sites and on September 2009 1.73 billion Internet users.

In the past few years we have also witnessed a new explosion in the usage of the World Wide Web (WWW) with what is commonly referred to as Web 2.0. The term Web 2.0 refers to the set of web sites whose contents are modified by visitor contributions and not only by the webmaster. This includes social networks such as Facebook (Facebook), LinkedIn (LinkedIn), Twitter (Twitter), MySpace (MySpace) as well as blogs and web sites where visitors can share pictures, such as Picasa (Picasa), or video files, such as YouTube (YouTube). The WWW has become an extremely interactive form of communication. The following statistics from Royal Pingdom (Royal Pingdom, 2009) illustrate the rapid penetration of Social Media Networks to everyday life:

- 126 million – the number of blogs on the Internet (as tracked by BlogPulse)
- 27.3 million – the number of tweets on Twitter per day (November, 2009)
- 350 million – people on Facebook
- 50% - percentage of Facebook users that log in every day
- 500,000 – the number of Facebook applications

Bearing in mind these huge numbers it is easy to image the massive amount of information available on the World Wide Web.

On the other hand significant advances in networking technology (such as very fast Internet connections) and search engines have created an impatient Internet culture. Internet users expect to be able to find the information they seek within seconds. Accessing the Internet is no longer an extracurricular activity that people perform at their spare time at home or a business tool used only by large corporations. It has become a common part of our everyday life. Many people have always an Internet connection available to them in their office and home equipment as well as through mobile devices.

Internet visitors are expecting to find information quickly and easily. They can be very harsh in the sense that they will not give a web site a second chance if they cannot find something interesting within the first few seconds of browsing. At the same time web sites are packed with information and hence presenting to every visitor the right information has become very complex. This has created two main challenges when maintaining a web site:

- Attracting visitors, i.e. getting people to visit the web site.
- Keeping visitors on the web site long enough so that the objective of the site can be achieved, e.g. if we are talking about an Internet store to make a sale.

This chapter deals with the second challenge, how to help web site visitors find information quickly and effectively by using clustering techniques. There is a plethora of methods for clustering web pages. These tools fall under a wider category of data mining called Web mining. According to Cooley (Cooley et al., 1997) Web mining is the application of data mining techniques to the World Wide Web. Their limitation is that they typically deal either with the content or the context of the web site. Cooley (Cooley et al., 1997) recognises that the term web mining is used in two different ways:

- Web content mining – information discovery from sources across the World Wide Web.
- Web usage mining – mining for user browsing and access patterns. In this paper we also refer to web usage mining as context mining.

The content of a web site can be analysed by examining the underlying source code of its web pages. This includes the text, images, sounds and videos that are included in the source code. In other words the content of a web site consists of whatever is presented to the visitor. In the scope of this chapter we examine the text that is presented to the visitor and not the multimedia content. Content mining techniques can be utilised in order to propose to the visitors of a web site similar web page(s) to the one that they are currently accessing. Metrics such as the most frequently occurring words can be used to determine the content of the web site (Petrilis & Halatsis, 2008). In this chapter we introduce an ontology-based approach for determining the content of the web site. However, it must be noted that the focus of this chapter is on the usage of SOMs and not on the usage of ontologies. Additional research is required for establishing the additional value of using ontologies for the purpose of context mining.

The page currently being viewed may be a good indicator of what the visitor is looking for, however it ignores the navigation patterns of previous visitors. The aim of context mining techniques is to identify hidden relationships between web pages by analysing the sequence of past visits. It is based on the assumption that pages that were viewed in some sequence by a past visitor are somehow related. Typically context miming is applied on the access-logs of web sites. The web server that is hosting a web site typically records important information about each visitor access. This information is stored in files called access logs. The most common data that can be found in access-logs is the following:

- the IP address of the visitor
- the time and date of access
- the time zone of the visitor in relation to the time zone of the web server hosting the web page
- the size of the web page
- the location (URL) of the web page that the visitor attempted to access
- an indication on whether the attempt to access the web page was successful
- the protocol and access method used

- the referrer (i.e. the web page that referred the visitor to the current page) and
- the cookie identifier

Clustering algorithms can be used to identify web pages that visitors typically visit on the same session (a series of web page accesses by the same visitor). The output of the clustering algorithms can be used to dynamically propose pages to current visitors of the web site.
The problem with most web mining clustering techniques is that they focus on either content, such as WEBSOM (Lagus et al, 2004), or context mining (Merelo et al, 2004). This way important data regarding the web site is ignored during processing. The combination of both content and context mining using SOMs can yield better results (Petrilis & Halatsis, 2008). However, when this analysis takes place in two discreet steps then it becomes difficult to interpret the results and to combine them so that effective recommendations can be made. In this chapter we are going to demonstrate how we can achieve better results by producing a single SOM that is the result of both content and context mining into a single step. In addition we are going to examine how the usage of ontologies can improve the results further.
To illustrate our approach and findings we have used the web pages and access-logs of the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens.

2. Kohonen's self-organising maps

It is not in the scope of this chapter to provide a detailed definition of Kohonen's Self-Organising maps since it is assumed that the reader already has some knowledge regarding this unsupervised neural network technique. According to Kohonen (Kohonen, 2001), the SOM in its basic form produces a similarity graph of input data. It converts the nonlinear statistical relationships among high-dimensional data into simple geometric relationships of their image points on a low-dimensional display, usually a regular two-dimensional grid of nodes. As the SOM thereby compresses information while preserving the most important topological and/or metric relationships of the primary data elements on the display, it may also be thought to produce some kind of abstractions. There are many variations of SOMs (Kohonen, 2001) and in the context of this research we are using the basic form that was proposed by Kohonen.
There is a plethora of different software packages that implement different variations of the SOM. In order to perform our research we use SOM_PAK (SOM_PAK and LVQ_PAK). This package includes command-line programs for training and labelling SOMs, and several tools for visualizing it: sammon, for performing a Sammon (Sammon, 1969) projection of data, and umat, for applying the cluster discovery UMatrix (Ultsch, 1993) algorithm. SOM_PAK was developed by Kohonen's research team.

3. Web Mining

The term Web Mining is often subject to confusion as it has been traditionally used to refer to two different areas of data mining:

- Web Usage Mining - the extraction of information by analysing the behaviour of past web site visitors
- Web Content Mining – the extraction of information from the content of the web pages that constitute a web site.

3.1 Web usage mining

Web usage mining, also known as Web Log Mining, refers to the extraction of information from the raw data that is stored in text files located on the web server(s) hosting the web pages of a web site. These files are called access-logs. Typically each entry in the access log is one line in the text file and it represents an attempt to access a file of the web site. Examples of such files include: static html pages, dynamically generated pages, images, videos and sounds amongst others. A typical access log entry can be seen below:

134.150.123.52 - - [19/Aug/2010:15:09:30 +0200] "GET /~petrilis/index.html HTTP/1.0" 200 4518 "http://www2.di.uoa.gr/gr/" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)" 62.74.9.240.20893111230291463

The data of this example is explained in the table that follows:

Data Item	Description
134.150.123.52	The IP address of the computer that accessed the page
-	The identification code (in this case none)
-	The user authentication code (in this case none)
[19/Aug/2010:15:09:30 +0200]	The date, time and time zone (in this case 2 hrs ahead of the timezone of the web server hosting the web site) of the access
"GET /~petrilis/index.html HTTP/1.0"	The request type (GET), the web page accessed and the protocol version
200	The server response code (in this case the page was accessed correctly)
4518	The number of bytes transferred
"http://www2.di.uoa.gr/gr/"	The referrer page
"Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)"	The user agent information, i.e. browser information
62.74.9.240.20893111230291463	Cookie string

Table 1. Data contained in an access-log

There is a large number of software solutions that can perform analysis of the access-logs. Most of these perform simple statistical analysis and provide information, such as the most commonly accessed page, the time of the day that the site has more access, etc. For example WebLog Expert (WebLog Expert) provides the following analysis:

- General statistics
- Activity statistics: daily, by hours of the day, by days of the week and by months
- Access statistics: statistics for pages, files, images, directories, queries, entry pages, exit pages, paths through the site, file types and virtual domains
- Information about visitors: hosts, top-level domains, countries, states, cities, organizations, authenticated users

- Referrers: referring sites, URLs, search engines (including information about search phrases and keywords)
- Browsers, operating systems and spiders statistics
- Information about errors: error types, detailed 404 error information
- Tracked files statistics (activity and referrers)
- Support for custom reports

Such information can provide some valuable information but it does not provide true insight on the navigational patterns of the visitors. Using clustering algorithms more in depth analysis can be performed and we can deduce more valuable information. For example we can identify clusters of visitors with similar access patterns. We can subsequently use this information to dynamically identify the most suitable cluster for a visitor based on the first few clicks and recommend to that visitor pages that other visitors from the same cluster also accessed in the past. There are different methods for performing such clustering ranging from simple statistical algorithms, such as the k-means, to neural network techniques, such as the SOM.

3.2 Web content mining

Web content mining is the application of data mining techniques to the content of web pages. It often viewed as a subset of text mining, however this is not completely accurate as web pages often contain multimedia files that also contribute to its content. A simple example of this is YouTube (YouTube) that mainly consists of video files. This is exactly the most important complexity of web content mining, determining the source of the content. The source code of the web pages, stripped of any tags, such as HTML tags, can be used as input (Petrilis & Halatsis, 2008). However, it is easy to see the limitation of such an approach bearing in mind that as we mentioned other types of files are also embedded in web pages. In addition quite often pages are dynamically generated and therefore we do not know their content in advance. Another additional constraint is the sheer volume of data that is often contained within web pages. In this chapter we attempt to address this issue by proposing an ontology based approach for determining the content of the web pages and for creating suitable input for SOM processing. It is not in the scope of this chapter to elaborate on ontology based techniques and this will be the subject of subsequent research by the authors. However, Paragraph 4 provides further details on our approach.

There are different methods that can be used for web content mining. Simple statistical analysis can provide some level of information such as the most popular words in each page or the most frequent words in the set of all the pages comprising the web site. However, this information is of limited use and does not unveil hidden relationships between web pages. Clustering algorithms can be used to unveil more complex relationships among the web pages by identifying clusters of web pages with similar content. This analysis can be used to dynamically propose web pages to visitors. WEBSOM (Lagus et al., 2004) utilises the SOM algorithm to generate a map that displays to the visitor pages of similar content with the page that is currently being viewed. The recommended pages are topographically placed in the map. The closer a recommended page is to the current location of the visitor within the map, the more relevant the recommendation is. A sample of output of WEBSOM can be seen in Figure 1.

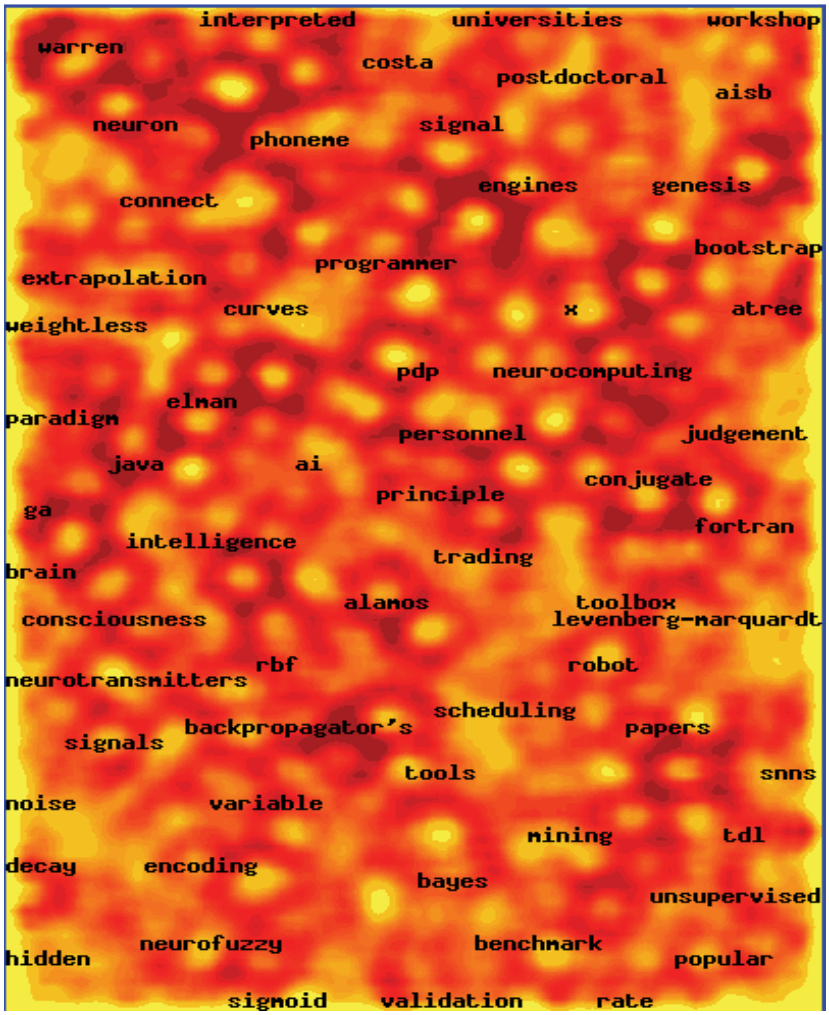


Fig. 1. Example output of WEBSOM

4. Ontology

It is not in the scope of this chapter to provide an in-depth analysis of ontologies and their usage on web mining. However, since a simple ontology has been used to achieve better results in our processing it is useful to provide an overview of ontologies.

Ontology as a term was originally used in philosophy to study the conceptions of reality and the nature of being. Looking at the etymology of the word "ontology", it originates from the Greek word "On" which means "Being". Hence, ontology is the study of "being". Ontology as an explicit discipline was created by the great ancient philosopher Aristotle. According to Gruber (Gennari, 2003) an ontology is an explicit specification of a conceptualization. A "conceptualization" is an abstract, simplified view of the world that we wish to represent for some purpose. According to Katifori (Katifori et al., 2007) it contains the objects, concepts and other entities that are presumed to exist in some area of interest

and the relations that hold them. An ontology is a formal explicit description of concepts in a logical discourse. In ontology concepts are known as classes, the properties of each concept describing various features and attributes of the classes are referred to as slots or properties and the restrictions on the slots as facets. A specific ontology with a set of class instances constitutes a knowledge base.

Ontologies are a very popular tool for adding semantics to web pages in order to facilitate better searching. Luke (Luke et al., 1996) proposes an ontology extension to HTML for exactly that purpose. Berners-Lee (Berners-Lee et al., 2001) suggests the usage of ontologies for enhancing the functioning of the Web with the creation of the Semantic Web of tomorrow. The WWW Consortium (W3C) has created the Resource Description Framework, RDF, a language for encoding knowledge on web pages to make it understandable to electronic agents searching for information. Ontologies are not only used for research purposes but also have many commercial applications. As an example many key players in the WWW, such as Yahoo and Amazon, use ontologies as a means of categorising their web pages.

In the context of the WWW typically the primary use of ontologies is not the description of the domain. It is the definition of the data and its inherent structure so that it can be used more effectively for further processing and analysis. A typical example is the Semantic Web. The goal of the Semantic Web is to make it possible for human beings and software agents to find suitable web content quickly and effectively. The definition of the underlying data itself is not the primary objective.

The focus of our research in the chapter is to achieve better results in clustering web pages by producing a single SOM that is the result of both content and context mining. By introducing the use of a very simple ontology in the content mining part we demonstrate improved results. The tool that was used for creating this simple ontology is Protégé. Protégé is an environment for knowledge-based systems that has been evolving for over a decade (Gruber, 1993). It implements a rich set of knowledge-modelling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats. Protégé has been selected because it is one of the most complete packages for the creation of ontologies and at the same time it is very simple to use. In addition a large number of extensions are available (Gruber, 1993). A comprehensive comparison of ontology development environments has been performed by Duineveld (Duineveld et al., 2000).

It is well known and documented that web mining as any other data mining technique can only produce useful results if a suitable data set is used. Hence, it is important to examine the data preparation steps in more detail.

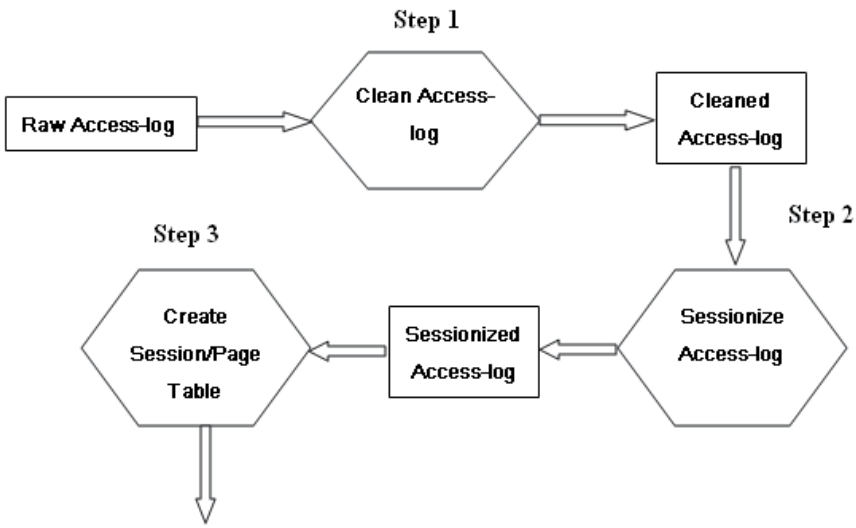
5. Data preparation

As it was previously mentioned the results of any data mining analysis can only be as good as the underlying data. Hence it is important to present the pre-processing steps that are required prior to applying the SOM.

5.1 Data preparation for context mining

As it was previously mentioned web site context mining deals with the analysis of the access-logs that are stored in web servers. Typically the access-logs contain a large amount

of noise. This is data that not only does not add any value to processing but on the contrary skews the results. Each time a visitor accesses a web page, a number of files are being accessed. These may include the main web page (typically HTML), images, videos and audio files. Some of these files, for example a logo that may be present in every web page of the site, generate noise to the access logs. In addition search engines use software agents called web robots that automatically traverse the hyperlink structure of the World Wide Web in an effort to index web pages (Noy & McGuniness, 2001). These software agents perform random accesses to web pages and hence generate access logs entries of no value. Identifying these robot accesses is a difficult task. Another important consideration when processing access-logs is that quite often an IP address does not uniquely identify a visitor. Therefore, we need to introduce the concept of a visitor session. A visitor session for the purposes of our research is a visitor access from a specific IP address within a specific time frame.



	page1	page2	.....	pagen
Session1	1	1	.....	0
Session2	0	0	.....	1
.....	.....	.....	.....	.....
Sessionm	4	0	.....	2

Fig. 2. Data preparation steps for context mining

In order to prepare context related data for input to the SOM the following pre-processing steps were followed that are also depicted in Figure 2:

- Noise Removal - removal of image, video, audio and web robot accesses from the access-logs. It must be noted that in order to simplify the processing all image, video and audio accesses were removed regardless of their content. WumPrep (WumPrep) is used for this purpose. WumPrep is a collection of Perl scripts designed for removing noise from access-logs and preparing them for subsequent processing.

- Session Identification –WumPrep was used to identify visitor sessions and assign to each access a suitable session identifier. Access-log entries with the same session identifier are part of the same session. It must be noted that WumPrep offers the option of inserting dummy entries at the beginning of each session for the referring site, if this is available. We have selected this option as we believe the origin of the access is valuable data.
- Session Aggregation – aggregation of sessions and creation of a session/page matrix that identifies how many times each session visited each of the web pages of the web site.

As a result of the data preparation for content mining we produce a matrix with the rows representing individual sessions and the columns the available web pages. Each row presents which pages and how many times each session visited. A value of zero denotes that the page was not visited by that session; a non-zero value of x indicates that the web page was visited x times during that session.

5.2 Data preparation content mining

In order to depict the contents of the web pages more accurately an ontology of the web site is created. The ontology, despite the fact that it is very simple, provides better results than other techniques such as counting the number of occurrences of words within the web pages (Petrilis & Halatsis, 2008). In the future the authors plan to use a more comprehensive ontology in order to further improve the results.

The ontology describes the set of the web pages that constitute the web site. The main classes, slots and role descriptions are identified. Protégé is used as the visualization tool for the ontology (Protégé). The classes and the value slots have been used to determine the content of each of the web pages. There are six main classes in the ontology that has been created:

- Person –the type of author of the web page
- Web Page – indicates whether it is an internal or an external page
- File – information about the web page file (e.g. name, type, etc)
- Company –company name and type that is associated to the specific web page
- Structure –the place of the web page in the structure of the web site
- URL – information about the URL (static or dynamic and the actual address)

The ontology that was created for the purposes of our processing is depicted in Figure 3. These classes have subclasses, which in turn may have subclasses of their own. In addition classes have slots. As an example the class “URL” has two slots “Static or Dynamic” and “URL”. The first denotes whether the specific web page is statically or dynamically generated and the latter the actual URL of the web page. We have placed great emphasis in encapsulating the structure of the web site. The reason is that in order to get a better understanding of the contents of a web page we need to understand how it relates to other pages within the site.

Using the ontology as a basis we create a matrix with the rows representing individual web pages and the columns the available classes and possible slot values. Each row presents what classes and slot values are relevant to the specific web page. A value of zero denotes that the specific class or slot value is not relevant; a non-zero value indicates that the specific class or slot value is of relevance to the specific web page. The values have been weighted in order to depict the significance of the specific class or slot value to the web page. We apply

greater weights to the classes and slot values that relate to the structure of the web site, since they provide very important information regarding the contents of the web page.

Fig. 3. The Department of Informatics and Telecommunications Ontology

Data Item	Ont. class/slot 1	Ont class/slot 2	…….	Ont class/slot n
Page 1	75	100		0
Page 2	0	100		0
……..	…	…	...	…
Page m	100	75		100

Table 2. Output of the data preparation for content mining

5.3 Combining content and context data

The input data for SOM processing is a combination of the output of the pre-processing steps described in paragraphs 5.1 and 5.2. A matrix is created with the rows representing individual sessions and the columns the available classes and possible slot values of the ontology of the web site. Table 3 shows a sample of the final output of the pre-processing.

Data Item	Ont. class/slot 1	Ont class/slot 2	…….	Ont class/slot n
Session 1	0	100		75
Session 2	75	0		0
……..	…	…	...	…
Session m	100	0		100

Table 3. Final output of the pre-processing

A value of zero indicates that the specific class or slot value is not relevant for the session, whereas a non-zero value denotes that the specific class or slot value is of relevance for the specific session, i.e. to the web pages this session accessed. Additionally a weight is applied to the non-zero values that signifies the relevant of the specific class or slot value to the session. A greater weight is applied to classes or slot values that relate to the structure of the web site, since this is more important in determining the content of the web page.

6. Clustering the data using the SOM

The output that is produced as part of the pre-processing steps described in Paragraph 5 is used as the basis for input to the SOM. The SOM_PAK application has specific formatting requirements and hence the matrix that can be seen in Table 3 is converted to the following format that can be seen in Table 4.

```
<dimensionality>
<class/slot value1 >  <class/slot value2> … < class/slot valuen> <session
id>
```

Table 4. Format of input file to the SOM

A sample of the input file can be seen in Table 5 below:

```
60
100 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.121.0.0
0 75 100 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.122.0.0
0 0 100 0 75 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.311.0.0
```

Table 5. Sample of the input file to the SOM

The dimensionality indicates the number of columns in the input file and it is a prerequisite for SOM_PAK. Each additional value in the input file denotes the relevance of each ontology class and slot value to the specific session. The session id is used as a label that appears in map produced by SOM and helps us identify individual sessions in the map.

SOM_PAK requires from the user to assign values to some parameters before initiating the processing. These parameter values were selected after evaluating the results with different combination of parameters. To assist with the evaluation of suitable parameters for SOM processing the Sammon projection (Sammon, 1969) of the data is used. The Sammon program of SOM_PAK is used for this purpose. This program provides a quick and easy visual way to evaluate the quality of produced maps for specific parameter values. The selected parameter values can be seen in the table that follows:

Data Item	Value
Neighborhood Type	Hexa
Neighborhood Function	Bubble
Map x size	20
Map y size	8
First Training Period Length	2000
First Training Neighborhood Radius	20
First Training Constant	0.5
Second Training Period Length	8000
Second Training Neighborhood Radius	5
Second Training Constant	0.05

Table 6. SOM Parameter Values

The output of processing is a map in the form of a text file with coordinates. This map is difficult to read and interpret and hence a means of visualising it is required. To visualise the results an UMatrix analysis is applied to the output of the SOM processing. UMatrix analysis provides a visual representation of the map making it easy to identify clusters of sessions. The UMatrix representation uses grey scale values to indicate the distance between nodes. The lighter the colour the closer two nodes are. The darker the colour the greater the distance. Clusters can be easily identified as areas of light hexagons separated by dark hexagons. The output of the UMatrix analysis can be seen in Figure 4.

In the map of Fig. 4 hexagons with a black dot in the centre represent nodes. The labels, such as 1.368.0.0, are session identifiers. The labels have been added to the node that best describes the specific input vector. Some of the clusters of this figure have been highlighted with colours to make it easier to demonstrate the results. Looking at the map of Fig. 4 and examining the underlying data we can quickly draw some conclusions:

- Red oval sessions have all accessed pages written by University staff that are relevant to research on algorithms.
- Sessions in the yellow cluster have accessed pages where the author is a member of the University Staff and/or University Teaching Staff and relate to research and research projects.
- The sessions in the blue cluster have only accessed the University's home page.
- Sessions in the violet cluster have accessed pages that relate to Research, Research Projects or Internet Applications. The authors of these pages are members of the University staff, University teaching staff or University students.
- Sessions in the grey cluster have accessed pages relating to Logic Algorithms and Computation or Internet Applications and were written by University Staff or University teaching staff.
- Sessions in the green cluster have accessed pages relevant to research, research areas, research projects and/or algorithms. These pages were written by University staff, University teaching staff or University students.

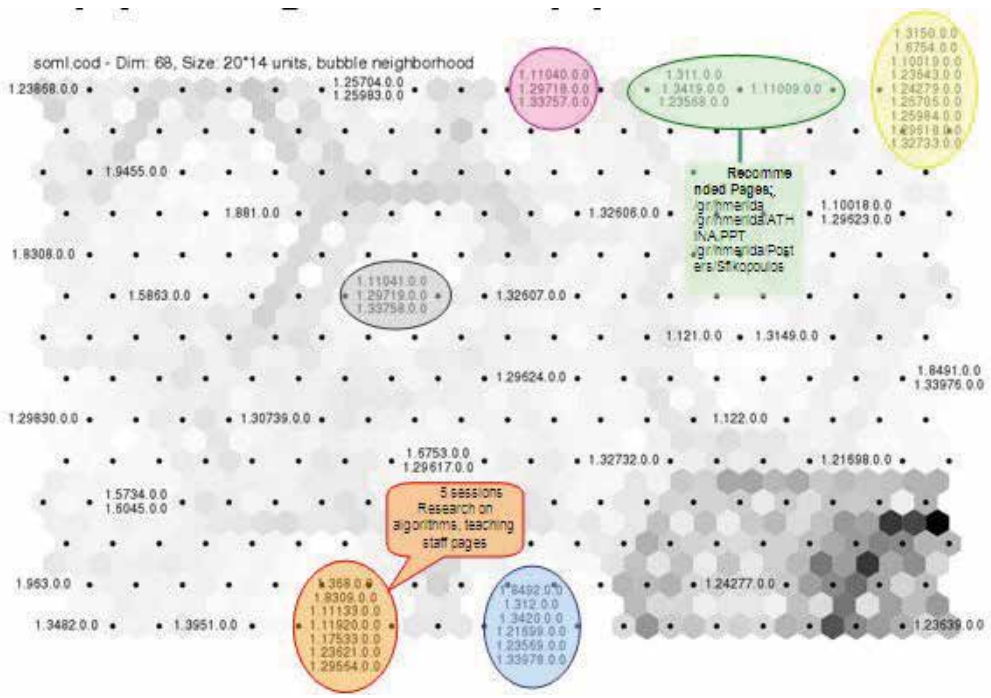


Fig. 4. UMatrix representation of the SOM output

As we have demonstrated by observing the map produced by SOM processing and by examining the underlying data we can quickly and easily extract useful information

regarding the web site. This output could be used to dynamically recommend pages to the visitors of the web site based on the contents of the page that they are currently viewing and on the behaviour of past visitors that have accessed the same or similar pages. As an example if a session accesses page /gr/hmerida then the rest of the pages relevant to the green cluster can be proposed (/gr/hmerida/ATHINA.PPT and /gr/hmerida/Posters/Sfikopoulos)

7. Conclusions

In recent years we have witnessed the rapid expansion of the Internet. There are billions of web pages that are registered by search engines. Web sites tend to increase in size accumulating an ever increasing amount of information. This is especially true for web sites that have been around for a number of years or are updated very often. Web 2.0 and the ever increasing popularity of Social Media Networks have created an Internet culture where visitors are no longer passive but they contribute to the contents of their favourite web sites on a regular basis. This has resulted in web sites that are very complex in their structure. In addition a large number of Internet users have always Internet access available to them through mobile devices. The demand to be able to find information quickly and easily is therefore apparent. Despite the continuous effort to improve the search engines, it is still often a challenge for web site visitors to achieve this.

There is a plethora of commercial applications as well as academic research on predicting web pages that will be useful to a visitor with the final goal of making recommendations to web site visitors. Clustering techniques have demonstrated a relatively good level of success compared to other methods, such as simple statistical applications. However, the current clustering techniques are typically incomplete in the sense they that focus either on the content or the context of the web site. This way important information is ignored when making recommendations because identifying the best web page to recommend depends on both the content of the pages that have been viewed already by the visitor but also on the behaviour of past visitors with similar interests.

In this chapter we present a method that combines both content and context mining. We demonstrate how we can achieve better results by producing a single Self Organising Map that combines data for both the content and context of a web site. Furthermore we demonstrate how a simplistic ontology of the web site can help in determining the content of the web pages. Our approach improves the results of previous research (Petrilis & Halatsis, 2008) and it correctly identifies hidden relationships within the data. In addition the results of the proposed method are easily visualized and interpreted and can be used to dynamically recommend web pages to web site visitors based on both the content of the page they currently viewing but also on the content of similar pages and on past visitor behaviour.

We intend to test our approach on a bigger and more complex web site. In addition it would be interesting to use a more diverse data set. The web sites and web pages of the Department of Informatics of University of Athens are limited in terms of the information they contain. Ideal candidates for our method would be the web site for an online store or an online newspaper and in general web sites with more diverse topics. In addition the ontology that was constructed to depict the contents of the web site pages is very simplistic. We strongly believe that a more comprehensive ontology will yield better results.

Furthermore in the future we plan to integrate the produced SOM with a recommender system that dynamically recommends pages to web site visitors.

8. References

Andrade MA.; Chacón P., Merelo-Guervós J. (1993). Evaluation of secondary structure of proteins from UV circular dichroism spectra, *Protein Eng,* Vol. 6, No. 4, pp. 383–390

Berners-Lee T.; Hendler J. and Lassila O. (2001). The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities, *Scientific American.com*

Chekuri C. et al. (1996). Web search using automatic classification, *Proceedings of Sixth World Wide Web conference*, San Jose, CA

Cooley R.; Mobasher B., Srivastava J. (1997). Web mining: Information and pattern discovery on the Worldwide Web, *International Conference on Tools with Artificial Intelligence,* pp. 558-567, Newport Beach, CA

Duineveld R. et al. (2000). Wondertools?: a comparative study of ontological engineering tools, *Int. J. Hum.-Comput. Stud.*, Vol. 52, No. 6. (June 2000), pp. 1111-1133

Facebook, http://www.facebook.com

Gruber T. (1993). A Translation Approach to Portable Ontology Specifications, *Knowledge Acquisition*, Vol. 5, No. 2, 199-220, California

Gennari J.H. et al. (2003). The evolution of Protégé: an environment for knowledge-based systems development, *International Journal of Human-Computer Studies*, Vol. 58. No. 1, pp. 89-123

Kaski S. (1997). Computationally efficient approximation of a probabilistic model for document representation in the websom full-text analysis method, *Neural Process Lett*, Vol. 5, No. 2, pp. 69–811

Katifori V. et al. (2007). Ontology Vizualization methods-a survey, *ACM Computer Surveys*, Vol. 39, No. 4

Kohonen T. (2001). *Self-organizing maps*, 3rd edn. Springer-Verlag, Berlin

Lagus K.; Kaski S., Kohonen T. (2004). Mining massive document collections by the WEBSOM method, *Information Sci*, Vol. 163, No. 1–3, pp. 135–156

LinkedIn, http://www.linkedin.com

Luke S.; Spector L. and Rager D. (1996). Ontology-Based Knowledge Discovery on the World-Wide Web, *Working Notes of the Workshop on Internet-Based Information Systems at the 13th National Conference on Artificial Intelligence*, pp. 96-102, USA

Merelo JJ. et al. (2004). Clustering web-based communities using self-organizing maps, In: *Proceedings of IADIS conference on web based communities,* Lisbon, Portugal

Mobasher B.; Cooley R., Srivastava J. (1999). Creating AdaptiveWeb Sites through Usage-based Clustering of URLs. *Proceedings of 1999 workshop on knowledge and data engineering exchange*, pp. 19-26, USA

MySpace, http://www.myspace.com

Netcraft (2008). *Weber Server Survey*, http://news.netcraft.com/archives/web_server_survey.html

Noy NF.; McGuniness D.L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology, Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880

Pang-Ning T.; Vipin K. (2002). Discovery of Web Robot Sessions Based on their Navigational Patterns, *Data Mining and Knowledge Discovery*, Vol. 6, No. 1, pp. 9-35

Petrilis D.; Halatsis C. (2008). Two-level clustering of web sites using self-organizing maps, *Neural Process Lett,* Vol. 27, No. 1, pp. 85-95

Picasa, http://picasa.google.com

Protégé, http://protege.stanford.edu/

Quesada J.; Merelo-Guervós J.J., Oliveras M.J. (2002). Application of artificial aging techniques to samples of rum and comparison with traditionally aged rums by analysis with artificial neural nets, *J Agric Food chem.,* Vol. 50, No. 6, pp. 1470–1477

Romero G. et al. (2003). Visualization of neural network evolution, *Lecture notes in computer science*, Nos. 2686–2687, pp. 534–541, LNCS , Springer-Verlag

Royal Pingdom. (2009). http://royal.pingdom.com/2010/01/22/internet-2009-in-numbers/, *Internet 2009 in numbers*

Sammon J.W. Jr. (1969). A nonlinear mapping for data structure analysis, *IEEE TransComput* Vol. 18, pp. 401–409

SOM_PAK and LVQ_PAK, http://www.cis.hut.fi/research/som-research/nnrc-programs.shtml

Twitter, http://www.twitter.com

Ultsch A. (1993). Self-organizing neural networks for visualization and classification , In: Opitz O, Lausen B, Klar R (eds) *Information and classification*, pp. 307–313, Springer, London, UK

Vesanto J. et al. (1999) Self-Organizing map in Matlab: the SOM Toolbox, *Proceedings of the Matlab DSP conference,* pp. 35–40, Espoo, Finland

WebLog Expert, http://www.weblogexpert.com/

WorldWideWebSize.com. (2010). http://www.worldwidewebsize.com/, *Daily Estimated Size of the World Wide Web*

WumPrep, http://www.hypknowsys.de

YouTube, http://www.youtube.com

Zhang J.; Caragea D., Honavar V. (2005). Learning ontology-aware classifiers, *Proceedings of the eight international conference on discovery science (DS 2005)*, Springer-Verlag, Berling

8

A Study on Facial Expression Recognition Model using an Adaptive Learning Capability

Masaki Ishii
Akita Prefectural University
Japan

1. Introduction

The study of facial expression recognition for the purpose of man-machine emotional communication is attracting attention lately (Akamatsu, 2002a; Akamatsu, 2002b; Akamatsu, 2002c; Akamatsu, 2003; Fasel & Luettin, 2003; Pantic & Rothkrantz, 2000; Tian et al., 2001). Most facial expression recognition models that have been proposed eventually create some classifier based on the expression images taken during a short period of time and using them as base data for learning (Pantic et al., 2005; Gross, 2005). However, because so many facial expression patterns exist that a human being cannot make representations of all of them, it is difficult to obtain and retain all available patterns and use them as learning data in a short time. The actual facial expressions that change from one time to another would show the other patterns at other times that are not contained in the learning data. For that reason, it is thought to be difficult to maintain and recognize those facial expressions just as they are without changing them continuously for a long time using the same classifier that was created at the initial stage.

For a facial expression recognition model to retain its high robustness along the time axis continuously for a long time, the classifier created at the initial stage should be evolved to be adaptive gradually over time. In other words, what is necessary for the model is that it retains existing knowledge (i.e. past facial patterns) and simultaneously learns to keep adding newly available knowledge (i.e. new facial patterns) as it becomes available.

As described in this chapter, we propose a method of creating a facial expression recognition model that can offer the adaptive learning capability described above. In addition, its degree of usefulness is described. We will show it from results of experiments made for evaluation of the incremental learning capability that the model has. Thereby, we will examine that point specifically.

2. Previous studies

Earlier reports (Ishii et al., 2008a; Ishii et al., 2008b) have presented a generation method of a subject-specific emotional feature space using the Self-Organizing Maps (SOM) (Kohonen, 1995) and the Counter Propagation Networks (CPN) (Nielsen, 1987). The feature space expresses the correspondence relationship between the change of facial expression pattern and the strength of emotion on the two-dimensional space centering on ″pleasantness″ and ″arousal″. Practically speaking, we created two kinds of feature space, Facial Expression

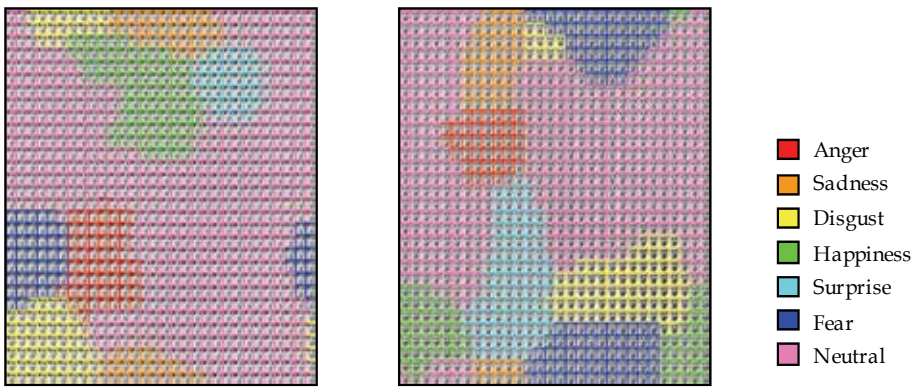
Map and Emotion Map, by learning the facial images using the CPN. The CPN is a supervised learning algorithm that combines the Grossberg learning rule with the SOM. With a facial image fed into the CPN after some learning process, the Facial Expression Map can determine the unique emotional category for the image that is fed in. Furthermore, the Emotion Map can quantize the level of the emotion of the image based on the level of facial pattern changes that occur.

Figures 1 and 2 respectively present the Facial Expression Map and Emotion Map generated using the proposed method. Figure 3 shows the recognition result for expression of "fear" and "surprise", which reveals pleasantness value and arousal value gradually change with the change of facial expression pattern. Moreover, the change of pleasantness value and arousal value is similar, although facial expression patterns of two subjects are different.

Figure 4 depicts the procedures of the previous method. The method consists of following three steps.

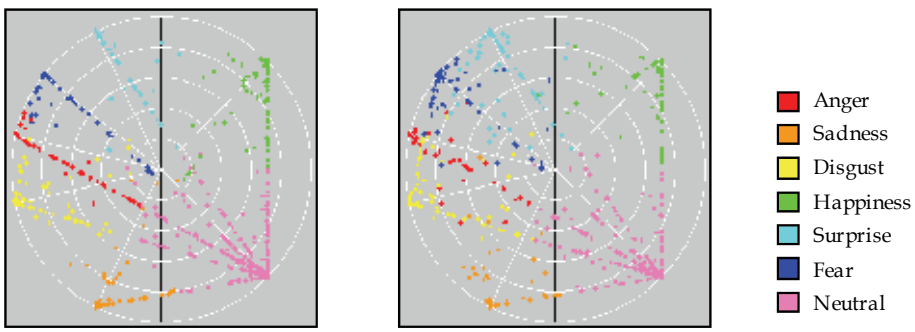
Step 1. Extraction of subject-specific facial expression categories using the SOM.
Step 2. Generation of Facial Expression Map using the CPN.
Step 3. Generation of Emotion Map using the CPN.

Details of target facial expression images and above three steps are explained below.



(a) Subject A. (b) Subject B.

Fig. 1. Generation results of Facial Expression Map



(a) Subject A. (b) Subject B.

Fig. 2. Generation results of Emotion Map

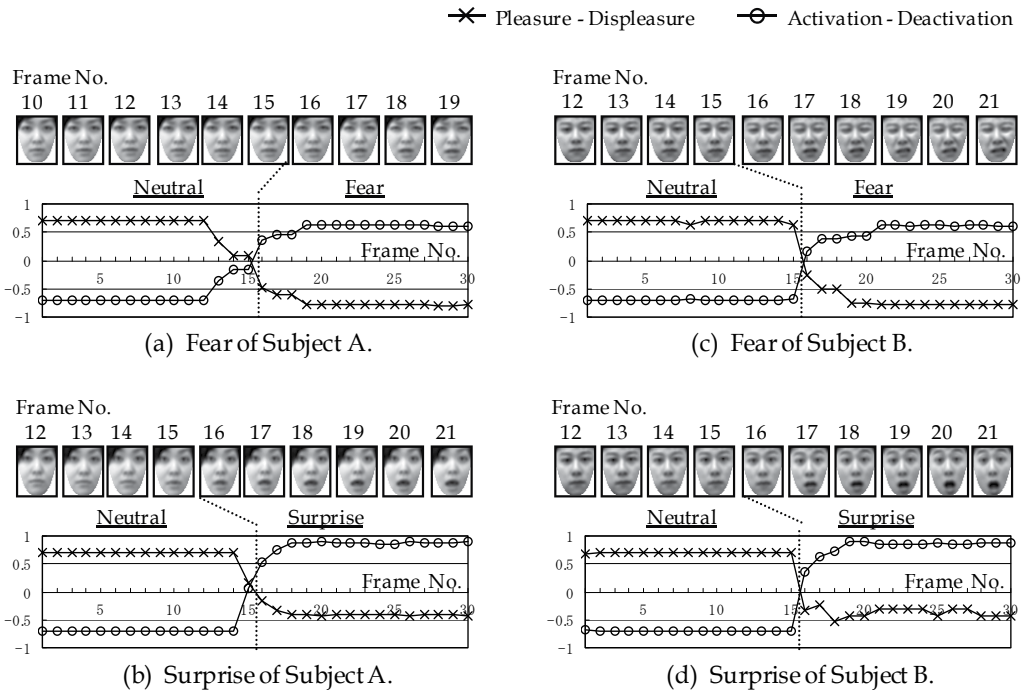


Fig. 3. Recognition result for "fear" and "surprise" of Subject A and B

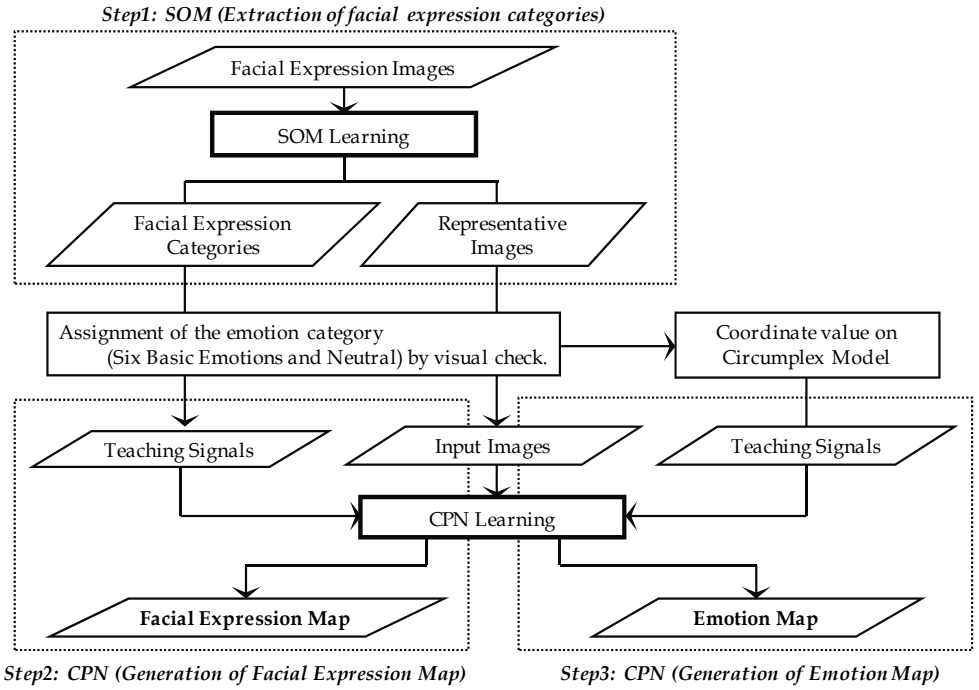


Fig. 4. Flow chart of proposal method in previous studies

2.1 Target facial expression images

Open facial expression databases are generally used in conventional studies (Pantic et al., 2005; Gross, 2005). These databases contain a few images per expression and subject. For this study, we obtained facial expression images of ourselves because the proposed method extracts subject-specific facial expression categories and the representative images of each category from large quantities of data.

This section presents a discussion of six basic facial expressions and a neutral facial expression that two subjects manifested intentionally. Basic facial expressions were obtained as motion videos including a process in which a neutral facial expression and facial expressions were manifested five times respectively by turns for each facial expression. Neutral facial expressions were obtained as a motion video for about 20 s. The motion videos were converted into static images (30 frame/s, 8 bit gray, 320 × 240 pixels) and used as training data. A region containing facial components was processed in this chapter; extraction and normalization of a face region image were performed according to the following procedures. Figure 5 shows an example of face region images after extraction and normalization.

1. A face was detected using Haar-like features (Lienhart & Maydt, 2002); a face region image normalized into a size of 80 × 96 pixels was extracted.
2. The image was processed using a median filter for noise removal. Then smoothing processing was performed after dimension reduction of the image using coarse grain processing (40 × 48 pixels).
3. A pseudo outline that is common to all the subjects was generated; the face region containing facial components was extracted.
4. Histogram linear transformation was performed for brightness value correction.

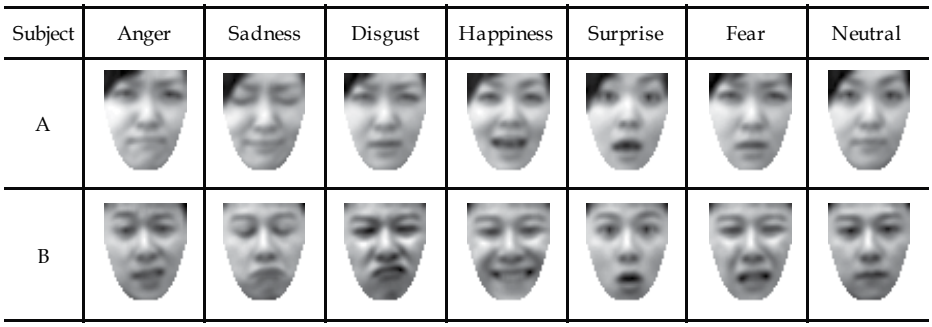
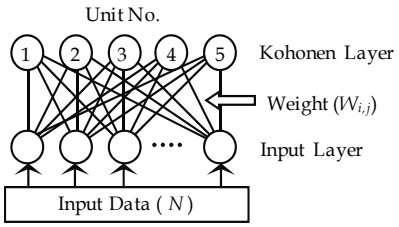


Fig. 5. Examples of facial expression images

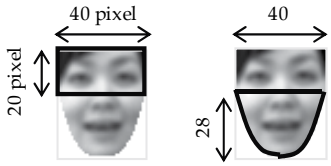
2.2 Extraction of facial expression category

The proposed method was used in an attempt to extract a subject-specific facial expression category hierarchically using a SOM with a narrow mapping space. A SOM is an unsupervised learning algorithm and classifies given facial expression images self-organizedly based on their topological characteristics. For that reason, it is suitable for a classification problem with an unknown number of categories. Moreover, a SOM compresses the topological information of facial expression images using a narrow mapping space and performs classification based on features that roughly divide the training data.

We speculate that repeating these hierarchically renders the classified amount of change of facial expression patterns comparable; thereby, a subject-specific facial expression category can be extracted. Figure 6 depicts the extraction procedure of a facial expression category. Details of the process are explained below.



(a) SOM architecture.



(c) Target region (Upper and Lower face).

Unit No.	1	2	3	4	5
Visualized Image ($W_{i,j}$)					
Classification Result	n_1	n_2	n_3	n_4	n_5
Correlation Coefficient		0.9946	**0.9749**	0.9865	0.9966
New Training Data	N_1		N_2		

* $N = n_1 + n_2 + n_3 + n_4 + n_5$ * $N_1 = n_1 + n_2$, $N_2 = n_3 + n_4 + n_5$

(b) Learning with SOM and setup of new training data.

Fig. 6. Extraction procedure of a facial expression category

1. Expression images described in Section 2.1 were used as training data. The following processing was performed for each facial expression. The number of training data is assumed as N frames.
2. Learning was conducted using a SOM with a Kohonen layer of five units and an input layer of 40 × 48 units (Fig. 6 (a)), where the number of learning sessions as set as 10,000 times.
3. The weight of the Kohonen layer $W_{i,j}$ ($0 \leq W_{i,j} \leq 1$) was converted to a value of 0 - 255 after the end of learning, and a visualized images were generated (Fig. 6 (b)), where n_1 - n_5 are the number of training data classified into each unit.
4. Five visualized images can be considered as representative vectors of the training data classified into each unit (n_1 - n_5). Therefore, whether a visualized image was suitable as a representative vector was judged using a threshold process. Specifically, for the upper and lower faces presented in Fig. 6 (c), a correlation coefficient between a visualized image and classified training data was determined for each unit. The standard deviation of those values was computed. When the standard deviation of both regions was 0.005 or less in all five units, the visualized image was considered to represent training data and the subsequent hierarchization processing was cancelled.
5. The correlation coefficient of weight $W_{i,j}$ between each adjacent unit in the Kohonen layer was computed. The Kohonen layer was divided into two bordering on between the units of the minimum (Fig. 6 (b)).

6. The training data (N_1 and N_2) classified into both sides of the border were used as new training data; processing described above was repeated recursively. Consequently, the hierarchic structure of a SOM was generated.
7. The lowermost hierarchy of the hierarchic structure was defined as a facial expression category. Five visualized images were defined as representative images of each category after learning completion. Then the photographer of the facial expression images performed visual confirmation to each facial expression category and conducted implication in emotion categories.

2.3 Generation of facial expression map

It is considered that recognition to a natural facial expression requires generation of a facial expression pattern (mixed facial expression) that interpolates each emotion category. The proposed method used the representative image obtained in Section 2.2 as training data and carried out data expansion of facial expression patterns between each emotion category using CPN with a large mapping space. The reason for adopting CPN, a supervised learning algorithm, is that the teaching signal of training data is known by processing in Section 2.2. The mapping space of CPN has a greater number of units than the number of training data, and has a torus structure because it is presumed that a large mapping space allows CPN to perform data expansion based on the similarity and continuity of training data. Figure 7 depicts the CPN architecture to generate Facial Expression Map. The details of processing are described below.

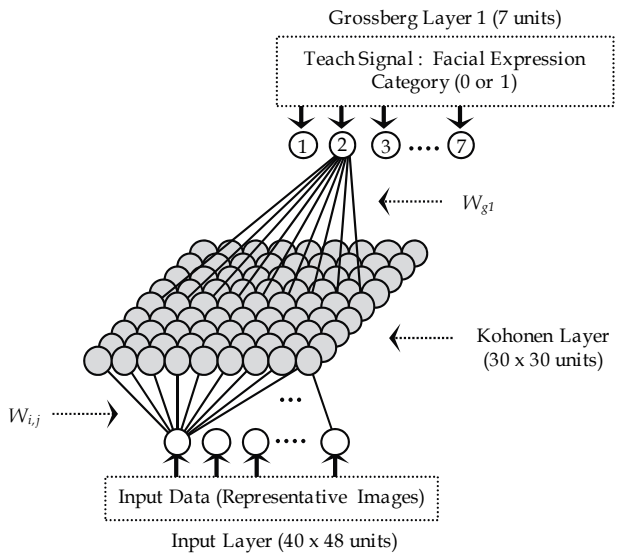


Fig. 7. CPN architecture for generation of Facial Expression Map

1. In fact, CPN has a structure comprising an input layer of 40 × 48 units and a two-dimensional Kohonen layer of 30 × 30 units. In addition, the Grossberg layer 1 of seven units was prepared, to which the teaching signal of six basic facial expressions and a neutral facial expression were input.
2. Representative images obtained in Section 2.2 were used as training data, and learning was carried out for each subject. As the teaching signal to the Grossberg layer 1, 1 was

input into units that mean emotion categories of representative images, otherwise 0. The number of learning was set to 20,000 times.

3. The weights (W_{g1}) of the Grossberg layer 1 were compared for each unit of the Kohonen layer after learning completion; an emotion category of the greatest value was used as the label of the unit. A category map generated by the processing described above was defined as a subject-specific Facial Expression Map.

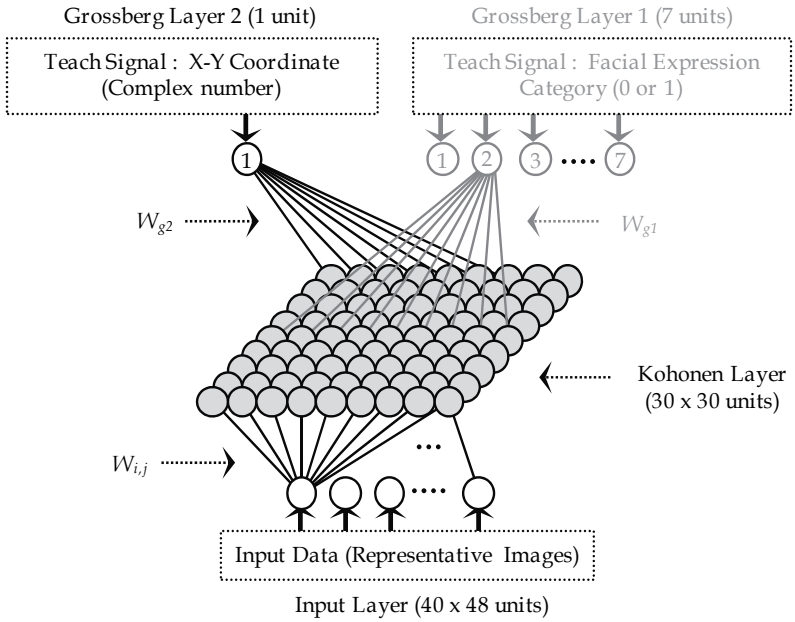
2.4 Generation of emotion map

Even if the facial expression pattern appearing on a face is peculiar to an individual, the internal emotion that humans express on the face and the emotion that humans recognize from the facial expression are considered to be person-independent and universal. Therefore, it is presumed necessary to match the grade of emotion based on a common index for each subject to the grade of change of facial expression patterns extended in Section 2.3. The proposed method is centered upon the Circumplex model of Russell (Russell & Bullock, 1985) as a common index. Specifically, the coordinate values based on the Circumplex model were input as teaching signals of CPN, in parallel to processing in Section 2.3. Then generation of an emotion feature space was tried, which matches the grade of change of facial expression patterns and the grade of emotion. Figure 8 depicts the generation procedure of Emotion Map. The details of processing are described as follows.

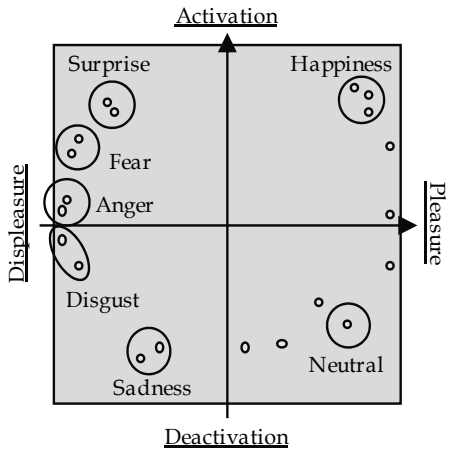
1. The Grossberg layer 2 of one unit that inputs the coordinate values of the Circumplex model was added to the CPN structure (Fig. 8 (a)).
2. Each facial expression stimulus is arranged in a circle on a space centering on "pleasantness" and "arousal" in the Circumplex model (Fig. 8 (b)). The proposed method expresses this circular space as the complex plane depicted in Fig. 8 (c), and complex number based on the figure were input to the Grossberg layer 2 as teaching signals. For example, when an inputted training data is an emotion category of happiness, a teaching signal for Grossberg layer 2 is $\cos(\pi/4) + i \sin(\pi/4)$.
3. This processing was repeated to the maximum learning number.
4. Each unit of the Kohonen layer was plotted onto the complex plane after learning completion based on the values of the real and imaginary parts of the weight (W_{g2}) on Grossberg layer 2. Then this complex plane was defined as a subject-specific Emotion Map.

3. Proposed method

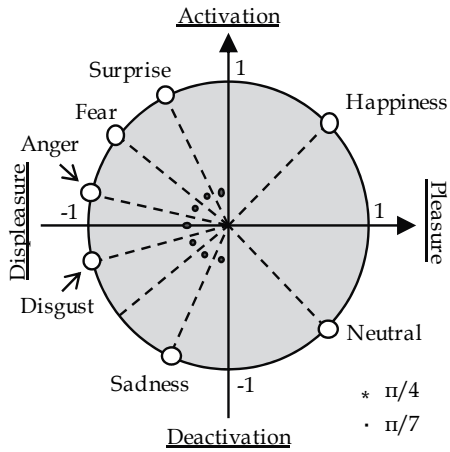
The facial expression feature space described in Section 2 above, Facial Expression map and Emotion Map, has generalization capability for facial expression images that have not been learned, but it has no learning capability for the facial expression images that are being added continually. From this perspective, we examined, specifically in our study, the algorithm of incremental learning capability, called Adaptive Resonance Theory (ART), which has characteristics of both stability and plasticity. ART is an unsupervised learning algorithm. When the matching level between the input data and the existing category data is lower than the vigilance parameter value provided in advance, it takes the input data to add as a new category of data. Actually, the input data used in the method we propose are the intensity of the facial expression images. For that reason, we used Fuzzy ART (Carpenter et al., 1991) in our study because it can accept analog inputs.



(a) CPN architecture.



(b) Circumplex model of Russell.



(c) Complex plane expression.

Fig. 8. CPN architecture for generation of Emotion Map

3.1 Fuzzy ART

Figure 9 shows the Fuzzy ART architecture. The Fuzzy ART is formed of two layers, the input layer F_1 and the output layer F_2. The quantities of neurons of the F_1 and F_2 layer are, respectively, M and N. Input I is an M-dimentional vector (I_1, ... I_M), where each component I_i is in the interval [0, 1]. A neuron of layer F_2 represents one category and is caracterized by its weight vector $W_j \equiv (W_{j1}, \ldots, W_{jM})$. Fuzzy ART dynamics are determined by a choice parameter a ($a > 0$); a learning rate parameter β ($0 \le \beta \le 1$); and a vigilance parameter ρ ($0 \le \rho \le 1$). The learning algorithm is described below.

1. Initially, each category is uncommitted,

$$w_{j1} = \ldots = w_{jM} = 1 \quad . \tag{1}$$

2. For each input I and each category j, the choice function T_i is defined by

$$T_j(I) = \frac{\left| I \wedge w_j \right|}{\alpha + \left| w_j \right|} \ , \tag{2}$$

where the fuzzy AND operator are defined by

$$\left(x \wedge y\right)_i \equiv \min\left(x_i, y_i\right) \ , \tag{3}$$

and where the norm is defined by

$$\left| x \right| \equiv \sum_{i=1}^{M} \left| x_i \right| \ . \tag{4}$$

3. The category choice is indexed by J, where

$$T_J = \max \left\{ T_j : \ j = 1 \ldots N \right\} \quad . \tag{5}$$

If more than one T_j is maximal, the smallest index is chosen.

4. Resonance occurs if the match function of the chosen category meets the vigilance criterion. Resonance occurs when

$$\frac{\left| I \wedge w_J \right|}{\left| I \right|} \ge \rho \quad . \tag{6}$$

Reset occurs when

$$\frac{\left| I \wedge w_J \right|}{\left| I \right|} < \rho \ . \tag{7}$$

5. Next layer F_2 wining nodes, T_J is inhibited for the duration of the input representation to prevent it from competing further. A new index J is then chosen by (5). The search

process continues until the chosen J satisfies (6). Ones equation (6) is fulfilled, the weight vector W_J is updated according to the equation

$$w_J^{(new)} = \beta \left(I \wedge w_J^{(old)} \right) + \left(1 - \beta \right) w_J^{(old)} \tag{8}$$

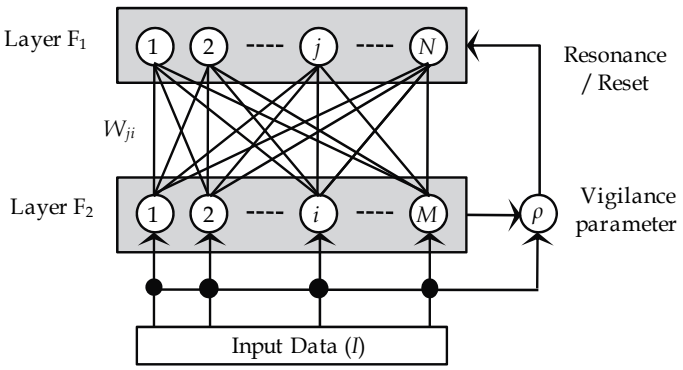


Fig. 9. Fuzzy ART architecture

3.2 Improvement to incremental learning capability of ART

Fuzzy ART creates a new category without fail when the matching function value between the input data and the existing category data is lower than vigilance parameter. In other words, when a facial expression that differs greatly from any existing one is fed in, a new category is created, which results in addition of further redundant knowledge in the F_2 layer. Consequently, in our study, the results of which are described in Fig. 10, we set two different vigilance parameters in Fuzzy ART, ρ_1 and, ρ_2 ($\rho_1 > \rho_2$). Thereby, the judgment made by the Fuzzy ART can be branched to any of the following three:

a. Update of the existing category data (Matching function value $\geq \rho_1$): When the input data are part of the existing category, weight of existing category gets updated (i.e. an existing knowledge update).
b. Addition of a new category ($\rho_2 \leq$ Matching function value $< \rho_1$): When the input data are similar to the existing category, they are newly added into that category as new facial pattern data (i.e. a new knowledge addition).
c. Rejection (Matching function value $< \rho_2$): When the input data differ from any existing category to a great degree, no incremental learning process is accomplished (as described in (a) or (b) above); rather, it is simply rejected.

Through the processing described above, the Fuzzy ART additionally learns only the facial expression that is similar to existing category data. Consequently, it can expand the level of the knowledge it holds little by little. It also suppresses creation of any redundant knowledge by itself.

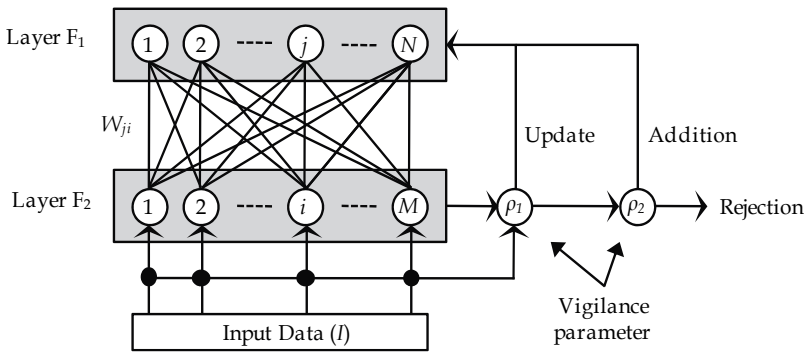


Fig. 10. Improvement of Fuzzy ART

3.3 Facial expression recognition model with adaptive learning capability

Figure 11 depicts the procedures of the proposed method. The method consists of a facial expression feature space that is created by the CPN and Fuzzy ART, which are linked with each unit of the feature space (Kohonen layer of CPN). Details of proposed method are explained below.

1. The processing described in Section 2 above creates a facial expression feature space using the initially created training data (CPN learning).
2. The weight W_{CPN} for each unit of the facial expression feature space is set after the learning process progresses, with it as the weight W_{ART} of the initial category of the Fuzzy ART.
3. Using the test data fed into the facial expression feature space, it searches the feature space for the winner unit whose Euclid distance to the weight of each unit is the least of all.
4. With the test data fed into the Fuzzy ART linked with the winner unit, it performs the processes described in 3.1 and 3.2 above and moves on to determine which process to take, choosing from either (a) an existing knowledge update, (b) a new knowledge addition, or (c) rejection. If either (a) or (b) is chosen, then that associated emotion category of the facial expression feature space becomes the finally determined recognition resulting from the test data that were entered. If it is (c), then the test data differ greatly from the feature of the existing category data so that the recognition is undetermined.
5. The method repeats the processes of (3) and (4) for some period. Then it moves on and sets, as new training data, the weight of the associated category unit that the Fuzzy ART holds, and performs re-learning for the facial expression feature space similarly to process (1). Through this series of facial expression recognition processes, the facial expression feature space can acquire new knowledge after the re-learning process is accomplished.

Repeating the processes of (3), (4), and (5) above, the facial expression feature space can additionally learn new knowledge in parallel to the process of recognizing the facial expression in the feature space while it holds existing pieces of knowledge continuously as they are.

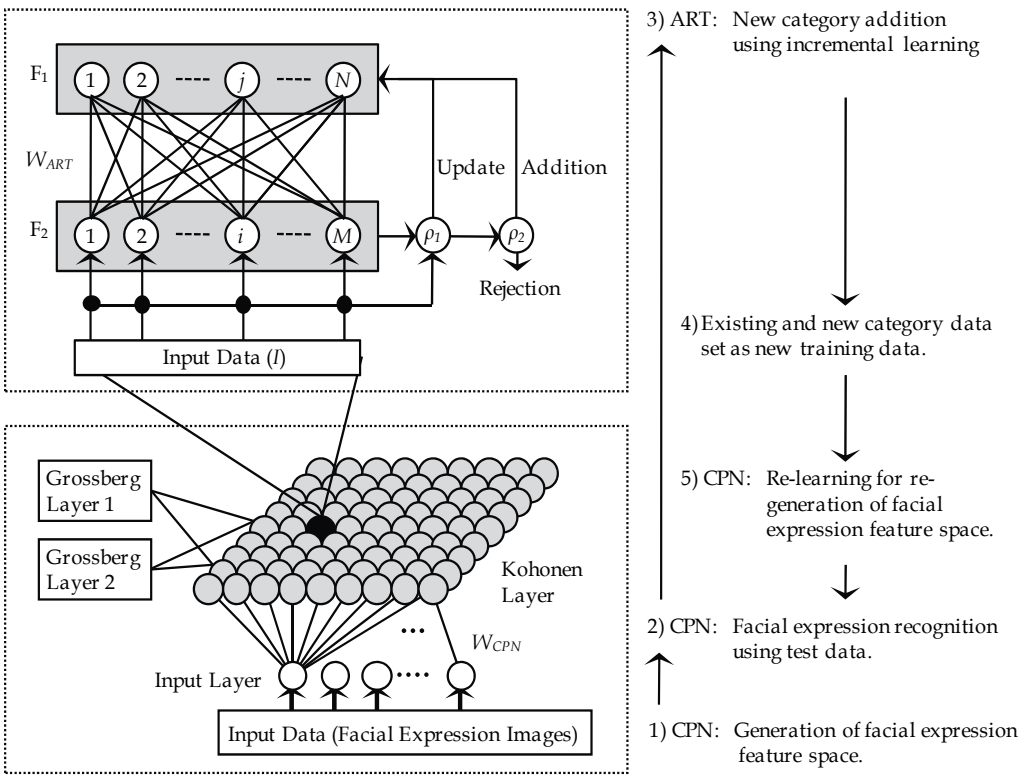


Fig. 11. Procedures of the proposed method

4. Experiment for evaluation

In our study, we conducted an experiment to evaluate the method we proposed, particularly addressing the incremental learning capability of the method. In practice, we had emotion categories of two kinds, which included joyful expression (happiness) and no facial expression (neutral); we also performed an experiment to evaluate the additional learning ability of the method with respect to joyful expression patterns. The feature space size is 30 units (one dimension). The vigilance parameters of the Fuzzy ART were ρ_1 = 0.98, and ρ_2 = 0.96, each of which was set empirically. Details of experiments are explained below.

1. We first created a facial expression feature space using initial training data of as many facial expression image pictures as given on 100 (which included 50 images of "joyful expression of a person with his mouth shut" and another 50 of "no facial expression").
2. We accomplished the facial expression recognition and the incremental learning using 200 images of the additional data that were fed in (which included joyful expressions with the person's mouth opening gradually). Subsequently, we conducted the re-learning process for those associated facial expression feature spaces. We repeated this series of processes 10 times.
3. We conducted facial expression recognition for as many test data as on the 2635 images fed in (which were of the additionally obtained joyful expressions) using the 10 feature spaces created in (2) above.

During the experiment, we used a USB camera to capture facial expression images. We used those images after applying a normalizing process to each described in Section 2.1 above.

5. Experimental results and discussion

5.1 Incremental learning capability of facial expression feature space

We next examine the experimental results we obtained from (2) of Section 4 above. Figure 12 portrays the number of occurrences of the additional learning process made for each re-learning of the associated facial expression feature space (which include existing category updates, new category additions, and rejections). For the feature space created at the initial stage of the whole process, we had existing category updates 62 times, and new category addition 25 times out of 200 images of additional learning data, all of which indicate that the facial expression recognition rate we obtained was 44%. These values of category update and addition increased gradually as the re-learning processes were repeated five times. Continuing that, the recognition rate we obtained at the 10th time of re-learning process improved to 88% (with 131 updates and 45 new additions). Conversely, the number of rejections we had decreased. The rejection was processed about 20 times serially from the 5th time to the 10th time of re-learning. One reason is that the face portions of the image data we used were slightly shifted in position, or that the facial expression data that had been entered differed greatly from the existing knowledge we had (i.e., from those facial expression patterns that had been held in the facial expression feature space).

Figure 13 presents occurrences in terms of the Euclidean distance between the winner unit of the facial expression feature space and the additional learning data that were fed in. The shorter this distance, the more precisely the level of recognition was proved to have been gained. Similarly, Table 1 shows the associated average and variance values of the Euclidean distances given on Fig. 13. Although the distances and the variances are all showing large values in the feature space at its initial stage, they are decreasing and converging as the re-learning process progresses.

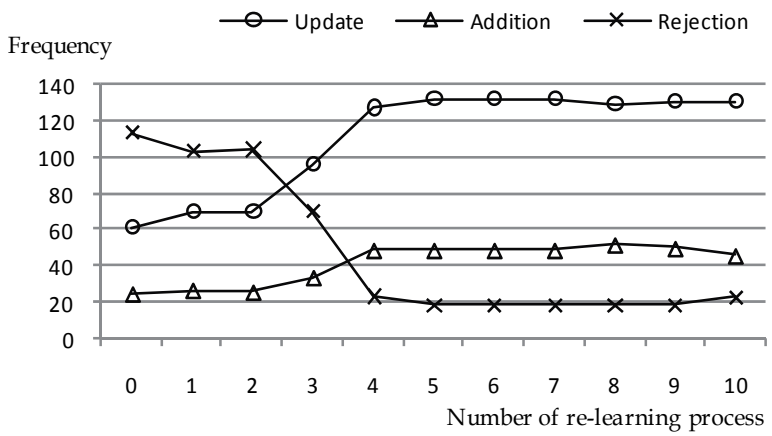
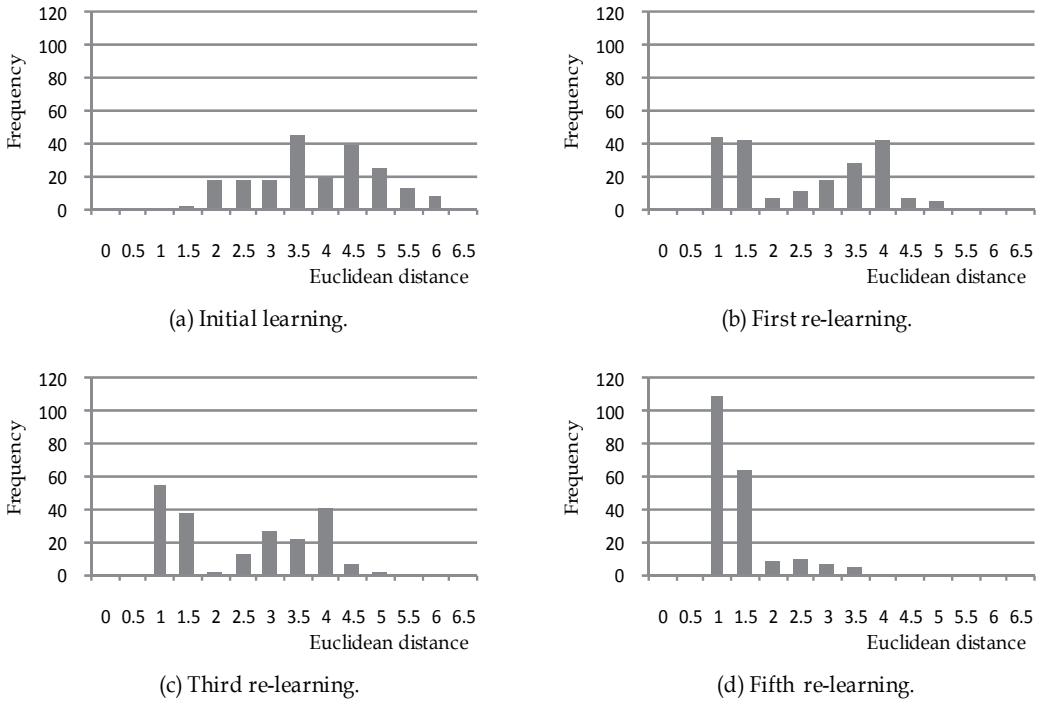


Fig. 12. Frequency of incremental learning process (200 addtional learning images)



(a) Initial learning. (b) First re-learning.

(c) Third re-learning. (d) Fifth re-learning.

Fig. 13. Frequency of the Euclidean distance between the weight of winner unit on the facial expression feature space and the additional learning data (200 images)

Number of re-learning process	0	1	2	3	4	5	6	7	8	9	10
Average value	3.63	2.33	2.18	2.19	1.44	1.15	1.16	1.16	1.19	1.17	1.17
Variance value	1.16	1.60	1.44	1.54	0.54	0.33	0.34	0.35	0.33	0.35	0.35

Table 1. Average and variance values of the Euclidean distance shown Fig. 13

Figure 14 portrays results of visualizing the weight of the facial expression feature space for up to the fifth iteration of the re-learning process. Looking at the visualized images of the facial expression feature space in Fig. 14, we recognize that the greater number of facial patterns of the person with his mouth gradually opening is certainly captured as the number of re-learning processes increases. In addition, it is readily understood is that the person's joyful expression with his mouth shut and the patterns of his lack of facial expression contained in the initial learning data are retained as existing knowledge just as they are, without changing, even as the re-learning process progresses.

Those results given and described above show that the facial expression feature space is learning to add new knowledge while the facial expression recognition process progresses with existing knowledge kept as it is without changing.

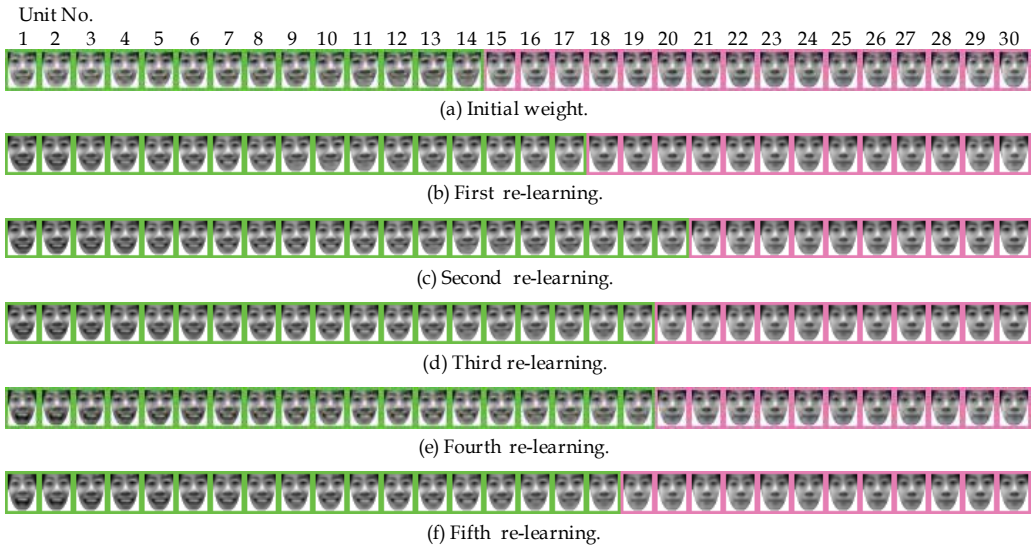


Fig. 14. Visualised images of the weight on the facial expression feature space

5.2 Experiment for evaluation using unlearning data

Figure 15 presents results of facial expression recognition when as many as 2635 images are fed into the facial expression feature space at each re-learning step. Figure 16 shows occurrences of each Euclidean distance taken by each winner unit of the feature space when the test data are fed in. Table 2 shows the average and variance values of the Euclidean distance. Examination of Fig. 15 reveals that the recognition rate for the initial feature space is 44% (with 870 times of update and 300 times of new addition), but it increases to 59% (with 1120 times of update and 430 times of new addition) at the 7th re-learning step of the feature space with the re-learning process examined. Investigation of Fig. 16 and Table 2 shows that the average and variance values of the Euclidean distance tend to decrease and converge as the re-learning process progresses in the same way as that shown in the experiment of Section 5.1.

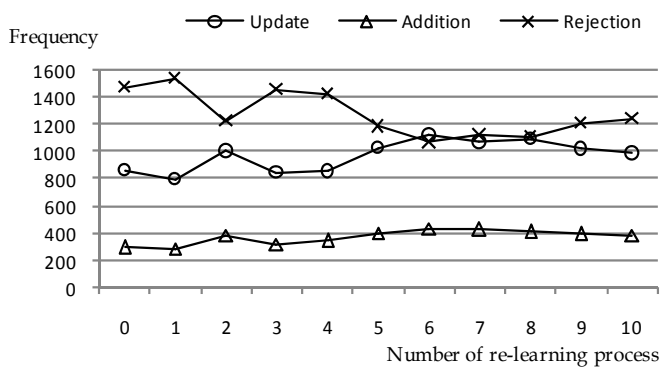
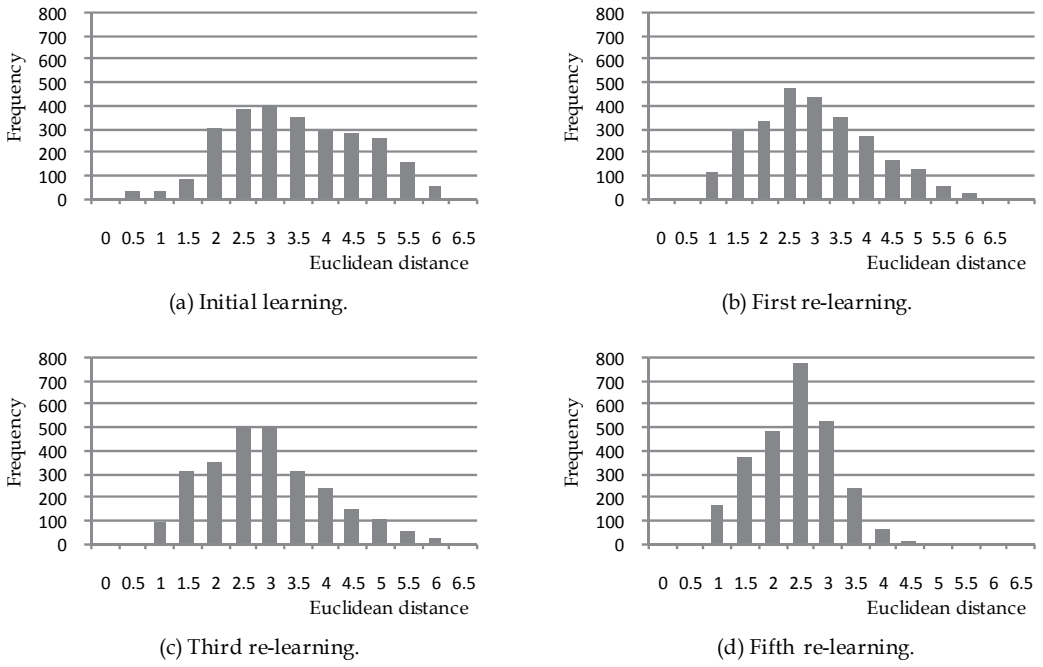


Fig. 15. Frequency of incremental learning process (2635 test images)

All of the points described above reveal that the facial expression feature space is capturing new knowledge without fail as it proceeds in parallel with the re-learning process. Results demonstrate that the method we propose is practical for use as an adaptive facial expression recognition model that has robustness along the time axis.



(a) Initial learning. (b) First re-learning.

(c) Third re-learning. (d) Fifth re-learning.

Fig. 16. Frequency of the Euclidean distance between the weight of winner unit on the facial expression feature space and the test data (2635 images)

Number of re-learning process	0	1	2	3	4	5	6	7	8	9	10
Average value	3.20	2.72	2.62	2.66	2.42	2.15	2.20	2.21	2.22	2.20	2.21
Variance value	1.45	1.26	1.13	1.14	0.69	0.51	0.54	0.54	0.55	0.53	0.53

Table 2. Average and variance values of the Euclidean distance shown Fig. 16

6. Conclusion

In this chapter, we proposed a facial expression recognition model with the adaptive learning capability. To demonstrate how useful it can be, we conducted some basic experiments for evaluation purposes with both joyful expressions and no facial expressions examined. Results show clearly that the facial expression feature space that we created using the method we proposed was capable of capturing additional knowledge while it maintained the existing knowledge as it was without alteration. The results we obtained demonstrate that the proposed method can be useful for an adaptive facial expression

recognition model that has a robustness feature along the time axis. We plan to conduct additional evaluative experiments in the future and examine the six basic facial expressions available. Moreover, we are planning to continue our work based on those facial expression images that we have accumulated during the long period of time we have continued this study.

7. Acknowledgment

This work was supported in part by Ministry of Education, Culture, Sports, Science and Technology (MEXT) Grants-in-Aid for Young Scientists (B): No. 18700192 and No. 20700174.

8. References

Akamatsu, S. (2002a). Recognition of Facial Expressions by Human and Computer [I]: Facial Expressions in Communications and Their Automatic Analysis by Computer, *The Journal of the Institute of Electronics, Information, and Communication Engineers*, Vol.85, No.9, pp.680-685. (in Japanese)

Akamatsu, S. (2002b). Recognition of Facial Expressions by Human and Computer [II]: The State of the Art in Facial Expression Analysis-1; Automatic Classification of Facial Expressions, *The Journal of the Institute of Electronics, Information, and Communication Engineers*, Vol.85, No.10, pp.766-771. (in Japanese)

Akamatsu, S. (2002c). Recognition of Facial Expressions by Human and Computer [III]: The State of the Art in Facial Expression Analysis-2; Recognition of Facial Actions, *The Journal of the Institute of Electronics, Information, and Communication Engineers*, Vol.85, No.12, pp.936-941. (in Japanese)

Akamatsu, S. (2003). Recognition of Facial Expressions by Human and Computer [IV: Finish]: Toward Computer Recognition of Facial Expressions Consistent with the Perception by Human, *The Journal of the Institute of Electronics, Information, and Communication Engineers*, Vol.86, No.1, pp.54-61. (in Japanese)

Carpenter, G.A.; Grossberg, S. & Rosen, D.B. (1991). Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System, *Neural Networks*, 4, pp.759-771

Fasel, B. & Luettin, J. (2003). Automatic Facial Expression Analysis: A Survey, *Pattern Recognition*, Vol.36, pp.259-275.

Gross, R. (2005). Face Databases, *Handbook of Face Recognition*, S.Li and A.Jain, ed., Springer-Verlag.

Ishii, M.; Sato, K.; Madokoro, H. & Nishida, M. (2008a). Generation of Emotional Feature Space based on Topological Characteristics of Facial Expression Images, *2008 IEEE Int. Conf. Automatic Face and Gesture Recognition* (FG2008), All the 6 pages (CD-ROM)

Ishii, M.; Sato, K.; Madokoro, H. & Nishida, M. (2008b). Extraction of Facial Expression Categories and Generation of Emotion Map Using Self-Mapping, *The IEICE Trans. Infomation and Systems* (Japanese Edition), Vol.J91-D, No.11, pp.2659-2672.

Kohonen, T. (1995). *Self-Organizing Maps*, Springer Series in Information Sciences.

Lienhart, R. & Maydt, J. (2002). An Extended Set of Haar-like Features for Rapid Object Detection, *Proc. IEEE Int. Conf. Image Processing*, Vol.1, pp.900–903.

Nielsen, R.H. (1987). Counterpropagation Networks, *Applied Optics*, vol.26, No.23, pp.4979-4984.

Pantic, M. & Rothkrantz, L.J.M. (2000). Automatic Analysis of Facial Expressions: The State of the Art, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.22, No.12, pp.1424-1445.

Pantic, M.; Valstar, M.F.; Rademaker, R. & Maat, L. (2005). Webbased Database for Facial Expression Analysis, *Proc. IEEE Int. Conf. Multimedia and Expo*, pp.317-321.

Russell, J.A. & Bullock, M. (1985). Multidimensional Scaling of Emotional Facial Expressions: Similarity from Preschoolers to Adults, *J. Personality and Social Psychology*, Vol.48, pp.1290–1298.

Tian, Y.L.; Kanade, T. & Cohn, J.F. (2001). Recognizing Action Units for Facial Expression Analysis, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.23, No.2, pp.97-116.

9

Self-Organization and Aggregation of Knowledge

Koichiro Ishikawa[1,2], Yoshihisa Shinozawa[2] and Akito Sakurai[2]
[1]*Aoyama Gakuin University*
[2]*Keio University*
[1,2]*Japan*

1. Introduction

Self-organization refers to formation processes of complex structures at the global level, which emerges solely from interactions among lower-level components without top-down controls or orders (Camazine et al. (2003)). Such processes are observed in many research areas (Johnson (2002); Buchanan (2007)), such as chemistry (e.g., Zhabotinsky (1991)), physics (e.g., Bak et al. (1988)), biology (e.g., Kauffman (1995)), brain and neuroscience (e.g., Haken (1996)), and economics (e.g., Krugman (1996)), etc.
On the other hand, phenomena, so-called "wisdom of crowds" (Surowiecki (2004)) or "collective intelligence" (Page (2007)) are also seen on the Internet, especially within the Web 2.0 (O'Reilly (2005)) environment (Tapscott & Williams (2006)). The mechanism for producing these phenomena has not yet been fully elucidated. But some of these phenomena have the same characteristics as self-organization, because individual user's diverse and spontaneous behavior or opinion aggregates to meaningful knowledge from the bottom up (Bonabeau et al. (1991)).
In the research area of computer science, a method called self-organizing map (SOM) has been developed and utilized (Kohonen (2001)). SOM is one of the neural network systems, which models the column structure of visual cortex (V1) in animal brains (Van Hulle (2000)) and learns based on an unsupervised machine learning algorithm. SOMs have been applied to problems in many fields such as data visualization, dimensional reduction, clustering, etc.
In this chapter, we focus on the multi-media content broadcasted on the Internet with the streaming technique. To serve most users by utilizing the bandwidth of the Internet, it is ideal to broadcast only important portions of the content that viewers really want to watch. If this is realized, viewers can avoid wasting their time by watching the portions that they do not desire to watch. Moreover, by not broadcasting less important portions unnecessary network traffic is decreased.
With the spread of broadband networks, it has become very popular to broadcast so-called, "rich" multi-media content including movies (e.g., *YouTube* (n.d.)). Therefore it is very urgent and meaningful to develop technology for sorting out important portions of the content.
In view of the current technological status, though, automatic recognition of importance is difficult to realize, since it is closely connected with meaning of the content. The difficulty is also obvious from the fact that almost all the search engines of Web pages (e.g., *Google* (n.d.)) provide the results based mainly on strings of characters and not their meaning. The reason why this method works is because we use words that reflect our intentions and the words are easy to recognize automatically. However, such methods cannot be applied to multi-media

content because in their expression, the meaning is implicit to the objects in them which are difficult to recognize.
The technique we propose in this chapter is to provide summary of viewing history of users who watched a particular multimedia content to users who are going to watch the content (Ishikawa et al. (2007)). To provide the summary of viewing history, the proposed technique aggregates users' behavior of content viewing. Our claim is that the resulted aggregation is in fact "wisdom of crowds" or "collective intelligence" is proven by experiments where new users have shortened their time to determine the important portions of the content. For the aggregation, based on similarity of learning mechanism of SOM and the aggregation process, we adopted two kinds of neural networks that are derived from SOM. We also reported on the experiments that were successful and that the implemented SOM worked efficiently in time and space.
We propose in this chapter two SOM-like algorithms that accept online, as input, the start and end of viewing of a multimedia content by many users; a neural network is finally self-organized, providing an approximation of the density distribution showing how many users see a part of a multimedia content. In this way "viewing behavior of crowds" information is accumulated as experience accumulates, summarized into one SOM-like network as useful knowledge for viewing is extracted, and is presented to new users as the knowledge is transmitted (See section 3.1.1 more detail).
Accumulation of multimedia content on the Internet increases the need for time-efficient viewing of the content and the possibility of compiling information on many users' viewing experiences. Under these circumstances, some methods have been proposed (e.g., Yu et al. (2003); Ishikawa et al. (2007)) that presents, in the Internet environment, a kind of summary of viewing records of many viewers of a certain multimedia content. The summary is expected to show that some part is seen by many users, but some other parts are rarely seen. The function is similar to websites utilizing "wisdom of crowds" and is facilitated by our proposed algorithm.
Our proposed methods are automatically and adaptively detect relatively important parts in multimedia content. The important parts of content should characterize the entire content and also be regarded as its summary. By applying our method, users can capture important screenshots of the content and also extract a set of important parts that contain characteristic scenes.
This chapter is organized as follows; section 2 introduces SOM. In section 3, each of the two methods proposed in this chapter are explained, the experiments performed for evaluating the effectiveness of the proposed methods, and the results are reported. Finally section 4 concludes this paper.

2. Self-organizing map

This section explains a self-organizing map (SOM) which is regarded as the most common process and concept which is referred to as self-organization. SOM is a method of unsupervised machine learning proposed by Kohonen, which is one of the neural networks that carries out competitive learning. SOM has been utilized for various purposes, such as function approximation, data clustering, and dimensional reduction of high-dimensional data.
The process of one-dimensional SOM learning is as follows. First, n neurons are allocated on a one-dimensional line. Without loss of generality, a line segment $[0, M]$ is considered, which is refered to as a line hereafter for simplicity. The position of each neuron, $y_i(t), 1 \leq i \leq n$ is

referred to as a feature vector. At an initial time $t = 0$, there is basically no difference in locating the neurons randomly or regularly (however it has been reported that, if many neurons are at the same initial positions, delay of convergence and deterioration in quality may occur). It has also been reported that the approximation accuracy obtained depends on the number of neurons.
Next, the network input $x(t)$ is presented to the SOM network. The neuron $y_i(t)$ with the shortest distance to $x(t)$ becomes the winner. Euclidian distance is commonly used for measuring the distance. The winner is then output.
After outputting, the network learning is done based on the input $x(t)$. First, neighborhood neurons with distance to the winning neuron shorter than the predefined threshold are selected as target neurons for learning where the distance is usually the same as used for selecting the winner. The distance may be determined differently from the winning neuron. Furthermore, the threshold used to judge the neighborhood neuron has often been decreased gradually according to the number of inputs presented to the network (i.e., the range of neighborhood neurons is narrowed down gradually over time). The feature vectors $y_i(t)$ for the winning and neighborhood neurons are updated by the following formula based on the network-input $x(t)$;

$$y_i(t+1) = y_i(t) + \alpha\big(x(t) - y_i(t)\big). \tag{1}$$

Here, α is the learning coefficient. It is also common to change this value in accordance with the presented number of network-inputs.
As is seen in Equation 1, as the result of all the above mentioned processes, neurons are re-located on the line according to the observation frequency of network input; namely, there is a dense re-location of neurons for parts of the content where a high frequency of input was observed, while there was a sparse re-location of neurons for parts of the content where a low frequency of input was observed. Therefore, the network approximates the probability distribution of inputs with the granularity that corresponds to the number of the network neurons (section 3.3.1 gives a detail analysis on the functions of the SOM algorithm).

3. Methods and results

3.1 Problem settings

3.1.1 Overview

Users usually view multimedia content such as video sequentially. Let us call the content viewed sequentially as time-series digital content hereafter. Sequential content viewing requires exactly the same time as the recording time. Since users need to spend long periods of time to view time-series digital content, they would be pleased if a method would make it possible for them to view only the important parts in a short amount of time. Here, we presume that there are parts that the users truly desire to view coexisting with parts that are insignificant, especially when the digital content has a long recording time. Therefore, the purpose mentioned above can be achieved if the importance of each part of the content could be estimated in some way.
A possible solution would be to use still images (called snapshots or thumbnails) to represent parts of time-series digital content. In fact many programs do this. One of the drawbacks of this method is that the intervals of taking snapshots are constant. In this case, the intervals between the shots are relatively spaced out in important parts whereas they are relatively dense in the insignificant parts.
Another possible solution, that in fact we adopted, is to turn to “wisdom of crowds” (Surowiecki (2004)) or “collective intelligence” which has been attracting much attention

(Ohmukai (2006)). Although "wisdom of crowds" has not yet been clearly defined, common features would be to record users' input, to show a kind of summary of the users' input, and to implicitly assume that the more users carry out a particular behavior, the more the behavior becomes valuable.
A "wisdom of crowds" website is considered to having the function of promoting knowledge sharing and developing by hopefully realizing phases of socialization and externalization of the SECI model proposed by Nonaka & Takeuchi (1995). It accepts many users' input, accumulates them, and presents their summaries. By looking at the summaries, the users might add new input. If the system works as desired, it helps knowledge to accumulate, to be shared, and to develop.
To realize the idea of "wisdom of crowds" in effective content viewing, we need a way to accumulate and summarize time-series content. Since the semantics of time-series content is hard to analyze, some new idea is necessary. We focus on the viewing frequency of each part of content. A system that records other users' viewing history and provides frequency about what other users view enables a new user to see how majority behaves, as with the above mentioned "wisdom of crowds" websites.
An actual system requires the function to reflect viewing operations of online users as well as to receive and deliver content on the Internet. The function should reflect promptly the input and operation carried out by each user, then such reflected information is used to facilitate user operation.
To handle vast quantities of data online, Ishikawa et al. (2007) proposed to record data on a relational database (RDB). If viewing behavior data of all users is recorded completely in a database, an accurate histogram can be obtained. However, considering the total quantity of multimedia content on the Internet, it is more costly than necessity to record all the viewing behavior data. If only smaller amount of data is recorded, the accuracy of the histogram degrades.
To solve the conflicts, we propose to use two SOM-like neural networks that successively approximate input data, and memorize the up-to-date results. Required data storage capacity is on the order of a fixed length of a real vector (array) of the neurons of the SOM in both of the two networks. Conventional SOM cannot be applied to the subject of this study. This is because the viewing history data does not represent viewing parts directly, but rather only the start and end points of the viewing. If representation of viewing parts could be obtained directly it would be possible to learn approximate viewing frequency with SOM, however, in fact only the starting and ending points of viewing is available. Consequently, we have developed two new algorithms and adopted them.

3.1.2 Sharing viewing history

This section explains summarization of multimedia digital contents through sharing viewing history. Ishikawa et al. (2007) proposed a system where online viewers of multimedia content transmit descriptions of where they viewed and skipped, as viewing history. The viewing history is recorded and processed on a server on the network (refer to Fig. 1). And then the server provides the processed summary to new viewers of the content (refer to Fig. 2).
It is likely that the parts that many viewers actually viewed indicate the important parts in the content. Viewers are able to reflect the summary of viewing history provided by the system on their viewing behavior, namely, they can utilize this information in order to view only the important parts of the content skipping other parts they deem unnecessary. In Ishikawa et al.

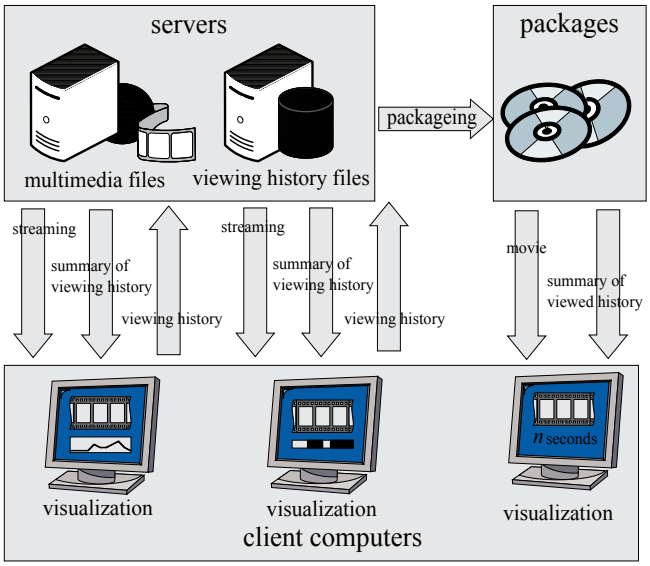


Fig. 1. Environment proposed in Ishikawa et al. (2007).

(2007), the data actually transmitted is a pair

$$R(t) = \langle position(t), operation(t) \rangle, \tag{2}$$

or, more precisely, either of the following:

$$R_{start}(t) = \langle position(t), 1 \rangle, \tag{3}$$
$$R_{stop}(t) = \langle position(t), -1 \rangle, \tag{4}$$

where t is the time when an event $R(t)$ occurs supposing that the time is so discretized that the two events do not occur at the same time, $position(t)$ is the position in a multimedia content where the viewing event starts or ends, operation is 1 (or -1) when viewing starts (or ends, respectively).
A position in time-series digital content is specified by the relative distance from its start in terms of replay time length, e.g., if the content is 10 minute length and the position is 2 minutes from its start, the position is said to be 0.2. $R_{start}(t)$ is sent when a play button is pressed, fast forward button is released, or slide bar is released after sliding; $R_{stop}(t)$ is sent when a stop button is pressed, fast forward button is pressed, or slide bar is moved first. The server stores $R_{start}(t)$ and $R_{stop}(t)$ in a RDB , according to the method proposed in Ishikawa et al. (2007). Furthermore, the server provides the following information to new viewers of the content.

$$C(p) = \sum_{t \in \{t | position(t) \leq p\}} operation(t). \tag{5}$$

This process is realized by executing the following SQL commands on the RDB of the content distribution server.

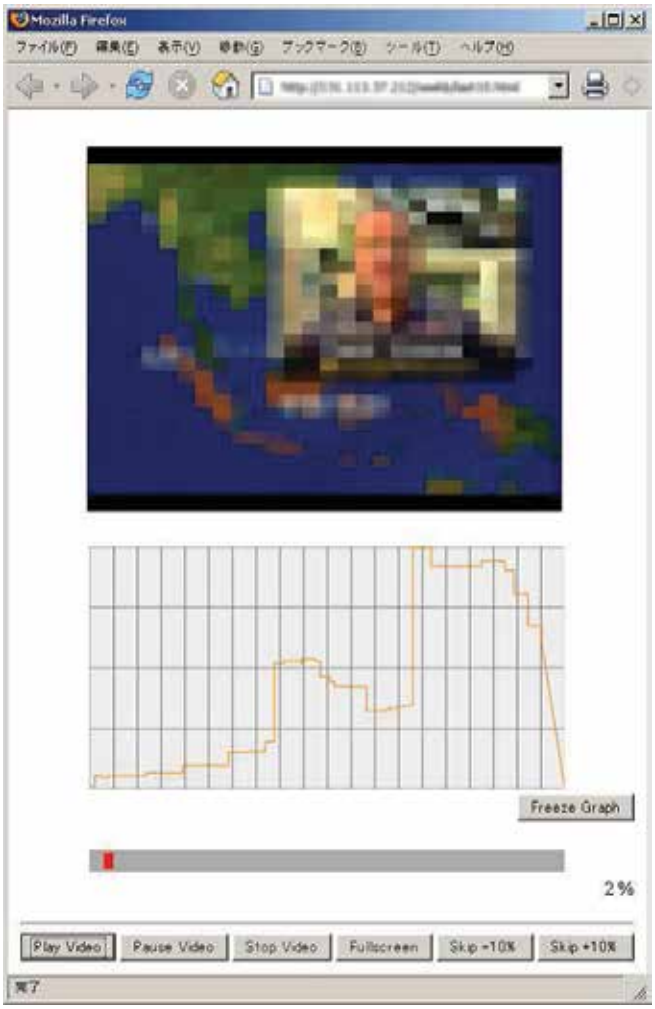


Fig. 2. A scrrenshot on a client computer proposed in Ishikawa et al. (2007).

```
CREATE VIEW tmpview AS
 SELECT position,
        sum(operation) AS count
   FROM history GROUP BY ROUND(position);
SET @i=0;
SELECT position, @i:=@i + count AS result
   FROM tmpview ORDER BY position;
```

The client computer that receives $C(p)$ draws a graph of $C(p)$ on a 2D plane (Fig. 2). From the graph, users are able to learn how frequently each part of the video content they are viewing was viewed, and to reflect it in their own viewing behavior, by skipping some parts of the content.

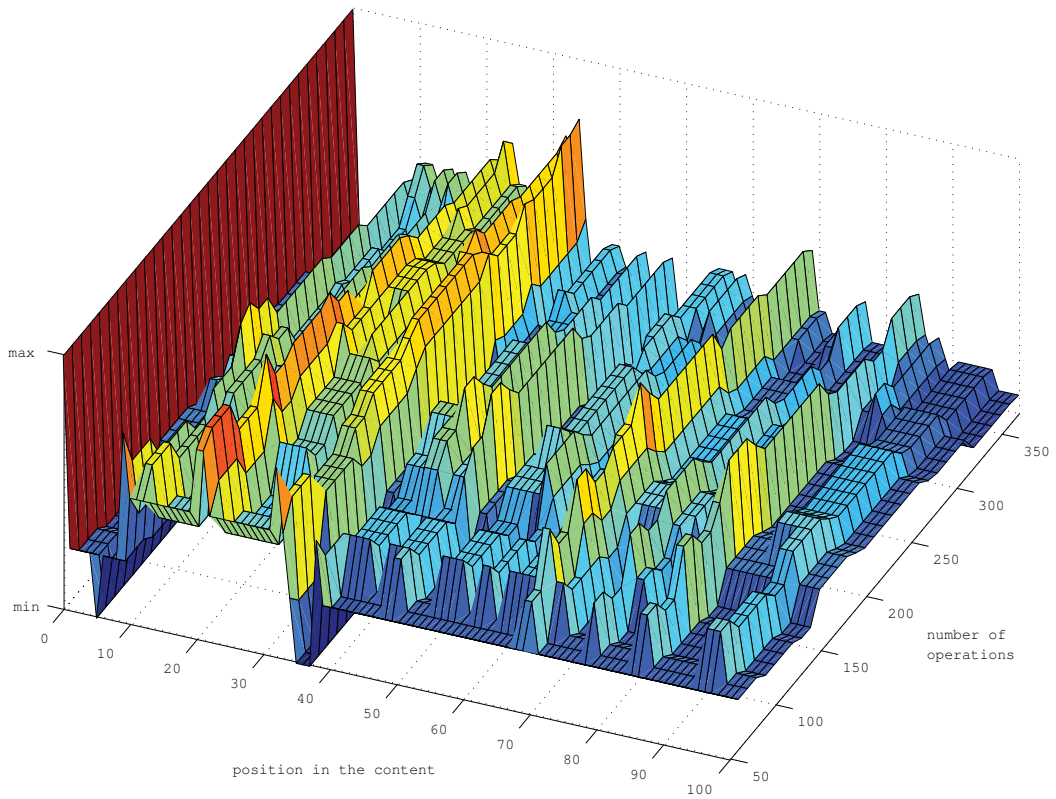


Fig. 3. Experimental result in Ishikawa et al. (2007). X axis shows position in a content. Z axis shows the viewed frequency at each X, i.e., X-Z graph is histogram of viewed frequency of the content at a moment. Y axis shows the number of operations. As the number of operations increases, the peak of histogram (X value of about 34) become distinctive (section 3.3.2.3 gives experimental details).

3.1.3 Online identification of summarizing points

This section proposes a method to identify content parts that characterize the entire content. Note that time-series digital content is in general considered to be composed of multiple parts. A part in this case, may be a still image of one scene, or time-series movie.

The method we considered is (1) to utilize viewing history described in section 3.1.2 and (2) to adopt SOM to summarize the viewing history. The reason why viewing history should be utilized is that the parts only a few user view will not characterize the whole content whereas the parts that many users view may characterize the whole. The reason why SOM should be adopted is that SOM learns frequency distribution of inputs as feature vectors of neurons gather to high frequency area of inputs and disperse from low frequency area of inputs. The reason why we combine them is that we could know important parts from where neurons gather and less important parts from where neurons dispers if SOM learns viewing history.

One additional merit of using SOM is that its space complexity is far less than that of RDB, since SOM is represented by feature vectors, i.e., an array of real numbers or integers. There are two disadvantages of using conventional SOM. One is that the conventional SOM accepts points as input, i.e., a scalar data, whereas since the viewing history is a list of range data, that

is, pairs of staring and ending of viewing, SOM must accepts range data as inputs. The other is that the starting and ending points of viewing do not necessarily come in a pair to a server computer as pointed out in Ishikawa et al. (2007).
To make matters worse, starting data and ending data from the same client computer do not necessarily have the same identification code, so that it is difficult to match the starting data and ending data from the same client computer. If we could assign identification code to the client computer, we could solve the problem. One possible ID is the IP address of the client computer, but IP addresses of many computers are assigned dynamically so that they may change between the starting of viewing and the end of viewing. The other possible ID is a cookie that would be set in a multimedia player and referred to by a server computer, but cookies by multimedia players are not popular and not standardized yet.
Since the advantages of SOM are indispensable, we devised two new methods, both of them consist of networks and their learning algorithm based on the conventional SOM. The proposed method described in section 3.2 has an SOM-like network that accepts starting points and ending points independently, that is, in any order without identifying its counterpart and learns viewing frequency distribution. The one in section 3.3 has two SOM-like networks, each of which accept one of starting and ending points and learn independently and one more SOM-like network that learns viewing frequency distribution from the former networks.

3.2 Method 1: with a single neural network

3.2.1 Overview

Our purpose is to recover frequency distribution of viewing events from their start and end events. In this section, we focus on equal density partition $x_0 < x_1 < \cdots < x_n$ of frequency distribution $p(x)$ such that $\int_{x_i}^{x_{i+1}} p(x)\,dx$ is a constant independent of i.
The proposed algorithm is shown in Fig. 4. Corresponding to the type of $position(t)$ and $operation(t)$ of network input (see Equation 2), the values of neurons, i.e., positions are updated (lines 10–31, 40). Since an update step α is a constant, a neuron might move past a neuron next to it. To prevent this, a neuron should maintain a certain distance (ε) from the neuron next to it (lines 32–39). Derivation of update formulae is as follows: Consider one-dimensional SOM-like network $\mathcal{X}$

$$\mathcal{X} = \langle x_1, \ldots, x_n \rangle,\ x_1 < x_2 < \cdots < x_n \in \Re^1.$$

If $\mathcal{X}$ is arranged such that for some c

$$\int_{x_i}^{x_{i+1}} p(x)\,dx = c,$$

then clearly $\mathcal{X}$ reflects the density function $p(x)$ in such a way that

$$\frac{c}{x_{i+1} - x_i} \approx p(x), \text{ for } x \in [x_i,\, x_{i+1}].$$

Suppose that $p(x)$ is sufficiently smooth and for simplicity the sign of $\partial p \,/\, \partial x$ does not change in (x_i, x_{i+1}). Then, by putting $y = p(x)$,

initialize network $\mathcal{Y} \leftarrow \mathcal{Y}^0 = \langle y_1^0, \ldots, y_{|\mathcal{Y}|}^0 \rangle, y_1^0 < y_2^0 < \cdots < y_{|\mathcal{Y}|}^0$
$t \leftarrow 0$
repeat forever
  $t \leftarrow t + 1$
  receive operation information $R(t) = \langle x(t), op(t) \rangle$
  $B = \langle b_0 \leftarrow 0, b_1 \leftarrow y_1, \ldots, b_{|\mathcal{Y}|} \leftarrow y_{|\mathcal{Y}|}, b_{|\mathcal{Y}|+1} \leftarrow \sup(\mathcal{Y}) \rangle$
  for $i \in \{1, 2, \ldots, |\mathcal{Y}|\}$
    $\Delta_i \leftarrow 0$
  end for
  if $op(t) = 1$ **then**
    for $i \in \{2, 3, \ldots, |\mathcal{Y}| - 1\}$
      if $x(t) < b_i$ **then**
        $\Delta_i \leftarrow 2y_i - b_{i+2} - b_i$
      elseif $x(t) < y_i$ **then**
        $\Delta_i \leftarrow 2y_i - b_{i+2} - x(t)$
      elseif $x(t) < y_i$ **then**
        $\Delta_i \leftarrow b_{i+2} - x(t)$
      end if
    end for
  else
    for $i \in \{2, 3, \ldots, |\mathcal{Y}| - 1\}$
      if $x(t) < b_i$ **then**
        $\Delta_i \leftarrow -(2y_i - b_{i+2} - b_i)$
      elseif $x(t) < y_i$ **then**
        $\Delta_i \leftarrow -\big(2y_i - b_{i+2} - x(t)\big)$
      elseif $x(t) < y_i$ **then**
        $\Delta_i \leftarrow -\big(b_{i+2} - x(t)\big)$
      end if
    end for
  end if
  $\mathcal{Z} = \langle z_1, \ldots, z_{|\mathcal{Y}|} \rangle \leftarrow \mathcal{Y} + \alpha\Delta$
  for $i \in \{2, 3, \ldots, |\mathcal{Y}| - 1\}$
    if $z_i \leq y_{i-1}$ **then**
      $z_i \leftarrow y_{i-1} + \varepsilon$
    end if
    if $z_i \geq y_{i+1}$ **then**
      $z_i \leftarrow y_{i+1} - \varepsilon$
    end if
  end for
  $\mathcal{Y} \leftarrow \mathcal{Z}$
end repeat

Fig. 4. Procedures of the proposed in section 3.2

$$
\begin{aligned}
& \int_{x_0}^{x_n} p(x)\,dx \\
= & \int_{p(x_0)}^{p(x_n)} y\,\frac{\partial p^{-1}}{\partial y}\,dy \\
= & \; p^{-1}(y)\,y\Big|_{p(x_0)}^{p(x_n)} - \int_{p(x_0)}^{p(x_n)} p^{-1}(y)\,dy \\
= & \; \big[x_n\,p(x_n) - x_0\,p(x_0)\big] \; - \sum_{i=0}^{n-1}\int_{p(x_i)}^{p(x_{i+1})} p^{-1}(y)\,dy \\
= & \; \big[x_n\,p(x_n) - x_0\,p(x_0)\big] \; - \sum_{i=0,\,x_i\le\exists x\le x_{i+1}}^{n-1} \big[p(x_{i+1}) - p(x_i)\big]x \\
= & \; (x_n - x_0)\,p(x_0) \; + \sum_{i=0,\,x_i\le\exists x\le x_{i+1}}^{n-1} \big[p(x_{i+1}) - p(x_i)\big]\,(x_n - x).
\end{aligned}
$$

Hereafter $p(x_0) = 0$ is assumed. If increment or decrement events, as Equation 3 or 4, $\Delta p(x)$ occur such that

$$\frac{\partial p}{\partial x}\,\Delta x \approx E[\Delta p(x)], \text{ for } x \in [x_0, x_n],$$

then

$$\int_{x_0}^{x_n} p(x)\,dx \approx E[\Delta p(x)\,(x_n - x)].$$

Therefore if we could arrange $\mathcal{X}$ such that for $x \in [x_i, x_{i+1})$

$$E[\Delta p(x)\,(x_n - x)]$$

is a constant independent of i, $\mathcal{X}$ is the one we want. From this we get the following update formulae for x_i

$$x_i \leftarrow x_i + \alpha\,\Delta x_i,$$

where

$$\Delta x_i = \begin{cases} \Delta p\,\big[(x_{i+1} - x_i) - (x_i - x_{i-1})\big], & (x < x_{i-1}) \\ \Delta p\,\big[(x_{i+1} - x_i) - (x_i - x)\big], & \big(x \in [x_{i-1}, x_i)\big) \\ \Delta p\,(x_{i+1} - x), & \big(x \in [x_i, x_{i+1})\big) \\ \quad 0, & (x \ge x_{i+1}), \end{cases}$$

and $\Delta p(x) = \Delta p$ for any x.

3.2.2 Experiments

We describe the results of experiments conducted to verify the proposed algorithm. The parameters were set for the experiments as follows;

- The number of neurons in the network $\mathcal{Y}$ (See line 1 in Fig. 5) is 41, and the neuron are initially positioned equally spaced between 0 and 100.
- The learning parameter α is fixed at 0.1.
- The parameter ε , the minimum separation between the neurons, is fixed at 0.01.

3.2.2.1 Single peak with straight slopes

We experimented on a single-peaked frequency distribution, which is a relatively simple example, as a viewing frequency distribution. The result is shown in Fig. 5.
To simulate such viewing history, the network input was given to the network with the following conditions:

- Viewing starts at the position p selected randomly from the range of positions 40 through 50 of content with 50% probability.
- Viewing ends at the position p selected randomly from the range of positions 75 through 85 of content with 50% probability.

The frequency of viewing operations is indicated by the solid line on the upper pane of Fig. 5. The horizontal axis is the relative position from the start of the content. The vertical axis indicates the sum of viewing operations, where the starting operation is 1 and the ending operation is -1 up to the position, thus $C(p)$ in Equation 5. The lower pane of Fig. 5 shows how the neuron positions in the network $\mathcal{Y}$ change as inputs are presented to the network. The change up to 10,000-th input is shown in the figure.
It shows that neurons gathered to the frequently-viewed part before 1,000 network-inputs. After that the neurons on x such that $p(x) = 0$ continued to be absorbed into the area $p(x) > 0$. The position of each neuron at 10,000-th inputs is plotted with circles overlapping on the upper pane of Fig. 5 where the relative vertical positions are not relevant. The plateau in this figure corresponds to high frequency of viewing, and neurons are located on these parts with gradual condensation and dilution on each side.

3.2.2.2 Double peaks with jagged slopes

Fig. 6 shows the result for a frequency distribution with double peaks with jagged slopes, which is more similar to practical cases. The axes in the figure are the same as Fig.5. In this experiment, neurons gathered at around two peaks, not around valleies after about 4,000-th inputs.

3.2.3 Discussion

In section 3.2 we focused on the utilization of a kind of "wisdom of crowds" based on observed frequency of viewing operations. "kizasi.jp" is an example of a site which utilizes "wisdom of crowds" based on word occurrence or co-occurrence frequencies which are observed in blog postings. Here words play the role of knowledge elements that construct knowledge.
Multimedia content has different characteristics than blogs, which causes difficulties. It is not constructed from meaningful elements. Even a state of the art technique would not recognize the meaningful elements in multimedia content.
A simple way to circumvent the difficulty is to utilize occurrences or frequency of viewing events for the content (Ishikawa et al. (2007)). But, since multimedia content is continuous, direct collection and transmission of viewing events are very costly. Since a viewing event consists of a start and an end point, we can instead use these and recover the viewing event.
In this section, we considered a new SOM-like algorithm which directly approximates the density distribution of viewing events based on their start and end points. We have developed a method based on SOM because SOM has an online algorithm, and the distribution of obtained neurons reflects the distribution of occurrence density of given data.
A clustering algorithm can also serve as a base algorithm for the problem. However, the problem that we want to solve is not to get clusters in viewing frequency but to present the overall tendency of viewing frequency to users.

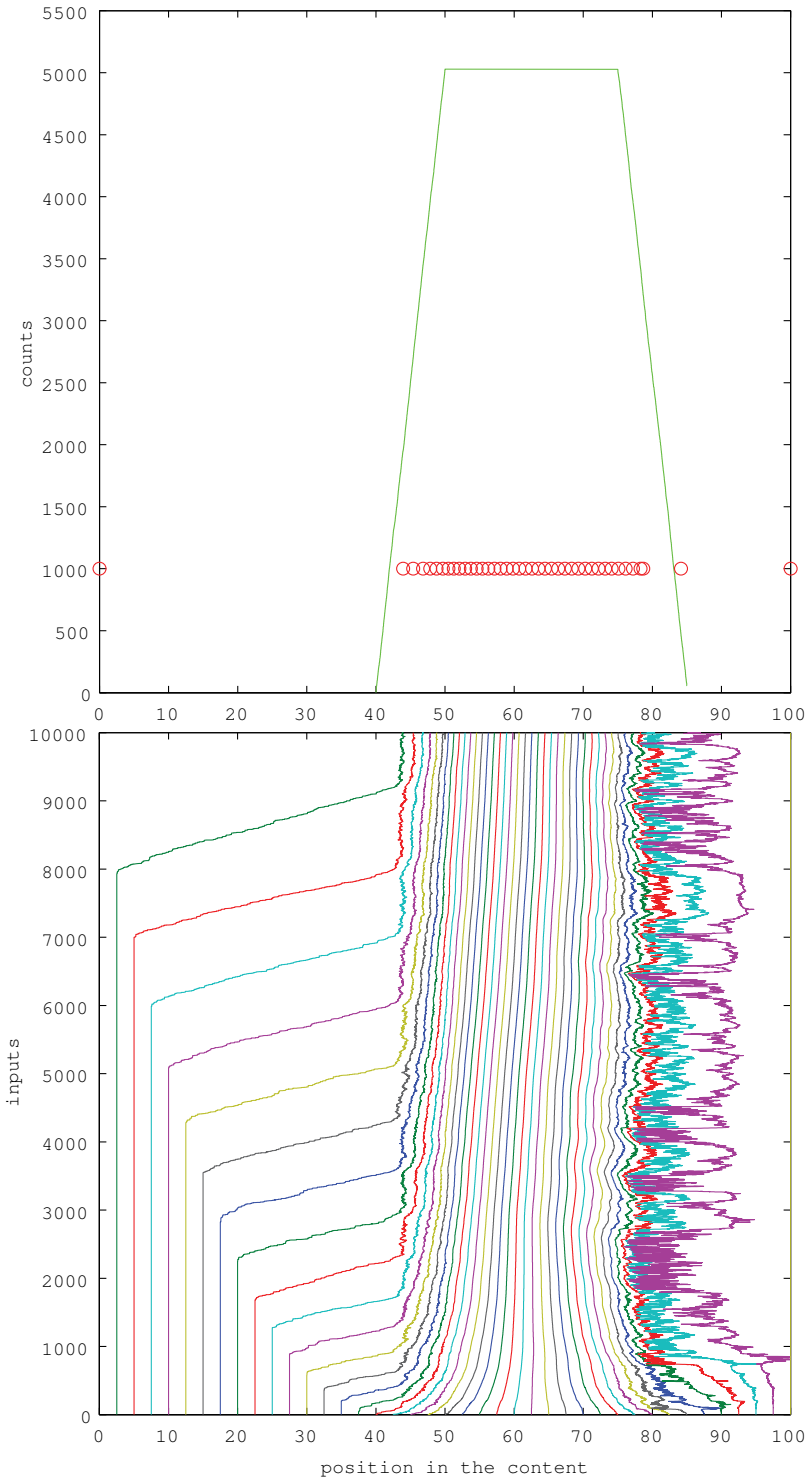


Fig. 5. Result of an experiment in section 3.2.2.1.

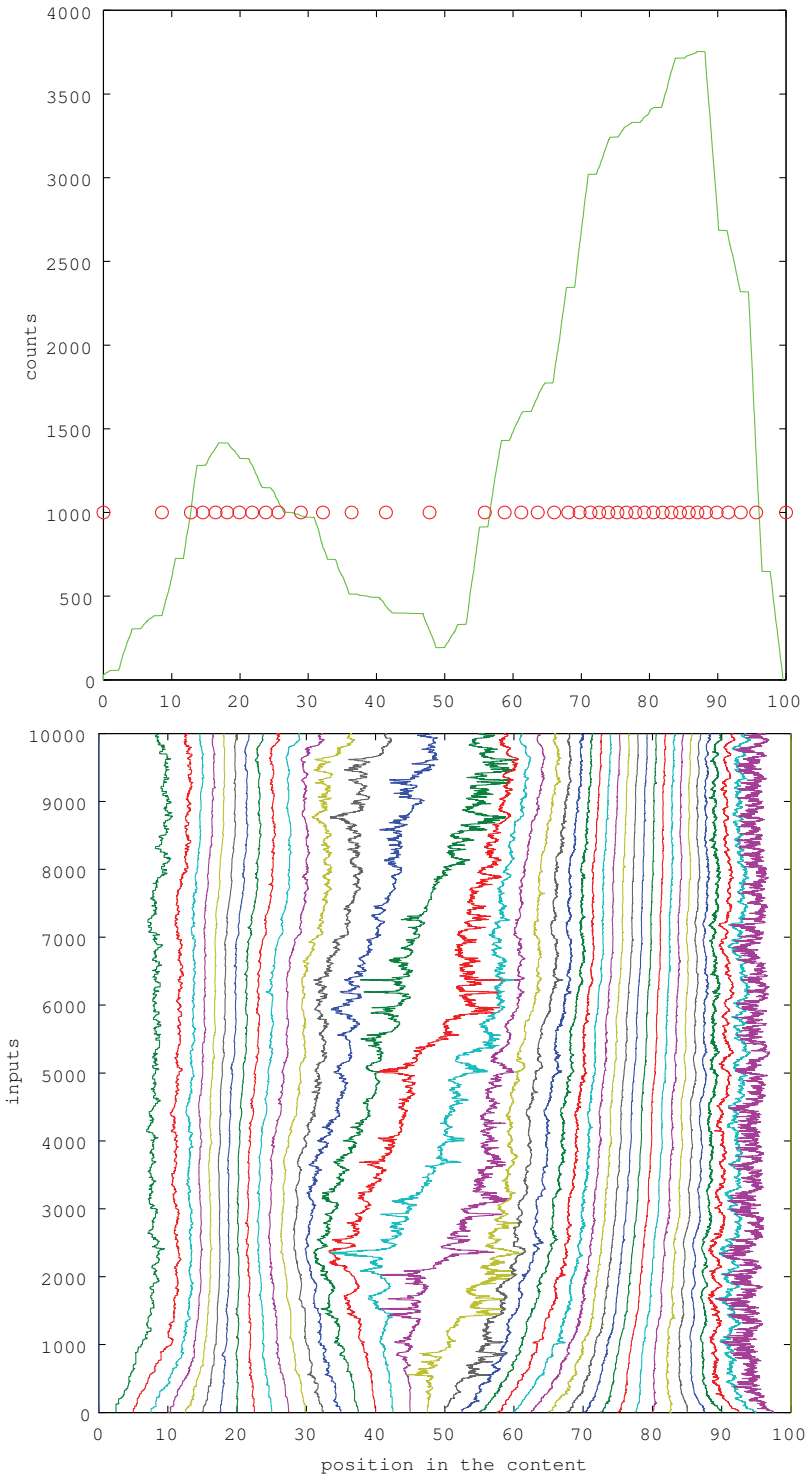


Fig. 6. Result of an experiment in section 3.2.2.2.

By applying the proposed algorithm, the computational complexity of time and space can be reduced substantially, compared with, for example, a method of recording all the viewing history data using RDB. Time complexity of an algorithm using RDB is as follows, where n is the number of histogram bins and corresponds to the number of neurons in our proposed algorithm.

1. $R(t) = \langle p, m \rangle$ is inserted into sorted array A which stores all the information (start and end points) received from users (see Ishikawa et al. (2007)):

   $\mathbf{\Omega}(\log t) + \mathbf{\Omega}(\log t)$

2. B, an array obtained from A by quantizing the time scale is calculated:

   $\mathbf{\Omega}(\log t)$

3. From $b_i = i \times \frac{\left(\max(B) - \min(B)\right)}{n} \in B$, $1 \leq i \leq n$, b_i is calculated:

   $\mathbf{\Omega}(\log t)$

On the other hand, the process of the algorithm proposed in this section does not require sorting (above in 1) and deciding the insertion location of data (above in 2), but requires the network learning process for each input observed data. Time complexity is calculated as follows;

4. $\mathrm{argmin}_i \left(\| p - y_i \| \right)$, in the network are calculated.

   $\mathbf{O}(n)$

5. The feature vectors of the winning neuron and the neurons to be updated are updated.

   $\mathbf{O}(n)$

Hence, time complexity does not depend on t. Space complexity is also only $\mathbf{O}(n)$.
To see how the algorithm converges, we kept on learning up to 50,000 inputs from the same distribution as in Fig. 5, decreasing the learning parameter (α) linearly from 0.1 to 0.001 after the 10,000-th network-input. Fig. 7 shows the frequency distribution $\left(1 \,/\, (y_{i+1} - y_i)\right)$ calculated using the neurons' final position y_i, plotted as "+", compared to that obtained directly from the input data by Equation 5, plotted as a solid line. The horizontal axis of Fig. 5 indicates the relative position in the content and the vertical axis indicates observation frequency normalized as divided by their maximal values. The result shows that the neurons converged well to approximate the true frequency.

3.3 Method 2: with multiple neural networks

3.3.1 Overview

3.3.1.1 Estimation of viewing section

For the reasons described at the end of section 3.1.3, in this section, we divide the proposed method into two phases, namely, the phase to estimate a viewing part by obtaining start/end points, and the other phase to estimate the characteristic parts through the estimation of

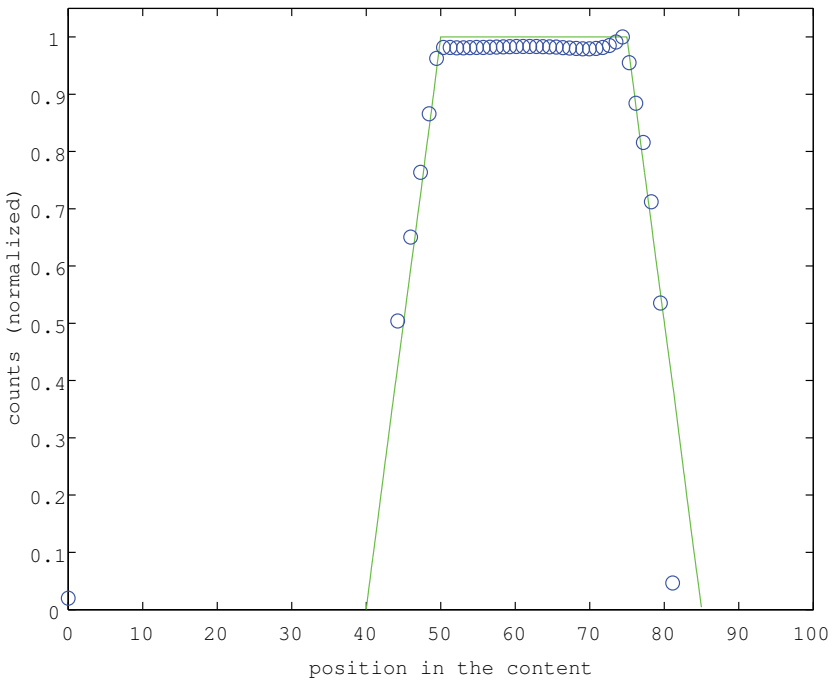


Fig. 7. True frequency distribution (solid line) versus approximated one (circle) for the same input distribution as in section 3.2.2.1.

viewing parts. SOM is utilized in each phase so that we are able to achieve the identification of characteristic parts in a content.
We want to clarify what the neurons will approximate with the SOM algorithm before describing the method proposed in this section. In the SOM or LVQ algorithm, when it has converged, neurons are re-located according to the frequency of network-inputs. Under the problem settings stated in section 3.1, the one-dimensional line is scalar quantitized by each neuron after learning (Van Hulle (2000)).
In other words, the intermediate point of two neurons corresponds to the boundary of the Voronoi cell (Yin & Allinson (1995)). The input frequency on a line segment, L_i, separated by this intermediate point is expected to be $1 \,/\, \|L_i\|^2$. This is because, when LVQ or SOM has converged, the expected value of squared quantitization error E is as follows;

$$E = \int \|x - y_i\|^2 \, p(x) \, dx$$

where $p(x)$ is the probability density function of x and i is the function of x and all $y_\bullet$ (See section 2 for x_i and y_i). This density can be derived as follows (Kohonen (2001));

$$\nabla_{y_i} E \quad = \quad -2 \int (x - y_i) \, p(x) \, dx.$$

It is shown, intuitively, by

$$(y^2 - 2xy)\Big|_{y_i}^{\frac{y_{i+1}+y_i}{2}} \approx (y^2 - 2xy)\Big|_{\frac{y_i+y_{i-1}}{2}}^{y_i}$$
$$y^2\Big|_{y_i}^{\frac{y_{i+1}+y_i}{2}} \approx y^2\Big|_{\frac{y_i+y_{i-1}}{2}}^{y_i}.$$

Refer to section 2 about x and y_i. Therefore, the input frequency z_i on the line segment L_i is expressed as follows;

$$\begin{aligned} z_i &\approx \|L_i\| \,/\, \|L_i\|^2 \\ &= \left(\frac{y_{i+1} - y_i}{2} + \frac{y_i - y_{i-1}}{2}\right)^{-1} \\ &= \frac{2}{y_{i+1} - y_{i-1}}, \end{aligned} \tag{6}$$

where $y_0 = 0$ and $y_{n+1} = M$. Moreover, the piecewise linear function in the two-dimensional plane that connects

$$\left(y_i, \sum_{j=1}^{i} z_j \Big/ \sum_j z_j\right) \tag{7}$$

approximates the cumulative distribution function of $x(t)$. Section 3.3.2.1 experimentally confirms this point (also refer to Fig. 7 in section 3.2.3).

Fig. 8 shows the procedure of the proposed method overall. In this section, we explain from line 1 through 32 only that shows how to estimate frequency of viewing parts by using the above mentioned function of SOM (From line 33 through 43 will be explained in the next section).

We prepared the network (a set of neurons) $\mathcal{S}$ that learns the start point of viewing and the network $\mathcal{E}$ that learns the end point of viewing. Here, the number of neurons of each SOM is set at the same number for simplicity. The learned result of the networks $\mathcal{S}$ and $\mathcal{E}$ will be used for approximating the viewing frequency by network $\mathcal{F}$ (the next section goes into detail) .

According to the network input type, R_{start} or R_{stop} in Equation 2 (lines 5–10), either network $\mathcal{S}$ or $\mathcal{E}$ is selected in order to have it learn the winning as well as neighborhood neurons by applying the original SOM updating formula of Equation 1 (lines 11–18). As stated above, the frequency of inputs in each Voronoi cell, whose boundary is based on the position of each neuron after the learning, is obtained in both networks $\mathcal{S}$ and $\mathcal{E}$.

Based on the above considerations, we propose the following process. When the input $x(t)$ is the start point of viewing (line 19) [1] , the frequency z_i is calculated as below, based on Equation 6.

$$z_i = \begin{cases} \frac{2}{y_{i+1} - y_{i-1}}, & \text{if } y_i \in \mathcal{S} \\ \frac{-2}{y_{i+1} - y_{i-1}}, & \text{if } y_i \in \mathcal{E} \end{cases} \tag{8}$$

[1] When the input $x(t)$ is the end point of viewing, we could reverse the process of Fig. 9 to calculate the cumulative sum until the sum becomes 0 or negative, coming back, in the relative position in a content, from the end point of viewing. And it is expected that applying this process could double the learning speed. However, this paper focuses on only the start point of viewing for simplicity.

neuron $y \leftarrow$ initial value, $\forall y \in$ SOM $\mathcal{S}, \mathcal{E}, \mathcal{F}$
$t \leftarrow 0$
repeat forever
  $t \leftarrow t + 1$
  receive operation data $R(t) = \langle x(t), op(t) \rangle$
  if $op(t) = 1$ **then**
    let $\{y_i \mid 1 \leq i \leq |\mathcal{S}|, y_i \leq y_{i+1}\} \leftarrow \mathcal{S}$ and $i_{max} \leftarrow |\mathcal{S}|$
  else
    let $\{y_i \mid 1 \leq i \leq |\mathcal{E}|, y_i \leq y_{i+1}\} \leftarrow \mathcal{E}$ and $i_{max} \leftarrow |\mathcal{E}|$
  end if
  $\hat{\imath} \leftarrow \text{argmin}_i \left(\|x(t) - y_i\| \right)$
  $y_{\hat{\imath}} \leftarrow y_{\hat{\imath}} + \alpha \left(x(t) - y_{\hat{\imath}} \right)$
  if $\hat{\imath} > 1$ **then**
    $y_{\hat{\imath}-1} \leftarrow y_{\hat{\imath}-1} + \alpha \left(x(t) - y_{\hat{\imath}-1} \right)$
  end if
  if $\hat{\imath} < i_{max}$ **then**
    $y_{\hat{\imath}+1} \leftarrow y_{\hat{\imath}+1} + \alpha \left(x(t) - y_{\hat{\imath}+1} \right)$
  end if
  if $op(t) = 1$ **then**
    ordered set $S' \leftarrow \{i \mid s_i \in \mathcal{S}, x(t) < s_i\}$
    ordered set $E' \leftarrow \{j \mid e_j \in \mathcal{E}, x(t) < e_j\}$
    $Z \leftarrow s_{\min S'}$
    remove $\min S'$ from S'
    while $Z > 0$ **do**
      $w \leftarrow \text{argmin}(s_{\min S'}, e_{\min E'})$
      remove w from S' or E'
      if w comes from S' **then**
        $z \leftarrow \frac{2}{s_{w+1} - s_{w-1}}$
      else
        $z \leftarrow \frac{-2}{e_{w+1} - e_{w-1}}$
      end if
      $Z \leftarrow Z + z$
    end while
    if $w > 0$ **then**
      $v \leftarrow x(t) + rand \times (e_w - x(t))$
      $\hat{\imath} \leftarrow \text{argmin}_i \left(\|v - f_i\| \right),\ f_i \in \mathcal{F}, 1 \leq i \leq |\mathcal{F}|$
      $f_{\hat{\imath}} = f_{\hat{\imath}} + \alpha \left(v - f_{\hat{\imath}} \right)$
      if $\hat{\imath} > 1$ **then**
        $f_{\hat{\imath}-1} \leftarrow f_{\hat{\imath}-1} + \alpha \left(v - f_{\hat{\imath}-1} \right)$
      end if
      if $\hat{\imath} < |\mathcal{F}|$ **then**
        $f_{\hat{\imath}+1} \leftarrow f_{\hat{\imath}+1} + \alpha \left(v - f_{\hat{\imath}+1} \right)$
      end if
    end if
  end if
end repeat

Fig. 8. Procedure of the method proposed in section 3.3.

Here, the cumulative sum Z of frequency z_i is calculated for the neurons $y_i (y_i \geq y_{\hat{i}}$, $y_i \in \mathcal{S}, \mathcal{E})$ which are right to the winning neuron $y_{\hat{i}}$ (lines 24–33). The neuron $e_w(t)(\in \mathcal{E})$ is identified at which Z becomes 0 or negative first from the right of the winning neuron. Thus, the cumulative frequency of the beginning/ending operations of viewing is accumulated from the winner neuron, as starting point, through the point of $e_w(t)$ at which the cumulative frequency becomes 0 or negative value for the first time, after decreasing (or increasing temporally) from a positive value through 0, that means the cumulative sum of positive values is equivalent to, or become smaller than, the cumulative sum of the negative values. Namely, $e_w(t)$ corresponds to obtaining the opposite side of the peak of the probability density function (refer to Fig. 9).
Hence, the interval is obtained, as mentioned above, that is between observed $x(t)$, as the beginning point of viewing, and the point $e_w(t)$ on the opposite side of the peak on the smoothed line of histogram. We could fairly say that this interval $[x(t), e_w(t)]$ is the most probable viewed part when the viewing starts at $x(t)$.

3.3.1.2 Adaptive estimation of partial content position

Based on the estimated most probable viewing parts obtained as the result of the process described in the previous section, the following process concentrates neurons to a content part viewed frequently. The third SOM $\mathcal{F}$ is prepared to learn the frequency based on the estimation of the above mentioned most probable viewing parts, namely, the targeted content parts.
From the most probable viewing parts, $[x(t), e_w(t)]$, estimated by applying the process of Fig. 9, a point v is selected randomly with uniform probability and presented to network $\mathcal{F}$ as the network-input (lines 35–36). Then network $\mathcal{F}$ learns by applying the original SOM procedure (lines 37–43). In other words, the winning and neighborhood neurons are determined for point v that was selected above, and then the feature vectors are updated based on Equation 1.
As the result of this learning, the neurons of network $\mathcal{F}$ are densely located in parts viewed frequently. In short, in the process of the method proposed above, the estimated most probable viewing part ($[x(t), e_w(t)]$ above) is presented to SOM network $\mathcal{F}$ as network-input in order to learn the viewing frequency of the targeted part. The neurons (feature vectors) of the SOM network $\mathcal{F}$ is re-located as their density reflects the content viewing frequency. Based on this property, we can extract the predetermined number (the number of the network $\mathcal{F}$ neurons) of frequently viewed content parts by choosing the parts corresponding to the point the neurons are located. The above mentioned process is incremental; therefore, the processing speed will not be affected significantly by the amount of stored viewing history data.
The proposed method can serve, for example, to obtain locations in the content to extract still images as thumbnails. We can extract not only still images at the position of neurons, but also content parts having specified length of time, the center of which is the positions of neurons, if we specify the time length within the content.

3.3.2 Experiments

The following sections describe the results of the experiments performed in order to evaluate the algorithm proposed in the previous section. The following conditions were adopted in the experiments performed.

– The number of neurons in either of the network $\mathcal{S}$, $\mathcal{E}$, or $\mathcal{F}$ is 11; the neurons are initially positioned equally spaced between 0 and 100.

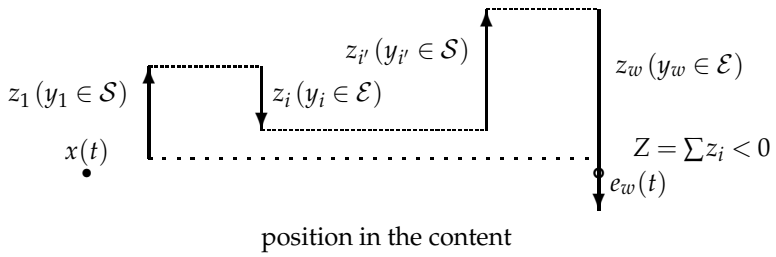


Fig. 9. Visualizing example of cumulative sum of z_i.

- The learning coefficient of each SOM is fixed at 0.01 from the beginning until the end of the experiment.
- Euclidian distance is adopted as the distance to determine the winning neuron.
- A neuron that locates adjacent to the winning neuron (topologically) is the neighborhood neuron (the Euclidian distance between the winning neuron and the neighborhood neuron is not considered).

3.3.2.1 Confirmation of proposed algorithm behavior (single peak)

This section uses a comparatively simple example which is a single-peak histogram in order to describe the proposed algorithm operation in detail. This example is borrowed from the case where the position at approximately 70% in the content is frequently viewed. To simulate such viewing frequency, the network input was given to the network under the following conditions.

- $R = \langle p, 1 \rangle$ (corresponds to starting viewing) is observed at the position p randomly selected in a uniform manner from the entire content with 10% probability
- $R = \langle p, -1 \rangle$ (corresponds to ending viewing) is observed at the position p randomly selected in a uniform manner from the entire content with 10% probability
- $R = \langle p, 1 \rangle$ is observed at the position p randomly selected in a uniform manner from position 55 through 65 in the entire content with 40% probability
- $R = \langle p, -1 \rangle$ is observed at the position p randomly selected in a uniform manner from position 75 through 85 in the entire content with 40% probability

The actual input probability density is shown in lines on the upper pane of Fig. 10. The horizontal axis indicates a position in the content indicated as a percentage of the content from the starting position of the entire content. The vertical axis indicates sum of user operations, where start operation is 1 and end operation is -1. Namely, it plots $C(p)$ in Equation 5 for p as the horizontal axis.

The lower pane of Fig. 10 shows how the neuron positions of the network $\mathcal{F}$ changed as inputs are presented to the network by applying the proposed method until 50,000-th input, where the vertical axis indicates the number of inputs t, and the horizontal axis indicates $y(t)$.

It seems that neurons converged after approximately 30,000 network inputs. The final position of each neuron is plotted with circles overlapped on the upper pane of Fig. 10 (the vertical position is meaningless). The plateau in Fig. 10 corresponds to high frequency of viewing; neurons are located centering on this portion.

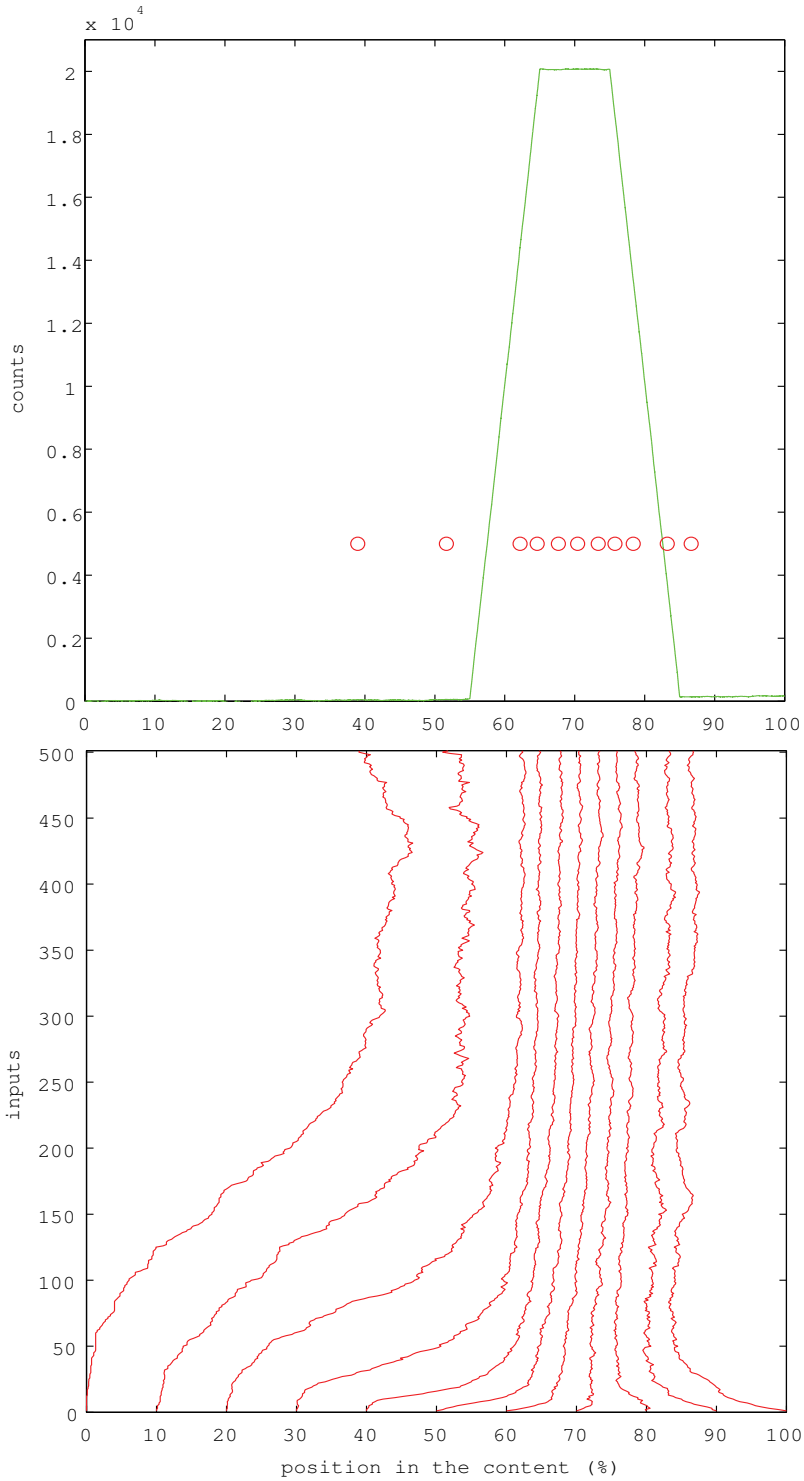


Fig. 10. Result of experiment in section 3.3.2.1.

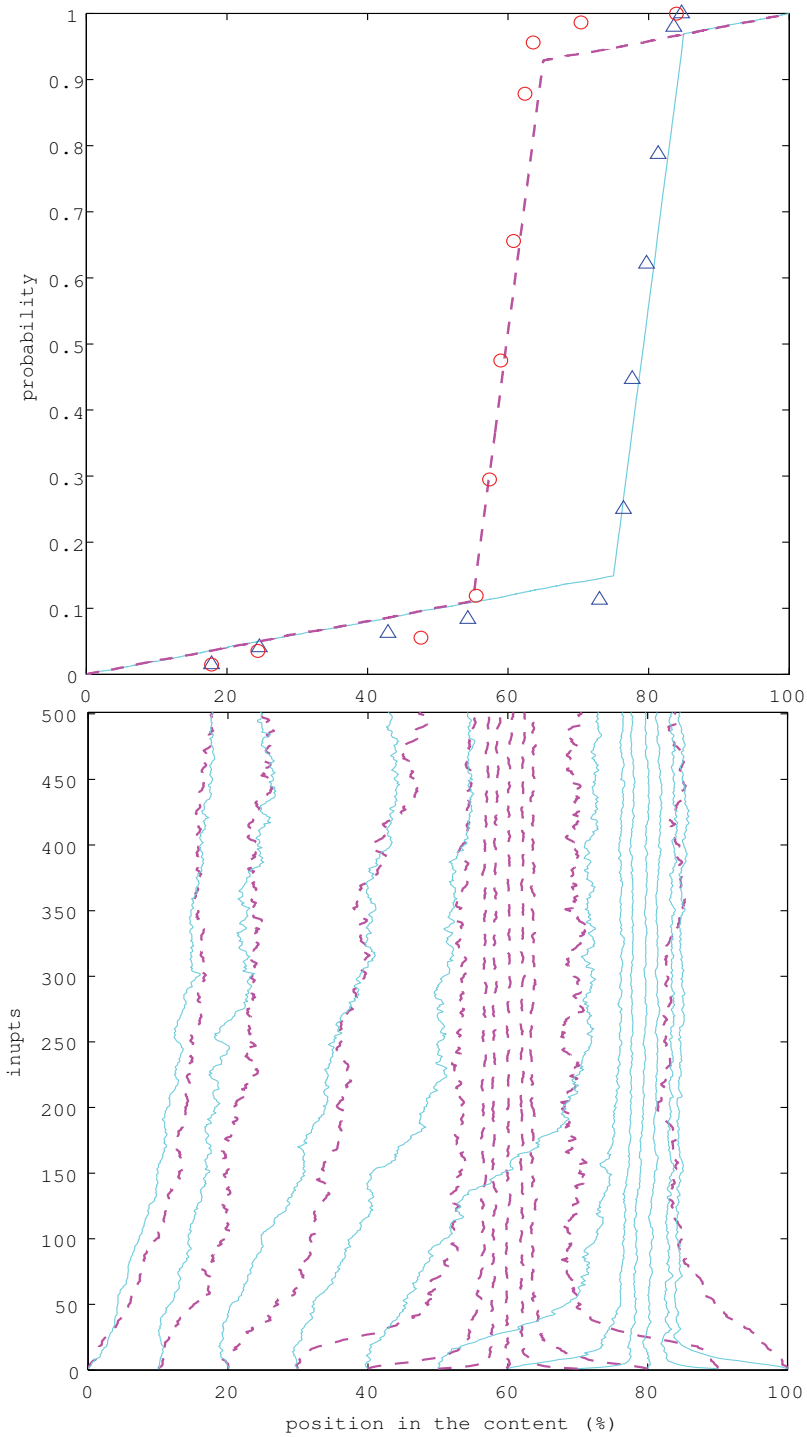


Fig. 11. The final positions of neurons and cumulative frequency of inputs (the upper pane) and the change of positions of neurons during the learning (the lower pane) of SOM $\mathcal{S}$ and $\mathcal{E}$.

The lower pane of Fig. 11 shows the change of positions of neurons during the learning of networks $\mathcal{S}$ and $\mathcal{E}$. The horizontal axis indicates a relative position in the content, while the vertical axis indicates the number of inputs. The dashed line indicates each neuron of network $\mathcal{S}$. It is likely that the neurons converged at a very early stage to the part with high frequency of input.
Here, as described in section 3.3.1.1, each neuron corresponds to scalar quantized bin of the viewing frequency. Accumulated number of neurons from the left approximates the cumulative distribution function.
The upper pane of Fig. 11 shows the cumulative frequency of the network inputs that designates the start points of viewing (Equation 3), in dashed line, and the cumulative frequency of the network inputs that designates the end points of viewing (Equation 4), in solid line (each of them is normalized to be treated as probability).
Moreover, in both of $\mathcal{S}$ and $\mathcal{E}$, the neurons (feature vectors) after learning were plotted with circles by Equation 7 in the upper pane of Fig. 11. Each point approximates the cumulative distribution function with adequate accuracy.

3.3.2.2 Confirmation of proposed algorithm behavior (double peaks)

Fig. 12 shows result of the proposed method applied to a more complicated case when the input frequency distribution has double peaks. The axes are the same as Fig. 10. In this experiment too, neurons converged after around 40,000 network-inputs. We see that neurons are located not around the valley, but around the peaks.

3.3.2.3 Application example of proposed method

The section shows a subject experiment that used an actual viewing history of multimedia content. We had 14 university students as the experimental subjects and used data that was used in Ishikawa et al. (2007).
The content used in the experiment was a documentary of athletes *Mao Asada One Thousand Days* (n.d.), with a length of approximately 5.5 min. We gave the experimental subjects an assignment to identify a part (about 1 sec.), in which a skater falls to the ground only one time in the content, within the content and allowed them to skip any part while viewing the content. Other than the part to be identified (the solution part), the content has the parts related to the solution (the possible solution part), e.g., skating scenes, and the parts not related to the solution, e.g., interview scenes. The percentage of the latter two parts in the whole content length was almost 50%.
The total operations of the experimental subjects were approximately 500, which was overwhelmingly small in number when compared to the number of inputs necessary for learning by the proposed method. For this reason, we prepared the data set whose viewing frequency of each part is squared value as that of obtained from the experiment originally. To circumvent the problem we presented data repeatedly to the network in a random manner until the total of network inputs reached 20,000.
Table 1 shows the experimental results. The far right column shows the distance to the solution part in the number of frames (1 sec. corresponds to 30 frames). We see that neuron 3 came close within approximately 1 sec. of the solution part. Neurons 1 through 3 gathered close to the solution part.
Furthermore, the column named goodness is filled in with circles when each neuron was on the possible solution part; otherwise, with the shortest distance to the possible solution part. Although the number of the circles did not change before and after learning, the average

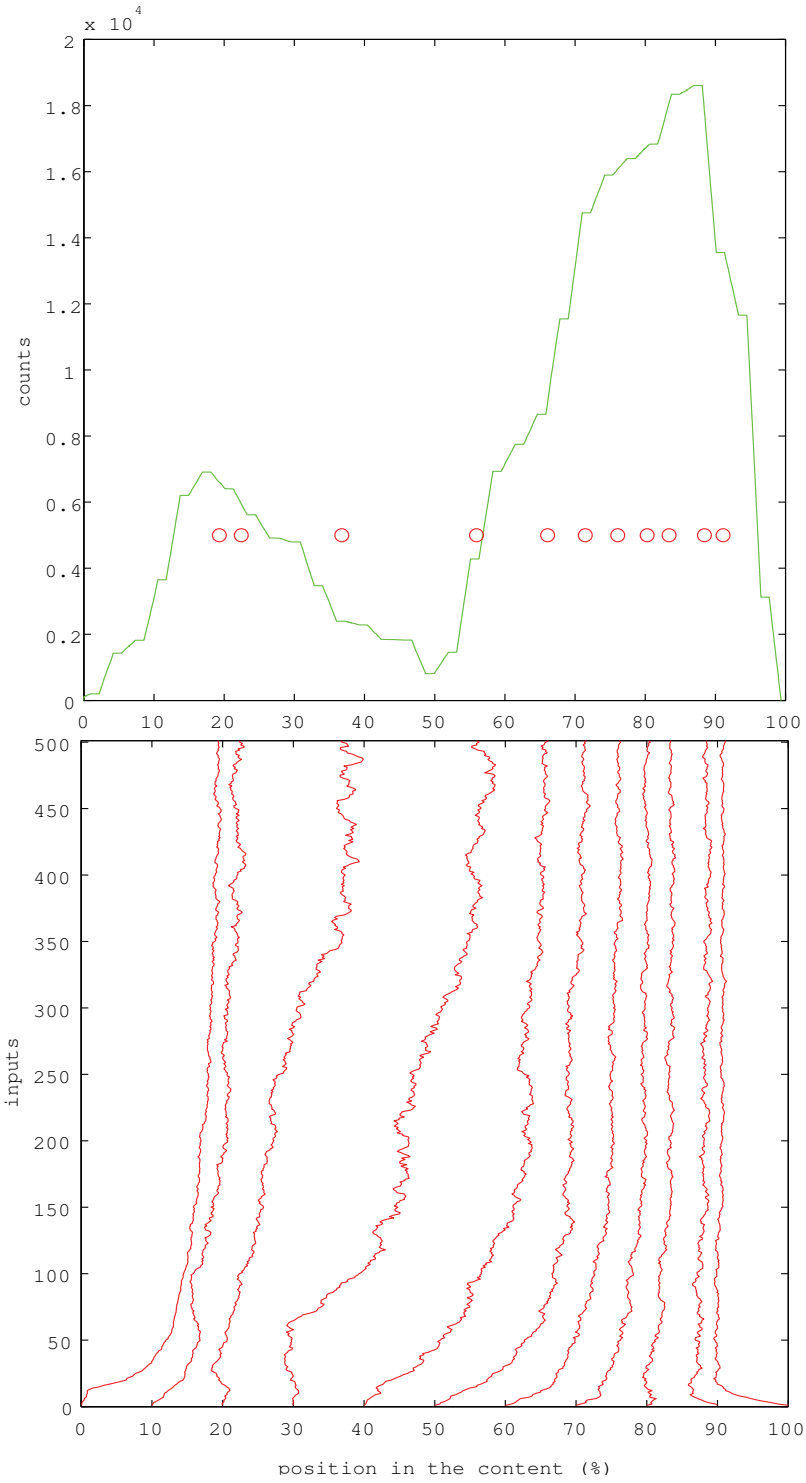


Fig. 12. Result of experiment in section 3.3.2.2.

neuron number	initial positions	converged positions		residuals
	goodness	positions (%)	goodness	(absolute values)
1	248	29.98	433	507
2	130	31.95	231	299
3	10	34.45	◯	35
4	435	41.70	166	704
5	346	50.76	337	—
6	413	56.91	◯	—
7	◯	66.22	13	—
8	◯	70.79	◯	—
9	◯	75.02	◯	—
10	◯	83.95	218	—
11	296	85.85	17	—
average	268.3		202.1	

Table 1. Result of experiment in section 3.3.2.3.

distance to the possible solution part was shortened, and neurons moved to the appropriate parts. Regarding neuron 1, the reason the distance to the possible solution part increased may be that neurons are concentrated on the solution part and it was pushed out. If so, when an adequate amount of data is given, neurons are expected to be located on the solution and possible solution parts.
Fig. 13 shows an example of extracted still images from the part that corresponds to each neuron position.
The proposed method gave a result, i.e., neuron or still image 3, close to the solution as is seen in Table 1 but is not a solution as is seen in Fig. 13. The reason is as follows; (1) as reported in Ishikawa et al. (2007), the existence of experimental subjects that preferred to do their own search and ignored the histogram intentionally could lead to lowering accuracy, (2) the total length of the content time could vary a few seconds because the content was distributed via networks (this was already confirmed under the experimental environment), (3) user operations were mainly done by using the content player's slider; there were fluctuations too significant to realize the accuracy on the second time scale level, and (4) as stated above, the amount of data for learning was not necessarily adequate causing the estimation to be significantly varied. We consider these things to be unavoidable errors.
However, the part identified by using the proposed method is about one frame before the part the skater started her jump. The time point to which this skater's jump lands on is the actual solution part, and this identified part is just 35 frames away from the solution part. Therefore, although the solution part is not extracted as a still image by proposed method, if we extract a content parts starting 2 seconds before the neuron positions and ending 2 second after of them, the solution part is included. Under such condition, the content was summarized to 44 seconds (which is total viewing times, $(2+2)$ sec. $\times 11$ neurons, for a set of partial content parts) from 5.5 minutes; in addition, the solution part was also included. For this reason, we claim that this proposed method is effective from the standpoint of shortening the viewing time.

3.3.3 Discussion

The method proposed in section 3.3 identifies a part that is important in the content automatically and adaptively, based on the accumulated viewing history. As explained

Fig. 13. Still images extracted at neurons' positions.

for example in Ishikawa et al. (2007), mainly in video content, multiple still images (also referred to as thumbnails) captured from a video file could be utilized as a summary of the video. Usually, capture tasks are conducted automatically at a constant time interval, or done manually by users while viewing the video content. The method proposed in this section can also be applied to capturing still images, and it is expected that this proposed method can make it possible to capture those still images in the parts with high frequency of viewing with less task burden.

As introduced in section 2, neurons are located according to the frequency of network-inputs by using the SOM method. However, it is impossible for conventional SOM to learn such data having a range. For this reason, we have formulated a new method that takes advantage of self-organizing features.

The method proposed in section 3.3.1 is a general method to extend SOM to approach the issue where the data to be learned has a range and only the starting or ending points of the range are given as input. We think this method will extend application range of SOM. The

experiments performed confirmed that, based on the data of start/end of viewing obtained by using the method in Ishikawa et al. (2007). This method has made a success to determine most appropriate time for extracting still images that reflect the viewing frequency of the content adaptively, incrementally, and automatically.
A process that corresponds to building a histogram is required in order to estimate the frequency of intervals such as time-series content part that consists of the starting and ending points. We realized this process by using two SOMs ($\mathcal{S}$ and $\mathcal{E}$ described in section 3.3.1.1) and the process shown in Fig. 9. The space and time complexity of the method using RDB and proposed method is as follows; where n is the number of the histogram bins (i.e., the number of partial content parts to be identified).

1. $R(t) = \langle p, m \rangle$ is inserted into the sorted array A:

   $\mathbf{\Omega}(\log t) + \mathbf{\Omega}(\log t)$

2. The cumulative integral array B of the array A is obtained:

   $\mathbf{\Omega}(\log t)$

3. From $b_i = i \times \dfrac{\big(\max(B) - \min(B)\big)}{n} \in B$, $1 \leq i \leq n$, b_i is calculated in order to extract partial content parts:

   $\mathbf{\Omega}(\log t)$

Space complexity is $\mathbf{\Omega}(t)$ (the SQL process described in section 3.1.2 is partly common to the processes of 1 and 2 above). The SOM of the proposed method ($\mathcal{F}$ in the previous section) that does learning which reflects frequency also makes it possible to conduct part of the processes of 2 and 3 above by combining the above mentioned methods.
On the other hand, the process described in this section (SOM $\mathcal{S}$ and $\mathcal{E}$) that estimates the cumulative distribution function by using SOM corresponds to part of the processes of 1 and 2 above. Use of SOM eliminates the need for sorting or determining the portion where data is inserted. The network conducts learning by inputting data observed as it is. Time complexity for this is as follows, independent from t.

4. $\mathrm{argmin}_i(||p - y_i||)$ in the SOM $\mathcal{S}$ or $\mathcal{E}$, and $R = \langle p, m \rangle$ are obtained:

   $\mathbf{O}(n) + \mathbf{O}(n)$

5. The feature vectors of the winning and neighborhood neurons are updated:

   $\mathbf{O}(1)$

6. S' and E' are obtained, z_i is accumulated, and the point e_w which is $Z < 0$ is obtained:

   $\mathbf{O}(n)$

Space complexity is only $\mathbf{\Omega}(n)$. Moreover, with respect to the part of the process 2 and 3, time complexity is as follows, while space complexity is $\mathbf{O}\big(|\mathcal{F}|\big)$.

proposed method		solutions calculated by	residuals
neuron number	converged position	cumulative distribution	(absolute values)
1	38.99	60.68	21.69
2	51.66	63.13	11.47
3	62.19	64.99	2.80
4	64.63	66.66	2.03
5	67.65	68.34	0.69
6	70.42	70.01	0.41
7	73.35	71.70	1.65
8	75.74	73.41	2.33
9	78.34	75.10	3.24
10	83.21	76.98	6.23
11	86.62	79.46	7.16

Table 2. Result of proposed versus true value.

7. One point within the section $[x(t), e_w(t)]$ is selected, and $\mathrm{argmin}_i \left(\|p - y_i\| \right)$, $R = \langle p, m \rangle$ are obtained:

   $\mathbf{O}(|\mathcal{F}|)$

8. The feature vectors of the winning and neighborhood neurons are updated :

   $\mathbf{O}(1)$

With respect to the experiments described in section 3.3.2.1, we compare the result of the proposed method to approximation of the cumulative distribution function with rectangular integration (corresponding to the processes of 1–3 above in this section). Table 2 shows the result, where the open interval $(0, 1)$ was divided into 12 intervals. As for the neurons 3 through 9, the difference was 5% or less; we can say that the proposed method showed good results. However, a significant difference was observed for the four neurons other than the above. As shown in the upper pane of Fig. 10, the proposed method unfortunately locates neurons so as to cover the parts with comparatively low frequency. On the other hand, all results based on the cumulative distribution were located within the plateau shown on the upper pane of Fig. 10; however, from the viewpoint of balance of summarization, we consider that the neurons, or extracting content parts, should not be concentrated to this extent as these results. The difference become larger gradually neuron 5 to 1 and 6 to 11. Thus, it could be possible that neurons 1, 2, 10, and 11 pulled other neurons in the result of the proposed method.

The studies so far on summarization of multimedia content mainly focused on what information can be used for summarizing content and how to summarize by such information (e.g., Yu et al. (2003); Yamamoto & Nagao (2005)). In this chapter, we adopted comparatively simple method, and examined its effectiveness. We proposed a method using SOM, which has smaller time and space complexity, thus a more scalable process could be realized when compared to the method RDB.

Self-organization typified by SOM can be understood as identifying most characteristic elements based on data given by the environment. When self-organization is applied to

functional approximation or clustering, it identifies features of a limited number representing the entire function or all features.
Such property is good for summarization, because, also in summarization, fewer resources represent the all. Therefore, it is possible to regard the proposed method as a self-organizing approach for time-series digital content summarization. The concept of summarizing content by means of self-organization is proposed by this paper for the first time. The method itself is effective; there still remains the room for research on the relationship between self organization and summarization that may lead to extension of the proposed method.
The conventional software programs (e.g., *Video Browser Area61* (n.d.)) that capture still images from multimedia content seem to have difficulty in realizing functions other than the followings; (1) to capture still images at equal (time) intervals, and (2) to capture still image whose position are determined manually. Function (1) is a mechanical process, so that it is impossible to reflect viewing frequency as realized by the proposed method. If it is applied to the experiments described in section 3.3.2.3, whether the solution part is extracted or not depends on chance completely.
By function (2), the solution part will be extracted properly. However, the following two problems remain: the result might depend on a user charged with this particular task, so that the validity of the results should be ensured, and the burden for the identification task is significant.
For these problems, similar to Ishikawa et al. (2007), we proposed a method utilizing "wisdom of crowds." In other words, we expect to obtain good results by majority voting of viewing behavior of comparatively large number of viewers (not particular viewers).
This viewpoint lead to a solution for the second problem of the above mentioned function (2), too. By calculating the users' viewing behavior, the proposed method can naturally identify proper positions for capture; nobody needs to bear the burden for the capture position identification.
Given the above mentioned facts, we consider that the proposed method has the advantages to both functions (1), where it can identify the proper part, and (2), where it does not increase related task burden.
We proposed, in section 3.2, a SOM-like method that uses only a single system as the solution for the problems considered in this section. This method is simple and also clear in theory when compared to the method proposed in section 3.3, whereas the latter realizes a process with smooth convergence. Moreover, although the proposed method in section 3.3 requires learning of the three networks, only the neurons near to the winner are selected and updated their feature vectors. This fact shows that the proposed method in section 3.3 has advantage in the time complexity compared to the method proposed in section 3.2. We are going to have further discussions and examinations about the detailed comparison of the two methods and extension of the method, proposed in section 3.3, with only one network in the future.

4. Conclusion

We proposed in this chapter two kinds of SOM-like algorithm that accepts online input as the start and end of viewing of a multimedia content by many users; a one-dimensional map is then self-organized, providing an approximation of the density distribution showing how many users see a part of a multimedia content. In this way "viewing behavior of crowds" information is accumulated as experience accumulates, summarized into one SOM-like network as knowledge is extracted, and is presented to new users.

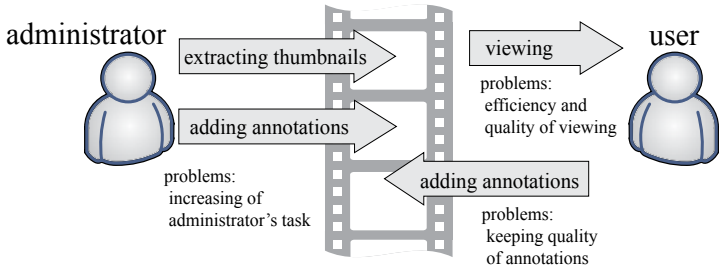


Fig. 14. Problems solved by the method proposed in Ishikawa et al. (2007).

SOM proposed by Kohonen can approximate frequency distribution of events and consequently can be used as a tool to form a "wisdom of crowds" or "collective intelligence" which is one of the important concepts in Web 2.0. But from its nature SOM cannot utilize interval data. However, the proposed algorithms, though limited to a one-dimensional case, approximate density of intervals from their starting and ending events.
Ishikawa et al. (2007) proposed, using RDB, a mechanism of providing only important parts within multimedia content by storing the viewing frequency of many viewers to those users who have just started viewing the content. This method realizes summarization which has been regarded as difficult without causing significant burden on creators of the content (e.g., He et al. (1999)), users, or computers.
In this chapter, we made an attempt to extend this method to promote additional automation. We tried to reduce the computational complexity in order to realize excellent scalability. We took up the concern of adaptively determining a proper time for capturing still images as the application case of the proposed method, and we described how to realize it.
We realized incremental as well as adaptive processes by means of learning, derived from the SOM algorithm. The process of SOM is originally an algorithm that accepts single points given as input; on the other hand, the proposed method extends it to process range data as input.
In terms of the process and concept, the method proposed by this chapter has a close relationship with what is called self-organization. We understand this proposed method as an approach to self-organize time-series digital content. We regard this proposed method as a promising approach, because it is expected that the effectiveness of this proposed method will be greatly improved by deepening our consideration of this relationship.

5. Acknowledgement

The research activity of first author is partly supported by Grant-in-Aid for Scientific Research (C), 20500842, MEXT, Japanese government.

6. References

Bak, P., Tang, C. & Wiesenfeld, K. (1988). Self-organized criticality, *Physical Review A* 38(1): 364–374.

Bonabeau, E., Theraulaz, G. & Dorigo, M. (eds) (1991). *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press.

Buchanan, M. (2007). *The Social Atom: Why the Rich Get Richer, Cheaters Get Caught, and Your Neighbor Usually Looks Like You*, Bloomsbury.

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G. & E., B. (eds) (2003). *Self-Organization in Biological Systems*, Princeton University Press.

Google (n.d.). `www.google.com`.

Haken, H. (1996). *Principles of Brain Functioning*, Springer.

He, L., Sanocki, E., Gupta, A. & Grudin, J. (1999). Auto-summarization of audio-video presentations, *Proc. of the 7th ACM Intl. Conf. on Multimedia*, pp. 489–498.

Ishikawa, K., Oka, M., Sakurai, A. & Kunifuji, S. (2007). Effective sharing of digital multimedia content viewing history, *Journal of ITE* 61(6): 860–867. in Japanese.

Johnson, S. (2002). *Emergence: The Connected Lives of Ants, Brains, Cities, and Software*, Scribner.

Kauffman, S. (1995). *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity*, Oxford University Press.

Kohonen, T. (2001). *Self-organizing Maps*, 3rd edn, Springer-Verlag.

Krugman, P. R. (1996). *The Self-Organizing Economy*, Blackwell.

Mao Asada One Thousand Days (n.d.).
`www.japanskates.com/Videos/MaoAsadaOneThousandDayspt1.wmv`.

Nonaka, I. & Takeuchi, H. (1995). *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*, Oxford University Press.

Ohmukai, I. (2006). Current status and challenges of Web2.0: Collective intelligence on Web2.0, *IPSJ Magazine* 47(11): 1214–1221. in Japanese.

O'Reilly, T. (2005). What is web 2.0: Design patterns and business models for the next generation of software. `oreilly.com/web2/archive/what-is-web-20.html`.

Page, S. E. (2007). *The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies*, Princeton University Press.

Surowiecki, J. (2004). *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*, Doubleday.

Tapscott, D. & Williams, A. D. (2006). *Wikinomics: How Mass Collaboration Changes Everything*, Portfolio.

Van Hulle, M. M. (2000). *Faithful Representations and Topographic Maps: From distortion- to information-based self-organization*, John Wiley & Sons.

Video Browser Area61 (n.d.).
`www.vector.co.jp/vpack/browse/pickup/pw5/pw005468.html`.

Yamamoto, D. & Nagao, K. (2005). Web-based video annotation and its applications, *Journal of JSAI* 20(1): 67–75. in Japanese.

Yin, H. & Allinson, N. M. (1995). On the distribution and convergence of feature space in self-organizing maps, *Neural Computation* 7(6): 1178–1187.

YouTube (n.d.). `www.youtube.com`.

Yu, B., Ma, W., Nahrstedt, K. & Zhang, H. (2003). Video summarization based on user log enhanced link analysis, *Proc. of the 11th ACM Intl. Conf. on Multimedia*, pp. 382–391.

Zhabotinsky, A. M. (1991). A history of chemical oscillations and waves, *Chaos* 1(4): 379–386.

10

Image Search in a Visual Concept Feature Space with SOM-Based Clustering and Modified Inverted Indexing

Md Mahmudur Rahman
Concordia University
Canada

1. Introduction

The exponential growth of image data has created a compelling need for innovative tools for managing, retrieving, and visualizing images from large collection. The low storage cost of computer hardware, availability of digital devices, high bandwidth communication facilities and rapid growth of imaging in the World Wide Web has made all these possible. Many applications such as digital libraries, image search engines, medical decision support systems require effective and efficient image retrieval techniques to access the images based on their contents, commonly known as content-based image retrieval (CBIR). CBIR computes relevance of query and database images based on the visual similarity of low-level features (e.g., color, texture, shape, edge, etc.) derived entirely from the images Smeulders et al. (2000); Liua et al. (2007); Datta et al. (2008). Even after almost two decades of intensive research, the CBIR systems still lag behind the best text-based search engines of today, such as Google and Yahoo. The main problem here is the extent of mismatch between user's requirements as high-level concepts and the low-level representation of images; this is the well known *"semantic gap"* problem Smeulders et al. (2000).

In an effort to minimize the *"semantic gap"*, some recent approaches have used machine learning on locally computed image features in a *"bag of concepts"* based image representation scheme by treating them as visual concepts Liua et al. (2007). The models are applied to images by using a visual analogue of a word (e.g., *"bag of words"*) in text documents by automatically extracting different predominant color or texture patches or semantic patches, such as, water, sand, sky, cloud, etc. in natural photographic images. This intermediary semantic level representation is introduced as a first step to deal with the *semantic gap* between low-level features and high-level concepts. Recent works have shown that local features represented by *"bags-of-words"* are suitable for scene classification showing impressive levels of performance Zhu et al. (2002); Lim (2002); Jing et al. (2004); Vogel & Schiele (2007); Shi et al. (2004); Rahman et al. (2009a). For example, a framework to generate automatically the visual terms ("keyblock") is proposed in Zhu et al. (2002) by applying a vector quantization or clustering technique. It represents images similar to the *"bags-of-words"* based representation in a correlation-enhanced feature space. For the reliable identification of image elements, the work in Lim (2002) manually identifies the visual patches ("visual keywords") from the sample images. n Jing et al. (2004), a compact and sparse representation of images is proposed based on the utilization of a region codebook generated by a clustering technique. A semantic

modeling approach is investigated in Vogel & Schiele (2007) for a small collection of images based on the binary classification of semantic patches of local image regions. A medical image retrieval framework is presented in Rahman et al. (2009b) that uses a visual concept-based feature space for which statistical models are built using a probabilistic multi-class support vector machine (SVM). The images are represented using concepts that comprise of color and texture patches from local image regions in a multi-dimensional feature space. It is demonstrated by experimental evaluation that approaches using intermediate semantic concepts are more appropriate to deal with the gap between low and high-level Boschet et al. (2007).

There exists a strong similarity between the keyword-based representation of documents in the vector space model (VSM) Yates & Neto (1999) and the majority of the concept-based image representation schemes as mentioned above. Besides the loss of all ordering structure, each concept is considered independent of all the other concepts in this model. However, this independent assumption might not hold in many cases as in general there exists correlated or co-occurring concepts in individual images as well as in a collection. For example, there is a higher probability of occurrence of a blue sky around the sun in an outdoor image. Whereas, a flower of yellow color has more probability to co-occur with green leaves in an image of a flower garden. In these examples, individual objects, such as sky, sun, flower, and leaf can be considered as visual concepts with their distinct color and texture patterns. Hence, there is indeed a need to exploit the correlation or co-occurrence patterns among the concepts to improve the effectiveness of the retrieval process. To overcome the limitation, we present a correlation-enhanced similarity matching and query expansion framework on the concept-based feature space. We explore the similarity matching technique based on the global correlation analysis of the concepts and query expansion based on a local neighborhood analysis of a SOM generated codebook by exploiting the topology preserving structure. The codebook or topology preserving SOM map is utilized to represent images as sparse feature vectors and an inverted index is created on top of this to facilitate efficient retrieval. In this approach, a global similarity/correlation matrix or thesaurus is generated off-line and utilized in a quadratic form of distance measure to compare the query and database images. However, due to the quadratic nature, the distance measure is computationally intensive. To overcome this, only a subset of the images of the entire collection is compared based on a local neighborhood analysis in an inverted index built on top of the codebook to reduce the search time and at the same time to ensure the retrieval effectiveness.

The organization of this chapter is as follows: In Section 2, the visual concept-based image representation approach is discussed. Section 3 presents the correlation-enhanced similarity matching approach based on the generation of several global matrices. In Section 4, we present the similarity matching approach in a modified inverted index. Exhaustive experiments and analysis of the results are presented in Sections 5 and 6. Finally, Section 7 provides our conclusions.

2. Visual concept-based image representation

By the term *"visual concept"*, we refer to the perceptually distinguishable color and/or texture patches that are identified locally in image regions. For example, in a heterogeneous collection of medical images, it is possible to identify specific local patches, such as homogeneous texture patterns in grey level radiological images, differential color and texture structures in microscopic pathology and dermoscopic images. The variation in these local patches can be effectively modeled by using unsupervised clustering or supervised classification techniques

Fukunaga (1990). There are three main steps to be considered before representing images in a visual concept-based feature space: the generation of a set of visual concepts from the local image regions; the construction of a codebook of prototype concepts analogous to a dictionary of keywords; and the encoding of the images with the concept indices of the codebook Rahman et al. (2009a).

Definition 1 *A codebook* $C = \{c_1, \cdots, c_j, \cdots, c_N\}$ *is a set of prototype visual concepts where each* c_j *is associated with a label* j *and a vector* $\mathbf{c}_j = [c_{j_1} \ \cdots c_{j_2} \cdots c_{j_d}]^{\mathrm{T}}$ *of dimension* d *in a combined color and texture-based feature space.*

To generate the codebook, a reasonable training set of images needs to be selected either manually or in a random manner. Let $\mathcal{D}$ be an image database and let a subset $\hat{\mathcal{D}} = \{I_1, \cdots, I_j, \cdots, I_m\} \subset \mathcal{D}$ form a training set of images. After forming the training set, the next step is to segment the training images into regions and extract the low-level image features from each region as a representative of the initial visual concept vectors. Since, the automatic segmentation schemes usually offer only an unreliable object description, we use a fixed partitioning scheme. Let an image $I_j \in \hat{\mathcal{D}}$ be partitioned into an $r \times r$ grid of l blocks as segmented regions to generate the region vectors $\{\mathbf{x}_{1_j}, \cdots, \mathbf{x}_{k_j}, \cdots, \mathbf{x}_{l_j}\}$ where each $\mathbf{x}_{k_j} \in \Re^d$ is a vector in a low-level feature space.

To represent each region as a feature vector $\mathbf{x}_i$, the mean and the standard deviation of each channel in the HSV (Hue, Saturation, and Value) color space as a 6-dimensional color feature vector and the second order moments (such as, energy, maximum probability, entropy, contrast, and inverse difference moment) as a 5-dimensional texture feature vector are extracted from a grey level co-occurrence matrix (GLCM) Haralick et al. (1973).

Finally the color and texture vectors are combined as a single region vector after re-scaling the feature attributes with zero mean and unit variance. There are in total m training images. So, the partition scheme will generate $n = (l \times m)$ region vectors for all the training images and collectively we can refer to them as a set of vectors $\mathbf{X} = \{\mathbf{x}_1, \cdots, , \mathbf{x}_i, \cdots \mathbf{x}_n\}$ where each $\mathbf{x}_i$ is a vector of dimension d. Since, the features from the blocks rather than individual pixels are used as vectors, some information on the spatial relationship among the neighboring pixels in the images are already retained. In general, there might be several similar regions in terms of the image features in an individual image as well as in different images in the same training set. Since our visual system should tolerate some small errors, if the difference between two regions is below a certain preset threshold, they are deemed to be the same. Hence, a subset of these representative vectors needs to be selected as a codebook of the visual concept prototype by applying a clustering algorithm, such as SOM Kohonen (1997).

2.1 Codebook generation by SOM

To generate a coodbook of prototype vectors (e.g., concept vectors) from the above features, we utilize the SOM-based clustering Kohonen (1997). The SOM is basically an unsupervised and competitive learning algorithm, which finds the optimal set of prototypes based on a grid of artificial neurons whose weights are adapted to match input vectors in a training set Kohonen (1997). It has been successfully utilized for indexing and browsing by projecting the low-level input features to the two-dimensional grid of the SOM map Laaksonen et al. (2002); Vesanto (2002); Yen & Zheng (2008). However, in this work it is utilized to generate a codebook of visual concepts based on a two-dimensional SOM map.

The basic structure of a SOM consists of two layers: an input layer and a competitive output layer as shown in Figure 1. The input layer consists of a set of input node vectors. The

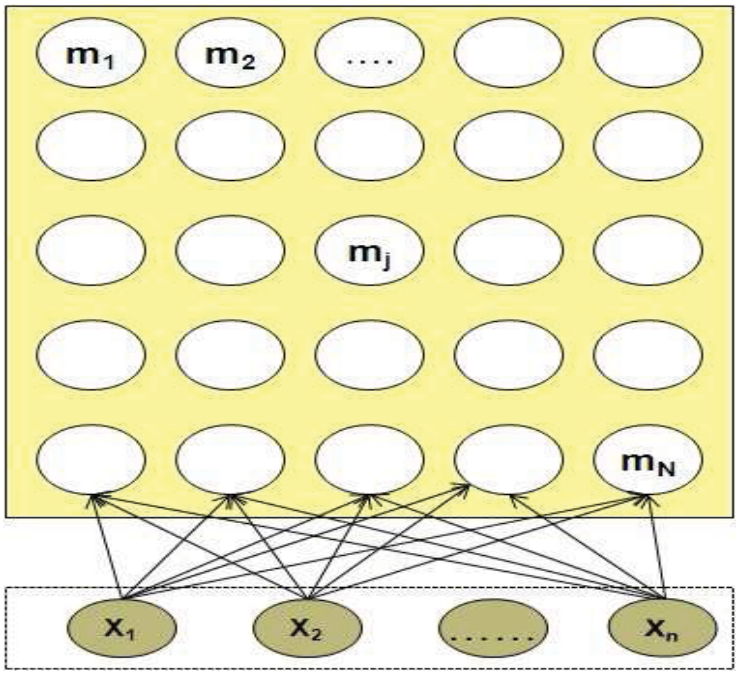


Fig. 1. Structure of the SOM

output map consists of a set of N units organized into either a one-or two-dimensional lattice structure where each unit m_j is associated with a weight vector $\mathbf{w}_j \in \Re^d$. During the training phase, the set of the input vectors is presented to the map a multiple number of times and the weight vectors stored in the map units are modified to match the distribution and topological ordering of the feature vector space. The first step of the learning process is to initialize the weight vectors of the output map. Then, for each input vector $\mathbf{x}_i \in \Re^d$, the distances between the $\mathbf{x}_i$ and weight vectors of all map units are calculated as

$$\|\mathbf{x}_i - \mathbf{w}_c\|^2 = \min_j \left\|\mathbf{x}_i - \mathbf{w}_j\right\|^2, \text{ for } j = 1,2,\cdots,N \tag{1}$$

where $\|.\|^2$ is a distance measure in the Euclidean norm. The unit that has the smallest distance is called the *best-matching unit* (BMU) or the *winning node*. The next step is to update the weight vectors associated with the BMU, m_c as

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t)\theta_{cj}(t)(\mathbf{x}_i(t) - \mathbf{w}_j(t)) \tag{2}$$

Here, t is the current iteration, $\mathbf{w}_j(t)$ and $\mathbf{x}_i(t)$ are the weight vector and the target input vector respectively at the iteration t, and $\theta(t)$ and $\alpha(t)$ are the smooth neighborhood function and the time-dependent learning rate.

Due to the process of self-organization, the initially chosen $\mathbf{w}_j$ gradually attains new values such that the output space acquires appropriate topological ordering. After the learning phase, the map can be used as a codebook where the map units represent the prototype visual concepts and their associated weight vectors represent the prototype concept vectors. Hence, a weight vector $\mathbf{w}_j$ of unit m_j resembles a visual concept vector $\mathbf{c}_j$ in the codebook C. In

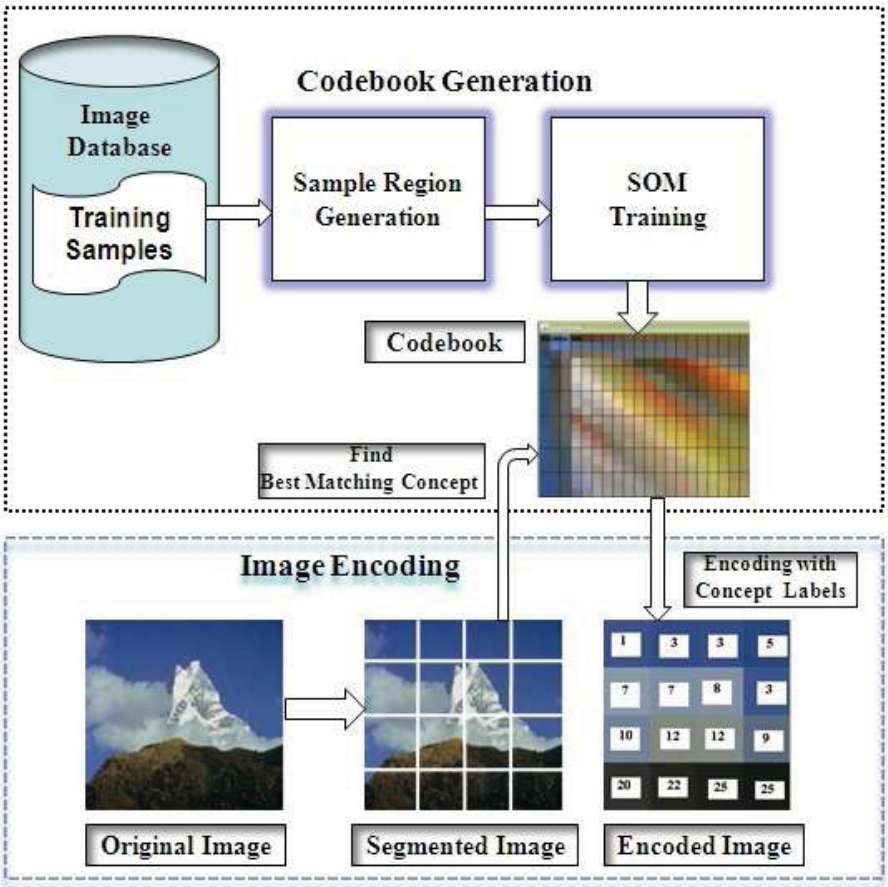


Fig. 2. Codebook generation and Image encoding process

general, the visual concept prototypes in the resulting codebook represent the most general structures extracted from all the training input vectors.

2.2 Image encoding and feature representation

The codebook can be effectively utilized as a simple image compression and representation scheme Zhu et al. (2002). To encode an image with the visual concept prototype labels or indices of the codebook, it is decomposed into an even grid-based $(r \times r)$ partition where similar low-level color and texture features are extracted from each region as is used as the training images. Let an image I_j be partitioned into $l = (r \times r)$ blocks or regions to generate vectors $\{\mathbf{x}_{1_j}, \cdots, \mathbf{x}_{k_j}, \cdots, \mathbf{x}_{l_j}\}$ where each $\mathbf{x}_{k_j} \in \Re^d$. For each vector $\mathbf{x}_{k_j}$ in I_j , the codebook is searched to find the best match concept prototype (e.g., BMU in the map)

$$c_k = \arg\min_{1 \leq l \leq N} \left\| \mathbf{x}_{k_j} - \mathbf{c}_l \right\|^2 \tag{3}$$

where k denotes the label of c_k and $\|.\|^2$ denotes the Euclidean distance between the region vectors of I_j and the concept prototype vectors.

After this encoding process, each image is represented as a two-dimensional grid of concept prototype labels where the image blocks are linked to the corresponding best matching concept prototypes in the codebook. Figure 2 shows schematically the codebook generation and image encoding processes. The codebook generation is performed in the top portion of Figure 2 and the bottom portion shows how an example image is encoded with the indices (e.g., prototype concept labels) of the codebook. Based on this encoding scheme, an image I_j can be represented as a concept vector

$$\mathbf{f}_j = [w_{1j} \cdots w_{ij} \cdots w_{Nj}]^{\mathrm{T}} \tag{4}$$

where each element w_{ij} represents the normalized frequency of occurrences of the visual concept label of c_i appearing in I_j.

3. Correlation-enhanced similarity matching

This section presents the similarity matching approach in the visual concept space by considering the correlations between the concepts in the collection. For the correlation analysis, we construct a global structure or thesauruses in the form of a correlation matrix where each element defines concept co-relationships. Finally, this global matrix is utilized in a Quadratic form of distance measure to compare a query and database images. The quadratic distance measure is first implemented in the QBIC Hafner et al. (1995) system for the color histogram-based matching. It overcomes the shortcomings of the L-norm distance functions by comparing not only the same bins but multiple bins between color histograms. Due to this property, it performs better compared to the Euclidean and histogram intersection-based distance measures for the color-based image retrieval Hafner et al. (1995). However, a similarity based on only the color feature does not always indicate semantic similarities between the images due to the *semantic gap* problem and does not imply any hidden correlation between feature attributes in a collection.
The visual concept-based feature representation is at a higher level then the simple pixel-based color feature representation due to the incorporation of both color and texture features in a region level. Since, the concept prototype vectors in the codebook are already represented in a feature space based on the color and texture features, we can use them directly to generate a *concept-concept* similarity matrix.

Definition 2 *The* concept-concept *similarity matrix* $\mathbf{S}_{N\times N} = [s_{u,v}]$ *is built through the computation of each element* $s_{u,v}$ *as the Euclidean similarity values between two vectors* $\mathbf{c}_u$ *and* $\mathbf{c}_v$ *of concept prototypes* c_u *and* c_v *as*

$$s_{u,v} = sim\,(\mathbf{c}_u, \mathbf{c}_v) = 1 - dis\,(\mathbf{c}_u, \mathbf{c}_v) \tag{5}$$

where $\mathbf{c}_u$ *and* $\mathbf{c}_v$ *are d-dimensional vector in a combined color and texture feature space and* $c_u, c_v \in C$ *where N is the size of the codebook C and dis* $(\mathbf{c}_u, \mathbf{c}_v)$ *denote the Euclidean distance between* $\mathbf{c}_u$ *and* $\mathbf{c}_v$.

Instead of using a matrix based on similarities in a color space, we can effectively utilize this global visual *concept-concept* similarity matrix $\mathbf{S}$ for the distance measure computation as

$$Dis_{\mathrm{S}}(I_q, I_j) = \sqrt{(\mathbf{f}_q - \mathbf{f}_j)^T \mathbf{S} (\mathbf{f}_q - \mathbf{f}_j)} \tag{6}$$

However, the visual similarities between the concepts might not always imply semantic similarities or hidden correlations between the concepts as mentioned earlier. Hence, we

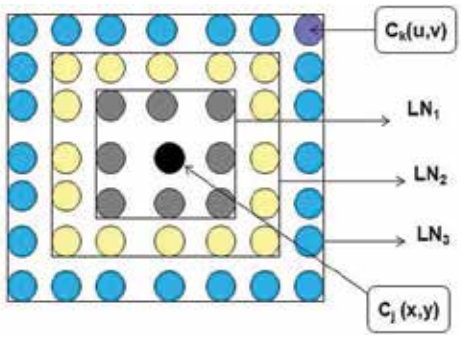


Fig. 3. Topological local neighborhoods

utilize another global matrix that is built by considering the co-occurrence of concepts inside images.

Definition 3 *Let* $\mathbf{A} = [a_{uv}]$ *be a* $N \times N$ *matrix in which the rows and columns are associated with the concepts in the codebook C. Each entry* a_{uv} *expresses a normalized correlation factor between concepts* c_u *and* c_v *as*

$$a_{uv} = n_{uv}/(n_u + n_v - n_{uv}) \tag{7}$$

where n_u *be the number of images in* S_l *that contain the concept* c_u, n_v *be the number of images that contain the concept* c_v, *and* n_{uv} *be the number of the images in the collection that contain both the concepts.*

The entry a_{uv} measures the ratio between the number of images where both c_u and c_v appear and the total number of images in the collection where either c_u or c_v appear and its value ranges to $0 \leq a_{uv} \leq 1$. If c_u and c_v have many co-occurrences in images, the value of a_{uv} increases and the images are considered to be more correlated. This matrix is termed as a *connection matrix* in Yasushi et al. (1991), which was successfully utilized in a fuzzy information retrieval approach. Finally we can easily replaces the above matrices with the distance matching function in (6) and perform the similarity search effectively.

4. Query expansion in a modified inverted index

The distance measure described in Section 3, computes the cross correlations/similarities between the concepts, hence it requires longer computational time as compared to the L-norm (e.g., Euclidean) or cosine based distance measures. One solution is to compare only a subset of images from the entire collection. In large database applications, the indexing or pre-filtering techniques are essential to avoid exhaustive search in the entire collection Gaede & Gunther (1998). The inverted file is a very popular indexing technique for the vector space model in IR Yates & Neto (1999). An inverted file contains an entry for every possible terms and each term contains a list of the documents if the documents have at least one occurrence of that particular term. In CBIR domain, an inverted index is used in a suitable sparse set of color and texture feature space of dimension more then ten thousands in Müller et al. (1999). Motivated by their success, we present an enhanced inverted index to reduce the search time that considers the similarities between the visual concept prototypes by exploiting the topology preserving property of the SOM generated codebook. Our goal is to decrease the response time where the codebook is acted as an inverted file to store

the mapping from concepts to images. In this index, for each visual concept prototype in a codebook, a list of pointers or references to images that have at least one region map to this concept is stored in a list. Hence, an image in the collection is a candidate for further distance measure calculations if it contains at least one region that corresponds to a concept c_i in a query image.
Now, to consider the similarity factor between the concepts, the simple lookup strategy in inverted index is modified slightly.

Definition 4 *Each visual concept prototype $c_j(x,y) \in C$ has a local γ-neighborhood $LN_\gamma(x,y)$ in a two-dimensional grid of codebook as depicted in Figure 3. We have*

$$LN_\gamma(x,y) = \{c_k(u,v) : \max\{|u-x|,|v-y|\}\} \tag{8}$$

Here, the coordinates (x,y) and (u,v) denote the row and column-wise position of any two concept prototypes c_j and c_k respectively where $x,u \in \{1,\cdots,P\}$ and $y,v \in \{1,\cdots,P\}$ for a codebook of size $N = P \times P$ units. The value of γ can be from 1 up to a maximum of $P-1$.

In this approach, for each concept prototype $c_i \in I_q$ with a weight (e.g., *tf-idf* based weighting) w_{iq}, we expand it to other $\lfloor w_{iq} \times (|S_\gamma| - 1) \rfloor$ concept prototypes based on the topology preserving ordering in a codebook. Here, S_γ contains all the concept prototypes including c_i up to a local neighborhood level LN_γ. For example, Figure 3 shows the local neighborhood structure of a concept prototype in a two-dimensional codebook based on Definition 4. Here, each concept prototype is visualized as a circle on the grid and the black circle in the middle denotes a particular concept prototype $c_j(x,y)$. The concept prototype $c_k(u,v)$ is three neighborhood level (e.g., LN_3) apart from $c_j(x,y)$ based on Definition 4 as the maximum distance between them (coordinate-wise) either in horizontal or vertical direction is three. Basically, all the gray circles within the square are positioned in the LN_1 neighborhood, the gray and yellow circles are positioned up to LN_2 and gray, yellow and blue circles in combine are positioned up to LN_3 neighborhoods of c_j as shown in the Figure 3. As the value of γ increases, the number of neighboring concept prototypes increases for c_j.
For the query expansion, the concepts other then c_i are considered by subtracting it from S_γ. After the expansion, those images that appear in the list of expanded concepts are deemed

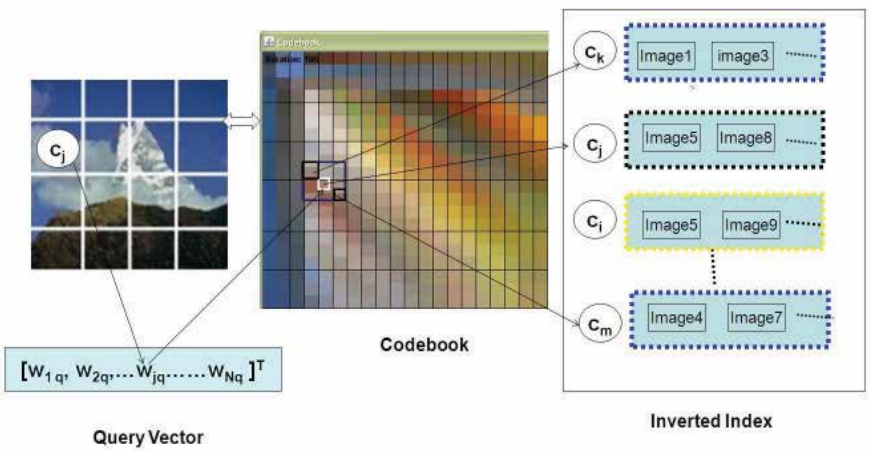


Fig. 4. Example process of Query Expansion in an Inverted File

Algorithm 1 Similarity Matching in Modified Inverted File

1: Initially compute the global similarity matrix **S** off-line. Let, the feature vector of a query image I_q be $\mathbf{f}_q = [w_{1q} \cdots w_{iq} \cdots w_{Nq}]^{\mathrm{T}}$ in a visual concept-based feature space. Initialize the list of candidate image as $L = \phi$.
2: **for** $i = 1$ to N **do**
3: **if** $w_{iq} > 0$ (i.e., $c_i \in I_q$) **then**
4: Locate the corresponding concept prototype c_i in the two-dimensional codebook C.
5: Read the corresponding list L_{c_i} of images from the inverted file and add it to L as $L \leftarrow L \cup L_{c_i}$.
6: Consider up to LN_γ neighborhoods of c_i to find related $|S_\gamma| - 1$ concept prototypes.
7: For each $c_j \in S_\gamma - \{c_i\}$, determine its ranking based on the similarity values by looking up corresponding entry s_{ij} in matrix **S**.
8: Consider the top $k = \lfloor w_{iq} \times (|S_\gamma| - 1) \rfloor$ ranked concept prototypes in set S^k for further expansion.
9: **for** each $c_k \in S^k$ **do**
10: Read the corresponding list as $L(c_k)$ and add to L as $L \leftarrow L \cup L_{c_k}$ after removing the duplicates.
11: **end for**
12: **end if**
13: **end for**
14: **for** each $I_j \in L$ **do**
15: Apply the distance matching functions of Equation (6) between I_q and I_j based on the Matrix **S** or **A** .
16: **end for**
17: Finally, return the top K images by sorting the distance measure values in ascending order (e.g., a value of 0 indicates closest match).

as candidates for further distance measure calculations while the other images are ignored. A larger γ will lead to more expanded concepts, which means that more images need to be compared with the query. This might lead to more accurate retrieval results in a trade off of the larger computational time. After finding the $|S_\gamma| - 1$ concept prototypes, they are ranked based on their similarity values with c_i by looking up the corresponding entry in the matrix $\mathbf{S}^*$. This way the relationship between two concepts are actually determined by both their closeness in the topology preserving codebook and their correlation or similarity obtained from the global matrices. Finally, the top $\lfloor w_{iq} \times (|S_\gamma| - 1) \rfloor$ concepts are selected as expanded concepts for c_i. Hence, a concept with more weight in a query vector will be expanded to the more closely related concepts and as a result will have more influence to retrieve candidate images. Therefore, the enhanced inverted index contains an entry for a concept that consists of a list of images as well as images from closely related concepts based on the local neighborhood property. The steps of the above process are describes in Algorithm 1.

Figure 4 shows an example of the above processing steps. Here, for a particular concept c_j with the associated weight in vector as w_{jq} that is presented in the query image I_q, the corresponding location of the concept in the codebook is found out. Suppose, based on the LN_1 neighborhood of the above algorithm, only two concepts c_k and c_m are further selected for expansion. After finding the expanded concept prototypes, the images in their inverted

lists are merged with the original set of images and considered for further distance measure calculation for ranked-based retrieval. Therefore, in addition to considering all the images in the inverted list of c_j (images under black dotted rectangle), we also need to consider the images in the list of c_k and c_m (under the blue dotted rectangle) as candidate images. Due to the space limitations, all the actual links are not shown in Figure 4. In this way, the response time is reduced while the retrieval accuracy is still maintained.

5. Experiments

To evaluate the effectiveness and efficiency of the proposed concept-based image representation and similarity matching approaches, exhaustive experiments were performed on two different benchmark image collections under ImageCLEF [1]. The first collection is termed as the IAPR TC-12 benchmark that was created under the Technical Committee 12 (TC-12) of the International Association of Pattern Recognition (IAPR) Grubinger et al. (2006) and it has been used for the ad-hoc photographic retrieval task in ImageCLEF'07 Grubinger et al. (2007). This collection is publicly available for research purposes and currently contains around 20,000 photos taken from locations around the world that comprises a varying cross-section of still natural images. The domain of the images in this collection is very generic that covers a wide range of daily life situations. Unlike the commonly used COREL images, this collection is very general in content with many different images of similar visual content but varying the illumination, viewing angle and background. This making it more challenging for the successful application of image retrieval techniques.

The second collection contains more than 67,000 bio-medical images of different modalities from the RSNA journals Radiology and Radiographics under ImageCLEFmed'08 Müller et al. (2008). For each image, the text of the figure caption is supplied as free text. The contents of this collection represent a broad and significant body of medical knowledge, which make the retrieval more challenging. As the entire collection contains variety of imaging modalities, image sizes, and resolutions, it makes really difficult to perform similarity search based on the current CBIR techniques.

The training set used for the SOM-based learning consists of only around 1% images of each individual data set. We set the initial learning rate as $\alpha = 0.07$ and iteration number as 300 due to their better performances. Based on the retrieval results, we found the the optimal combination when the images are partitioned as 64 sub-images and a codebook size of 225 (15×15) units is used. Hence, the images are indexed based on this configuration for both the collections for the experimental pupose.

For a quantitative evaluation of the retrieval results, we used *"query-by-example"* as the search method where the query images are specified by the topics that were developed by the CLEF organizers. Each topic is a short sentence or phrase describing the search topic with one to three "relevant" images. The query topics are equally subdivided into three categories: visual, mixed, and semantic Grubinger et al. (2007); Müller et al. (2008). A total of 60 topics were provided by the ImageCLEF'07 Grubinger et al. (2007) for the ad-hoc retrieval of general photographic images with a short description of each topic. Similarly, for the ad-hoc medical image retrieval in ImageCLEF'08, a total of 30 query topics were provided Müller et al. (2008) that were initially generated based on a log file of Pubmed [2].

Results for different methods are computed using the latest version of TREC-EVAL [3]

[1] http://imageclef.org/

[2] http://www.pubmed.gov

[3] http://trec.nist.gov/$trec-eval$/

Method	MAP	GMAP	P(20)
EHD-Euclid	0.0131	0.0028	0.0467
CLD-Euclid	0.0198	0.0040	0.0642
Concept-Euclid	0.0164	0.0028	0.0525
Concept-Quad(A)	0.0179	0.0037	0.0583
Concept-Quad(S)	0.0178	0.0035	0.0575
Concept-Quad(A)-IF	0.0176	0.0036	0.0581
Concept-Quad(A)-IF(LN_2)	0.0178	0.0037	0.0583

Table 1. Retrieval Results (Photographic Collection)

software based on the relevant sets of all topics, which were crated by the CLEF organizers by considering top retrieval results of all submitted runs of the participating groups in ImageCLEF'08 Müller et al. (2008); Grubinger et al. (2007). Results were evaluated using un interpolated (arithmetic) Mean Average Precisions (MAP)to test effectiveness, Geometric Mean Average Precision (GMAP) to test robustness, and Precision at rank 20 (P20) because most online image retrieval engines like Google, Yahoo, and Altavista display 20 images by default.

6. Results

This section presents the experimental results of the retrieval approaches with and without using the correlation-enhanced similarity matching and inverted indexing schemes. The performances of the different search schemes are shown in Table 1 and Table 2 for the retrieval of the photographic and medical collections respectively based on the query image sets as discussed previously. The proposed correlation-enhanced similarity matching approach is compared (using different matrices) to the case when only Euclidean distance measure is used in the visual concept-based feature (e.g., method "Concept-Euclid") space. In addition, we consider the MPEG-7 based Edge Histogram Descriptor (EHD) and Color Layout Descriptor (CLD) Chang et al. (2001) (e.g., methods "EHD-Euclid" and "CLD-Euclid") and compare our search approach with these features based on the Euclidean distance measure. The EHD

Method	MAP	GMAP	P(20)
EHD-Euclid	0.0095	0.0004	0.0333
CLD-Euclid	0.0006	0.0001	0.0050
Concept-Euclid	0.0248	0.0009	0.0300
Concept-Quad(A)	0.0255	0.0014	0.0433
Concept-Quad(S)	0.0256	0.0012	0.0467
Concept-Quad(A)-IF	0.0251	0.0012	0.0430
Concept-Quad(A)-IF(LN_2)	0.0254	0.0014	0.0433

Table 2. Retrieval Results (Medical Collection)

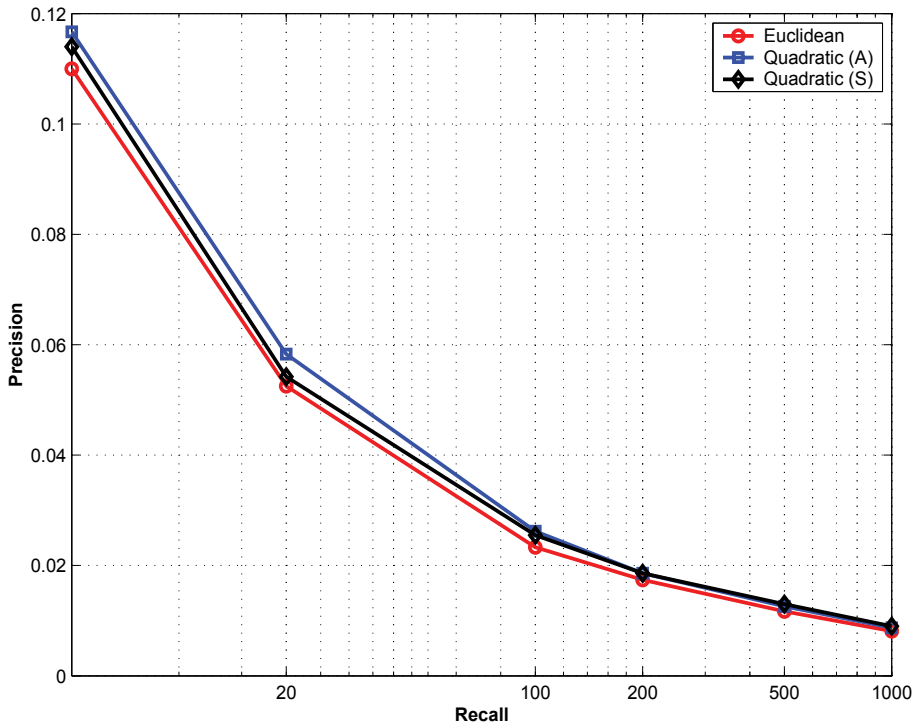


Fig. 5. Precisions at different ranks for the photographic collection

represents local edge distribution in an image by dividing the image into 4×4 sub-images and generating a histogram from the edges present in each of these sub-images. The CLD represents the spatial layout of the images in a very compact form. It is obtained by applying the discrete cosine transformation (DCT) on the 2-D array of local representative colors in the *YCbCr* color space where *Y* is the luma component and *Cb* and *Cr* are the blue and red chroma components Chang et al. (2001).

We can observe from Table 1 and Table 2 that the retrieval performances in general have improved on visual concept space for both the collections based on the different performance measures. Only the MPEG-7 based CLD feature performed better with a MAP score of 0.0198 in the photographic image collection due to the presence of many color images of natural scenery, whereas it performed worst in the medical collection due to the lack of presence of such images. On the other hand, the performance of the visual concept feature is quite consistent in both the collections due to the incorporation of both color and texture features for the codebook generation process based on the SOM learning. Overall, the retrieval results in both the collections in terms of different performance measures are quite comparable with the results of the participant of the previous ImageCLEF Müller et al. (2008); Grubinger et al. (2007). The low precision scores as obtained are due to the nature of the image collections and the complexity of the query topics.

In addition, we can observe the improvement in performances in terms of the MAP, GMAP, and BPREF scores, when searches are performed with different correlation-enhanced similarity matching functions based on using the global matrices: $\mathbf{S}$ (e.g., method "Concept-Quad(S)"), and matrix $\mathbf{A}$ (e.g., method "Concept-Quad(A)"), for both collections.

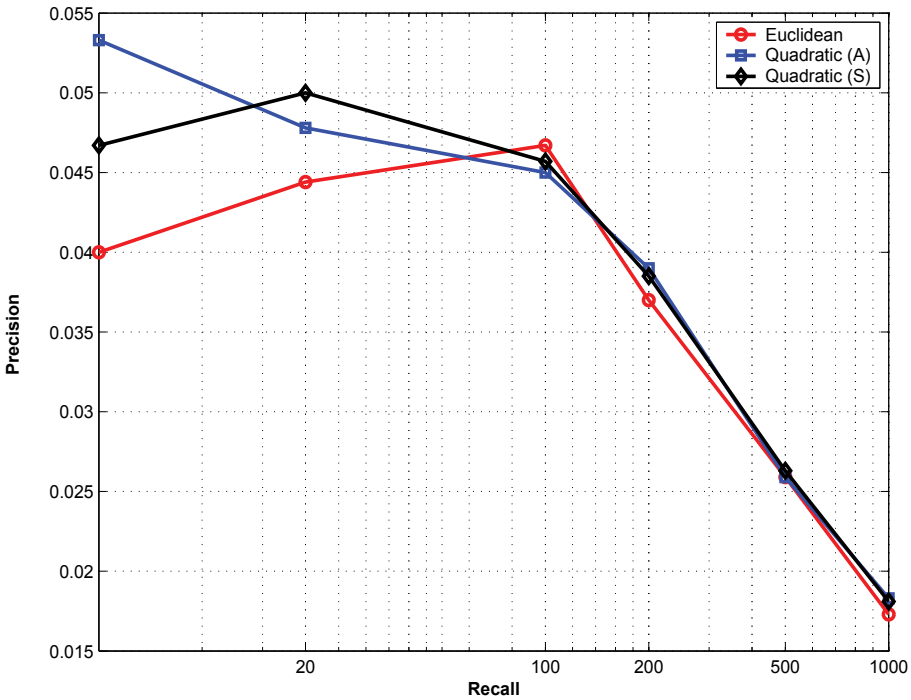


Fig. 6. Precisions at different ranks for the medical collection

Figure 5 and Figure 6 show the precision at different rank position (e.g., 5, 20, 100, 200, 500, and 1000) for the photographic and medical collections respectively. For a better visibility, the X-axis is represented by the logarithmic scale for both the figures. Although the precision curves are looking different in Figure 5 and Figure 6, we can conjecture one thing about the nature of the curves. It is that the precision in initial rank positions (up to 100) are comparatively better (especially in medical collection) for the quadratic distance measures when compared to the Euclidean distance measure in the concept space. This improvement in performance is important as users are usually interested only on the first few top retrieved images.

It is also noticeable that the performances of the retrieval result are decreased slightly for both collections when searches are performed in the inverted index with the quadratic distance measure based on the global connection matrix (e.g., method "Concept-Quad(A)-IF") and the performances are almost comparable to linear search when the matching are performed in the modified inverted index based on a local neighborhood of LN_2. The major gain in searching on a inverted index is that it takes less computational time compared to a linear search in the entire collections. Hence, to test the efficiency of the search schemes for the concept-based feature, we also compared the average retrieval time (in milliseconds) with and without the indexing scheme (in an Intel Pentium 4 processor with Windows XP as the operating system and 2 GB memory) for both the query sets.

From the results in Table 3, it is observed that the search with the quadratic distance measure with global matrix $\mathbf{A}$ in the inverted index of concept feature, is about two times faster as compared to the linear search for both the data sets. Although, it took longer time to perform search in the medical collection due to the presence of around three times more images

DataSet	Euclid-Linear	Quad-Linear	Quad (A)-IF	Quad (A)-IF(LN_2)
Photograph	126	800	436	520
Medical	580	2470	1320	1570

Table 3. Average retrieval time (ms) for the query images

compared to the photographic collection. However, the percentage of improvement in terms of using an inverted index is almost same for both the collections. In addition, we found a trade-off in between the time and precision by performing the same search in the modified inverted index with a local neighborhood of LN_2. By observing closely in Table 1, Table 2, and Table 3, it is found that the MAP scores are increased slightly with the expense of a little longer search time when compared to the search without the modification of the inverted index. Hence, the quadratic distance matching in the modified inverted index with the query expansion has proved to be both effective and efficient.

7. Conclusions

We have investigated a correlation-enhanced similarity matching and a query expansion technique in CBIR domain, inspired by the ideas of the text retrieval domain in IR. The proposed technique exploit the similarities/correlations between the concepts based on a global analysis approach. Due to the nature of the image representation schemes in the concept-based feature spaces, there always exists enough correlations between the concepts. Hence, exploiting this property improved the retrieval effectiveness. For the feature representation, we limited our approaches by modeling only the intermediate level visual concepts. This limitation is obvious due to the current state of the object recognition techniques in the broad domain images. It would be more effective, if specific objects can be identified in large collections irrespective of their variations and occlusions. However, the main focus of our approach is to perform the retrieval that can exploit concept correlations at the global level. In future, when the object recognition techniques will be mature enough to a certain level, our approaches could be easily extendible to a higher level concept-based representation.

8. References

Smeulders, A.; Worring, M.; Santini, S.; Gupta, A. & Jain, R. (2000). Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 12, 2000, 1349–1380

Liua, Y.; Zhang, D.; Lu, G.; & and Ma, W. Y. (2007). A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, Vol. 40, 2007, 262–282

Datta, R.; Joshi, D.; Li, J.; & Wang, J. Z. (2008). Image Retrieval: Ideas, Influences, and Trends of the New Age. *ACM Computing Surveys*, Vol. 40, No. 2, 2008, 1–60

Zhu, L.; Zhang, A.; Rao, A.; & Srihari, R. (2002). Theory of keyblock-based image retrieval. *ACM Transactions on Information Systems*, Vol. 20, No. 2, April 2002, 224–257, ISSN:1046-8188

Lim, J. H. (2001). Building visual vocabulary for image indexation and query formulation. *Pattern Analysis & Applications (Special Issue on Image Indexation)*, Vol. 4, No. 2-3, 125–139, DOI: 10.1007/PL00014574

Jing, F.; Li, M.; Zhang, H. J. & Zhang, B. (2004). An efficient and effective region-based image

retrieval framework. *IEEE Transaction on Image Processing*, Vol. 13, No. 5, May 2004, 699–709, PMID: 15376601

Vogel, J. & Schiele, B. (2007). Semantic Modeling of Natural Scenes for Content-Based Image Retrieval. *International Journal of Computer Vision* , Vol. 72, No. 2, April 2007, 133–157, ISSN:0920-5691

Shi, R.; Feng, H.; Chua, T. S. & Lee, C. H. (2004). An adaptive image content representation and segmentation approach to automatic image annotation. *Proceedings Lecture Notes in Computer Science*, Vol 3115, Springer 2004, ISBN 3-540-22539-0, Third International Conference on Image and Video Retrieval (CIVR 2004), Dublin, Ireland, July 21-23, 2004.

Rahman, M. M.; Bhattacharya, P. & Desai, B. C. (2009). A Unified Image Retrieval Framework on Local Visual and Semantic Concept-Based Feature Spaces. *Journal of Visual Communication and Image Representation*, Vol. 20, No. 7, October 2009, 450–462, ISSN:1047-3203

Rahman, M. M.; Antani, S. K.; & Thoma, G. R. (2009). A Medical Image Retrieval Framework in Correlation Enhanced Visual Concept Feature Space. *Proceedings of 22nd IEEE International Symposium on Computer-Based Medical Systems (CBMS)*, August 3-4, 2009, Albuquerque, New Mexico, USA.

Bosch, A.; Muoz, X.; & Mart, R. (2007). Which is the best way to organize/classify images by content?. *Image and Vision Computing*, Vol. 25, No. 6, June 2007, 778–791, doi:10.1016/j.imavis.2006.07.015

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*, 2nd Ed. Academic Press, ISBN:0-12-269851-7

Kohonen, T. (1997). *Self-Organizating Maps*, New York, Springer-Verlag, ISBN: 978-3-540-67921-9

Yates, R. B. & Neto, B. R. (1999). *Modern Information Retrieval*, 1st ed., Addison Wesley, ISBN-13: 978-0201398298

Yasushi, O.; Tetsuya, M. & Kiyohiko, K. (1991). A fuzzy document retrieval system using the keyword connection matrix and a learning method. *Fuzzy Sets and Systems*, Vol. 39, No. 2, 163–179

Haralick, R. M.; Shanmugam & Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-3, No. 6, 610-621

Chang, S. F.; Sikora, T. & Puri, A. (2001). Overview of the mpeg-7 standard. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 11, No. 6, June 2001, 688–695, ISSN: 1051-8215

Laaksonen, J.; Koskela, M. & Oja, E. (2002). PicSOM: Self-Organizing Image Retrieval With MPEG-7 Content Descriptors. *IEEE Transaction On Neural Networks*, Vol. 13, No, 4, Jul 2002, 841–853, ISSN: 1045-9227

Vesanto, J. (1999). SOM-Based Data Visualization Methods, *Intelligent Data Analysis*, Vol. 3, No. 2, August 1999, 111–126, doi:10.1016/S1088-467X(99)00013-X

Yen, G. G. & Zheng, W. (2005). Ranked Centroid Projection: A Data Visualization Approach With Self-Organizing Maps, *IEEE Transaction On Neural Networks*, Vol. 19, No. 2, Feb. 2008, 245–259, ISSN: 1045-9227

Gaede, V. & Gunther, O. (1998). Multidimensional Access Methods. *ACM Computing Surveys*, Vol. 30, No. 2, June 1998, 170–231, ISSN:0360-0300

Müller, H.; Squire, D. M.; Mueller, W. & Pun, T. (1999). Efficient access methods for content-based image retrieval with inverted files. *Proceedings of SPIE*, Vol. 3846,

461–472, Multimedia Storage and Archiving Systems IV, Sethuraman Panchanathan; Shih-Fu Chang; C.C. J. Kuo; Eds.

Hafner, J.; Sawhney, H. S.; Equitz, W.; Flickner, M. & Niblack, W. (1995). Efficient color histogram indexing for quadratic form distance functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 7, Jul 1995, 729-736, ISSN: 0162-8828

Grubinger, M.; Clough, P.; Müller, H. & Deselears, T. (2006). The IAPR-TC12 benchmark: A new evaluation resource for visual information systems. *Proceedings of International Workshop on Language Resources for Content-Based Image Retrieval*, 13–23, 2006.

Grubinger, M.; Clough, P.; Hanbury, A. & Müller, H. (2007). Overview of the ImageCLEF 2007 Photographic Retrieval Task. *Working Notes of the 2007 CLEF Workshop,* Sep., 2007, Budapest, Hungary.

Müller, H.; Cramer, J. K.; Kahn, C. E.; Hatt, W.; Bedrick, S. & Hersh, W. (2008). Overview of the ImageCLEFmed 2008 medical image retrieval task. *9th Workshop of the Cross-Language Evaluation Forum, CLEF 2008*, Aarhus, Denmark.

11

Mel-Frequency Cepstrum Coefficients as Higher Order Statistics Representation to Characterize Speech Signal for Speaker Identification System in Noisy Environment using Hidden Markov Model

Agus Buono[1], Wisnu Jatmiko[2] and Benyamin Kusumoputro[3]
[1]*Computer Science Department, Bogor Agriculture University,*
[2]*Faculty of Computer Science, University of Indonesia,*
[3]*Faculty of Engineering, University of Indonesia*
Indonesia

1. Introduction

Sound is an effective and efficient magnitude for biometric characterization. However, the sound is a phenomenon that is a fusion of multidimensional and influenced many aspects, such as speaker characteristics (articulator configuration, emotions, health, age, sex, dialect), languages, and the environment (background and transmission media), so that the system has been developed until now has not been able to work well in real situations. This is the background of this research.

In this research, we investigate higher order statistics (HOS) and Mel-Frequency Cepstrum Coefficients (MFCC) as a feature extraction, and integrated with a Hidden Markov Model (HMM) as a classifier to get a more robust speaker identification system, especially for Gaussian Noise. Research carried out more focused on feature extraction part of the speaker identification system. In classifier process stage, we use the HMM. This is a technique that has been widely used in voice processing provides good results. At the beginning, we empirically showed the failure of conventional MFCC using power spectrum in noisy environment. Then proceed with reviewing the matter, and proposed HOS-based extraction techniques to overcome these problems. Next is an experiments to demonstrate the effectiveness of the proposed method. Data used in this study came from 10 people who say the phrase 'PUDHESA' as much as 80 times with different ways of utterance. In this research, we use signals that are spoken with different variations of pressure, duration, emotional, loud and weak. Figure 1 presents the forms of signals for different utterances of a speaker. In accordance with the focus of this research is to build models that are more robust to noise, then we add a Gaussian noise signal to each original signal with a signal-to-noise ratio (SNR) of 20 dB, 10 dB, 5 dB and 0 dB.

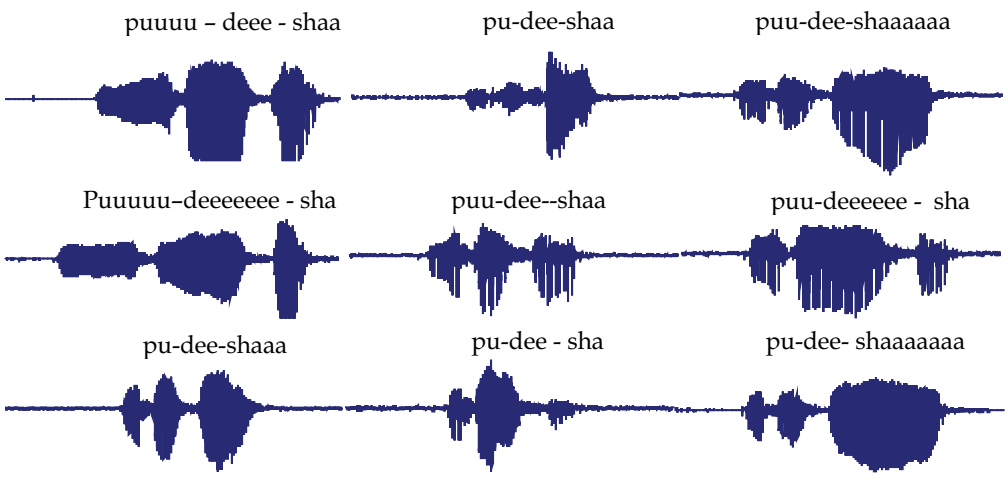


Fig. 1. Various forms of signals according to a speaker utterance mode

Figure 2 presents a comparison between the original speech signal with the original signal that has been contaminated by gaussian noise signal with a level of 20 dB, 10 dB, 5 dB and 0 dB. From the pictures it can be seen that the more severe the noise is given, then the more the signal is distorted from its original form.

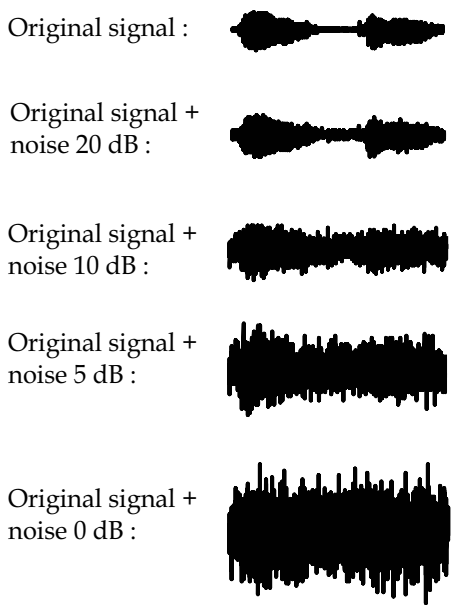


Fig. 2. Comparison of the original signal with the signal that is contaminated by noise

3. Speaker identification system

3.1 Overview

Speaker identification is an automatic process to determine who the owner of the voice given to the system. Block diagram of speaker identification system are shown in Figure 3. Someone who will be identified says a certain word or phrase as input to the system. Next, feature extraction module calculates features from the input voice signal. These features are processed by the classifier module to be given a score to each class in the system. The system will provide the class label of the input sound signal according to the highest score.

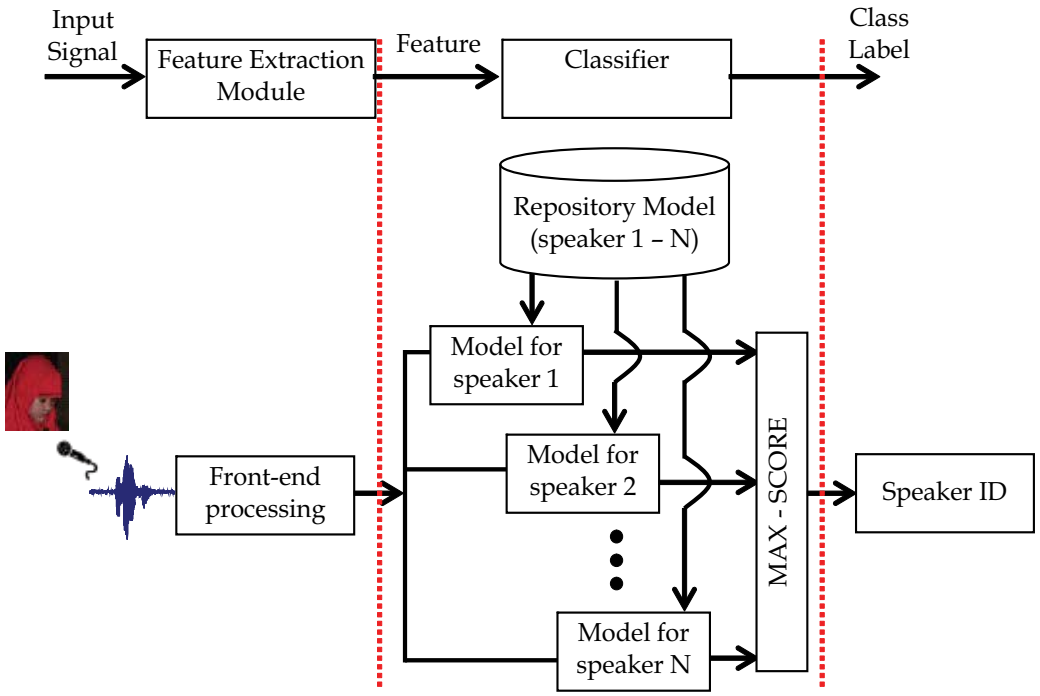


Fig. 3. Block diagram of speaker identification system

Input to the speaker identification system is a sound wave signal. The initial phase is to conduct sampling to obtain digital signals from analogue voice signal. Next perform quantization and coding. After the abolition of the silence, these digital signals are then entered to the feature extraction module. Voice signals are read from frame to frame (part of signal with certain time duration, usually 5 ms up to 100 ms) with a certain length and overlapped for each two adjacent frames. In each frame windowing process is carried out with the specified window function, and continued with the process of feature extraction. This feature extraction module output will go to the classifier module to do the recognition process. In general there are four methods of classifier (Reynold, 2002), namely: template matching, nearest neighbour, neural network and hidden Markov model (HMM). With the template matching method, the system has a template for each word/speaker. In the nearest neighbour, the system must have a huge memory to store the training data. While the neural network model is less able to represent how the sound signal is produced naturally. In the Hidden Markov Model, speech signal is statistically modelled, so that it can represent how

the sound is produced naturally. Therefore, this model was first used in modern speaker recognition system. In this research we use the HMM as a classifier, so the features of each frame will be processed sequentially.

3.2 MFCC as feature extraction

Feature extraction is the process for determining a value or a vector that can characterize an object or individual. In the voice processing, a commonly used feature is the cepstral coefficients of a frame. Mel-Frequency Cepstrum Coefficients (MFCC) is a classical feature extraction and speech parameterization technique that widely used in the area of speech processing, especially in speaker recognition system.

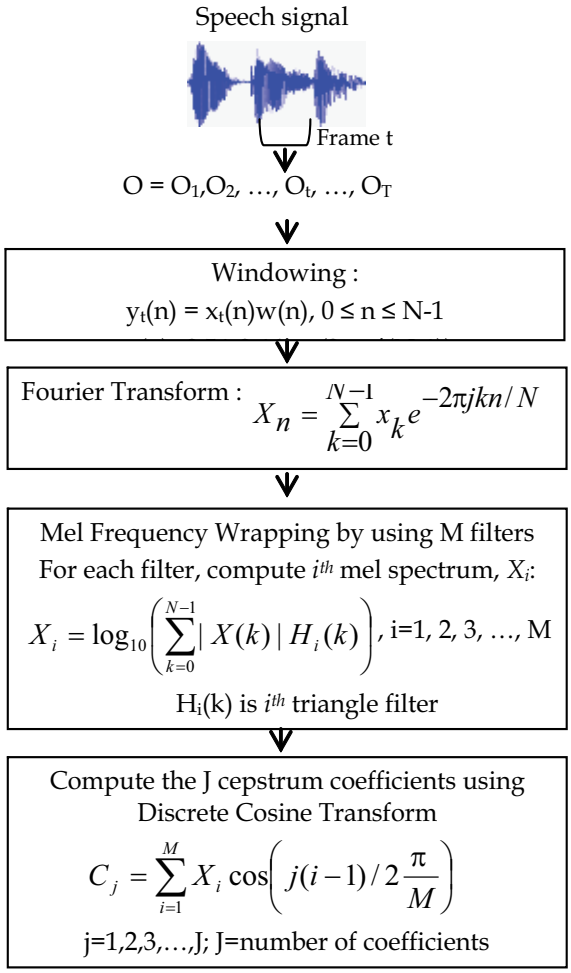


Fig. 4. MFCC process flowchart

Compare to other feature extraction methods, Davis and Mermelstein have shown that MFCC as a feature extraction technique gave the highest recognition rate (Ganchev, 2005). After its introduction, numerous variations and improvements of the original idea are

developed; mainly in the filter characteristics, i.e, its numbers, shape and bandwidth of filters and the way the filters are spaced (Ganchev, 2005). This method calculates the cepstral coefficients of a speech signal by considering the perception of the human auditory system to sound frequency. Block diagram of the method is depicted in Figure 4. For more detailed explanation can be read in (Ganchev, 2005) and (Nilsson, M & Ejnarsson, 2002).
After a process of windowing and Fourier transformation, performed wrapping of signals in the frequency domain using a number of filters. In this step, the spectrum of each frame is wrapping using M triangular filter with an equally highest position as 1. This filter is developed based on the behavior of human ear's perception, in which a series of psychological studies have shown that human perception of the frequency contents of sounds for speech signal does not follow a linear scale. Thus for each tone of a voice signal with an actual frequency f, measured in Hz, it can also be determined as a subjective pitch in another frequency scale, called the 'mel' (from Melody) scale, (Nilsson, M & Ejnarsson, 2002). The mel-frequency scale is determined to have a linear frequency relationship for f below 1000 Hz and a logarithmic relationship for f higher than 1000Hz. One most popular formula for frequency higher than 1000 Hz is, (Nilsson, M & Ejnarsson, 2002):

$$\hat{f}_{mel} = 2595 * \log_{10}\left(1 + \frac{f}{700}\right) \tag{1}$$

as illustrated by Figure 5 below:

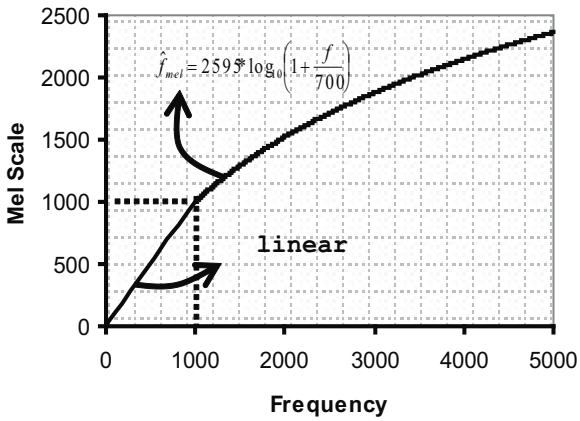


Fig. 5. Curve relationship between frequency signal with its mel frequency scale

Algorithm 1 depicted the process for develop those M filters, (Buono et al., 2008).
Algorithm 1: Construct 1D filter

a. Select the number of filter (M)
b. Select the highest frequency signal (f_{high}).
c. Compute the highest value of $\hat{f}_{mel}$:

$$\hat{f}_{mel}^{high} = 2595 * \log_{10}\left(1 + \frac{f_{high}}{700}\right)$$

d. Compute the center of the i^{th} filter (f_i), i.e.:

d.1. $f_i = \frac{1000}{0.5 * M} * i$ for i=1, 2, 3, …, M/2

d.2. for i=M/2, M/2+1, …, M, the f_i formulated as follow :

1. Spaced uniformly the mel scale axis with interval width Δ, where:

$$\Delta = \frac{\hat{f}_{mel}^{high} - 1000}{0.5 * M}$$

According to the equation (1), the interval width Δ can be expressed as:

$$\Delta = \frac{5190}{M} \log\left(\frac{700 + f_{high}}{1700}\right)$$

2. The mel-frequency value for the center of ith filter is:

$$a = 1000 + (i - 0.5 * M) * \Delta$$

3. So, the center of ith filter in frequency axes is:

$$f_i = 700 * \left(10^{a/2595} - 1\right)$$

Figure 6 gives an example of the triangular i^{th} filter:

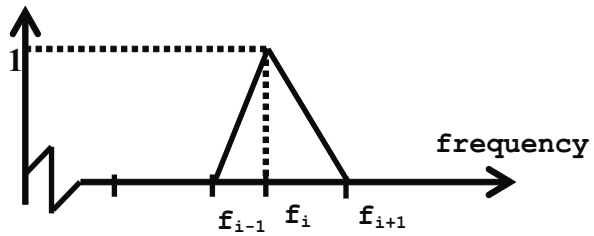


Fig. 6. A triangular filter with height 1

The mel frequency spectrum coefficients are calculated as the sum of the filtered result, and described by:

$$X_i = \log\left(\sum_{f=0}^{N-1} abs(X(j)) * H_i(f)\right) \quad (2)$$

where i=1,2,3,…,M, with *M* the number of filter; *N* the number of FFT coefficients; *abs(X(j))* is the magnitude of j^{th} coefficients of periodogram yielded by Fourier transform; and $H_i(f)$ is the i^{th} triangular at point *f*.

The next step is cosine transform. In this step we convert the mel-frequency spectrum coefficients back into its time domain using discrete cosine transform:

$$C_j = \sum_{i=1}^{M} X_i * \cos\left(\frac{j * (i - 0.5) * \pi}{20}\right) \quad (3)$$

where j=1,2,3,…,K, with *K* the number of coefficients; *M* the number of triangular filter; X_i is the mel-spectrum coefficients, as in (2). The result is called mel frequency cepstrum coefficients. Therefore the input data that is extracted is a dimensionless Fourier coefficients, so that for this technique we refer to as 1D-MFCC.

3.3 Hidden Markov model as classifier

HMM is a Markov chain, where its hidden state can yield an observable state. A HMM is specified completely by three components, i.e. initial state distribution, Л, transition probability matrix, A, and observation probability matrix, B. Hence, it is notated by λ = (A, B, Л), where, (Rabiner, 1989) and (Dugad & Desai, 1996):

A: NxN transition matrix with entries $a_{ij}=P(X_{t+1}=j \mid X_t=i)$, N is the number of possible hidden states

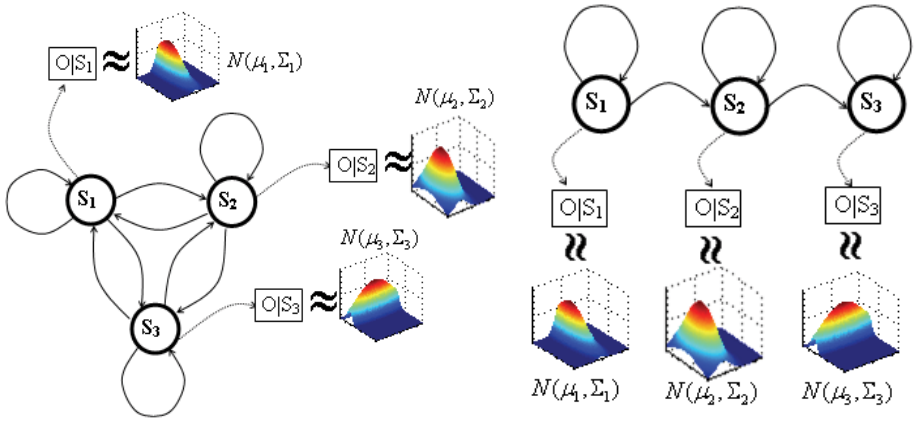
B: NxM observation matrix with entries $b_{jk}=P(O_{t+1}=v_k \mid X_t=j)$, k=1, 2, 3, …, M; M is the number of possible observable states

Л: Nx1 initial state vector with entries $\pi_i=P(X_1=i)$

For HMM's Gaussian, B consists of a mean vector and a covariance matrix for each hidden state, μ_i and Σ_i, respectively, i=1, 2, 3, …, N. The value of $b_j(O_{t+1})$ is $N(O_{t+1},\mu_j,\Sigma_j)$, where :

$$N(\mu_j,\Sigma_j)=\frac{1}{(2\pi)^{d/2}\,|\Sigma_j|^{1/2}}\exp\left[-\frac{1}{2}(O_{t+1}-\mu_j)\Sigma_j^{-1}(O_{t+1}-\mu_j)'\right] \tag{4}$$

There are three problems with HMM, (Rabiner, 1989), i.e. evaluation problem, $P(O \mid \lambda)$; decoding problem, $P(Q \mid O, \lambda)$; and training problem, i.e. adjusting the model parameters A, B, and Л. Detailed explanation of the algorithms of these three problems can be found in (Rabiner, 1989) and (Dugad & Desai, 1996).

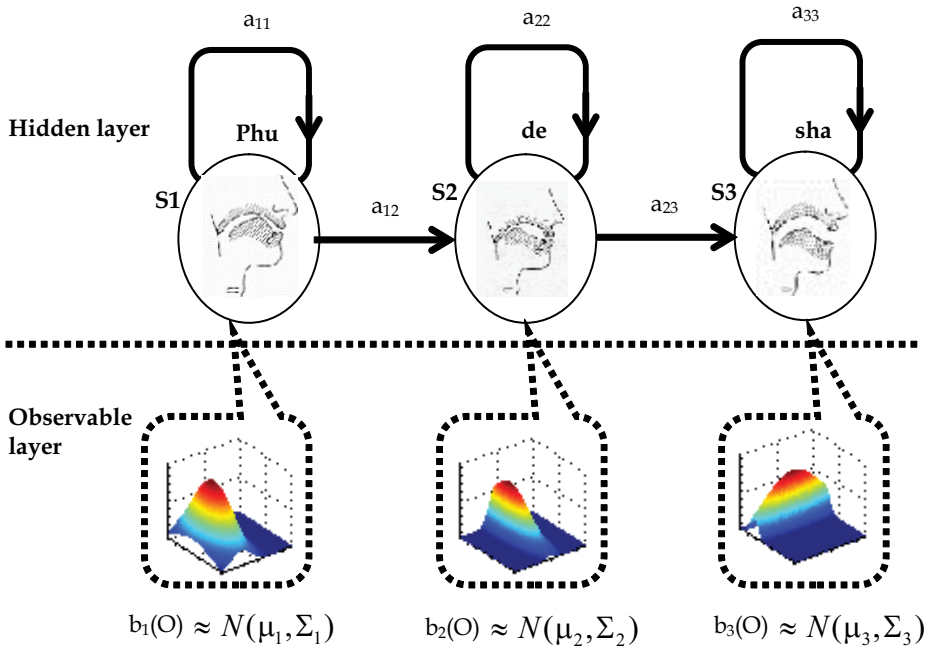


(a) (b)

Fig. 7. Example HMM with Three Hidden State and distribtion of the evidence variable is Gaussian, (a) Ergodic, (b) Left-Right HMM

In the context of HMM, an utterance is modeled by a directed graph where a node/state represents one articulator configuration that we could not observe directly (hidden state). A graph edge represents transition from one configuration to the successive configuration in the utterance. We model this transition by a matrix, A. In reality, we only know a speech signal produced by each configuration, which we call observation state or observable state. In HMM's Gaussian, observable state is a random variable and assumed has Normal or Gaussian distribution with mean vector μ_i and covariance matrix Σ_i (i=1, 2, 3, …, N; N is number of hidden states). Based on inter-state relations, there are two types of HMM, which

is ergodic and left-right HMM. On Ergodic HMM, between two states there is always a link, thus also called fully connected HMM. While the left-right HMM, the state can be arranged from left to right according to the link. In this research we use the left-right HMM as depicted by Figure 8.



Note : a_{ij} is the transition probability from state i into state j
$b_i(O)$ is the distribution of observable O given hidden state Si

Fig. 8. Left-Right HMM model with Three State to Be Used in this Research

3.4 Higher order statistics

If {x (t)}, t = 0, ± 1, ± 2, ± 3, ... is a stationary random process then the higher order statistics of order n (often referred as higher order spectrum of order n) of the process is the Fourier transform of $\{c_n^x\}$. In this case $\{c_n^x\}$ is a sequence of n order cumulant of the {x (t)} process. Detailed formulation can be read at (Nikeas & Petropulu, 1993). If n=3, the spectrum is known as bispectrum. In this research we use bispectrum for characterize the speech signal. The bispectrum, $C_3^x(\omega_1,\omega_2)$, of a stationary random process, {x (t)}, is formulated as:

$$C_3^x(\omega_1,\omega_2) = \sum_{\tau_1=-\infty}^{+\infty}\sum_{\tau_2=-\infty}^{+\infty} c_3^x(\tau_1,\tau_2)\exp\{-j(\omega_1\tau_1,\omega_2\tau_2)\} \quad (5)$$

where $c_3^x(\tau_1,\tau_2)$ is the cumulant of order 3 of the stationary random process, {x (t)}. If n=2, it is usually called as power spectrum. In 1D-MFCC, we use power spectrum to characterize the speech signal. In theory the bispectrum is more robust to gaussian noise than the power

spectrum, as shown in Figure 9. Therefore in this research we will conduct a development of MFCC technique for two-dimensional input data, and then we refer to as 2D-MFCC.
Basically, there are two approaches to predict the bispectrum, i.e. parametric approach and conventional approach. The conventional approaches may be classified into the following three classes, i.e. indirect technique, direct technique and complex demodulates method. Because of the simplicity, in this research we the conventional indirect method to predict the bispectrum values. Detail algorithm of the method is presented in (Nikeas & Petropulu, 1993).

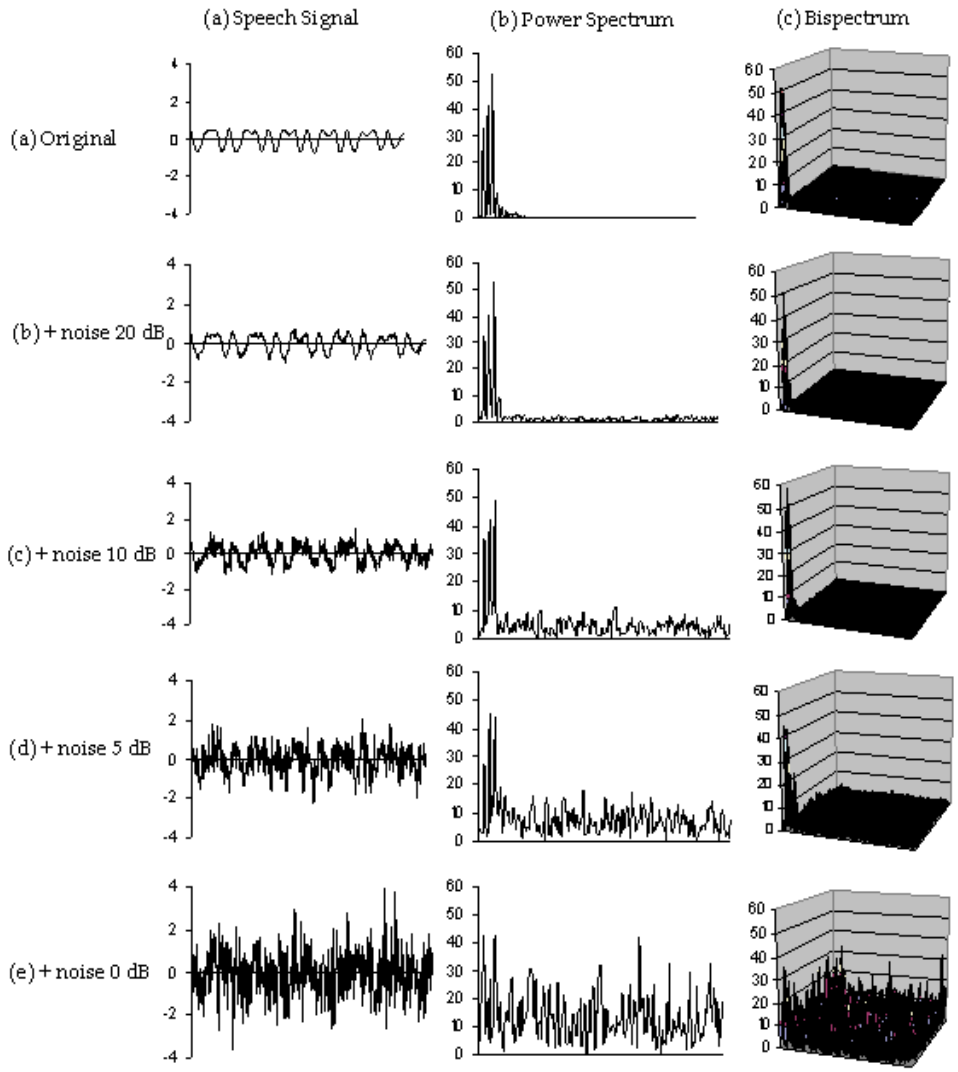


Fig. 9. Comparison between the power spectrum with the bispectrum for different noise

4. Experimental setup

First we show the weakness of 1D-MFCC based on power spectrum in capturing the signal features that has been contaminated by gaussian noise. Then we proceed by conducting two experiments with similar classier, but in feature extraction step, we use 2D-MFCC based on the bispectrum data.

4.1 1D-MFCC + HMM

Speaker identification experiments are performed to follow the steps as shown in Figure 10.

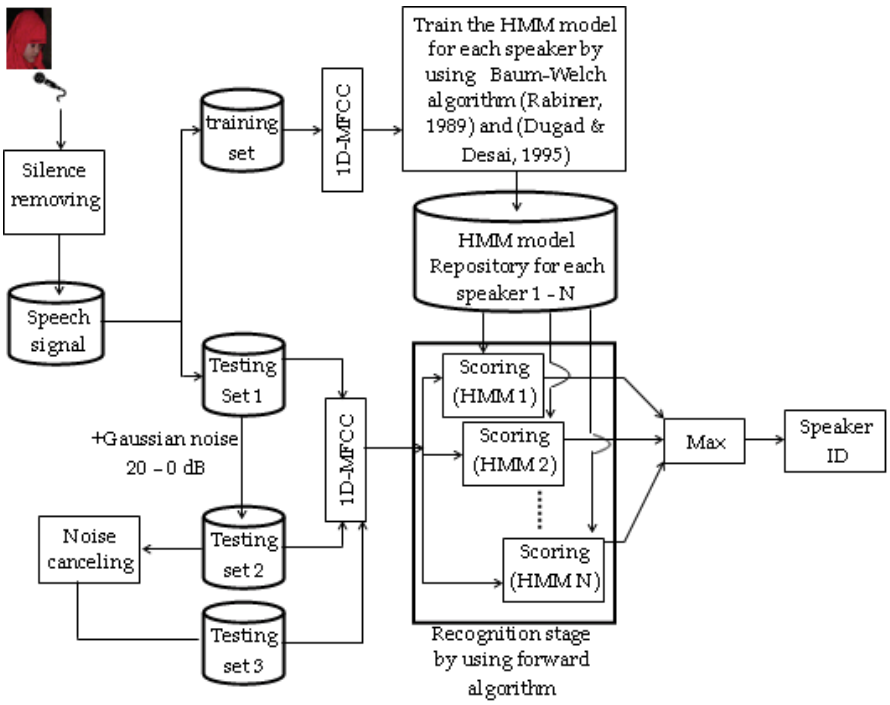


Fig. 10. Block diagram of experimental 1D-MFCC + HMM

The data used comes from 10 speakers each of 80 times of utterance. Before entering the next stage, the silence of the signal has been eliminated. Then, we divide the data into two sets, namely training data set and testing data set. There are three proportion values between training data and the testing data, ie 20:60, 40:40 and 60:20. Furthermore, we established three sets of test data, ie data sets 1, 2 and 3. Data set 1 is the original signal without adding noise. Data set 2 is the original signal by adding gaussian noise (20 dB, 10 dB, 5 dB and 0 dB), without the noise removal process. Data set 3 is the original signal by adding gaussian noise and noise removal process has been carried out with noise canceling algorithm, (Widrow et al., 1975) and (Boll, 1979). Next, the signal on each set (there are four sets, namely training data, testing data 1, testing data 2, and testing data 3) go into the feature extraction stage. In this case all the speech signals from each speaker is calculated its characteristic that is read frame by frame with a length 256 and the overlap between adjacent frames is 156, and forwarded to the appropriate stage of 1D-MFCC technique as

has been described previously. The next stage is to conduct the experiment according to the specified proportion, so that there are three experiments. In each experiment, in general there are two main stages, namely training stage and the recognition stage. In the training phase, we use the Baum-Welch algorithm to estimate the parameters of HMM, (Rabiner, 1989) and (Dugad & Desai, 1995). Data used in this training phase is the signal in training data that has been through the process of feature extraction. Our resulting HMM parameters stored in the repository, which would then be used for the recognition process. After the model is obtained, followed by speaker identification stage. In this case each signal on the test data (one test data, test data second and third test data) that has been through the process of feature extraction will be given a score for each speaker model. For a signal to be identified, compute the score for model 1 to model the N (N is the number of models in the repository). Score for model i, Si, is calculated by running the forward algorithm with the HMM model i. Further to these test signals will be labeled J, if $S_j > S_i$, for i=1,2,3, ...,j-1, j+1, ..., N.

Experimental result

Table 1 presents the accuracy of the system for various noise and various proportions of training data and test data.

Tipe of test data set	Training:test		
	20:60	40:40	60:20
Original signal	85.5	93.8	99.0
+noise 20 dB	37.0	41.1	52.8
+noise 10 dB	14.4	15.4	22.5
+noise 5 dB	12.7	13.8	17.3
+noise 0 dB	10.4	10.0	11.3

Table 1. The accuracy of the system at various proportions of training data and test data

From the table it can be said that for the original signal, the system with feature extraction using 1D-MFCC and HMM as a classifier able to recognize very well, which is around 99% for the original data on the proportion of 75% training data. The table also shows that with increasing noise, the accuracy drops drastically, which is to become 52% to 20 dB noise, and for higher noise, the accuracy below 50%. It is visually apparent as shown in Figure 11. The failure of this system is caused by the power spectrum is sensitive to noise, as shown in Figure 9 above.

To see the effect of number of hidden states to the degree of accuracy, in this experiment, the number of hidden state in HMM model varies from 3 to 7. Based on the results, seen that level of accuracy for the original signal is ranged from 99% to 100%. This indicates that the selection of number of hidden state in HMM does not provide significant effect on the results of system accuracy.

Table 1 also indicates that the amount of training data will affect the HMM parameters that ultimately affect the accuracy of the system. In this research, a signal consisting of about 50 frames. Therefore, to estimate HMM parameters that have a state of 3 to 7 is required sequence consisting of 3000 (50x60) samples.

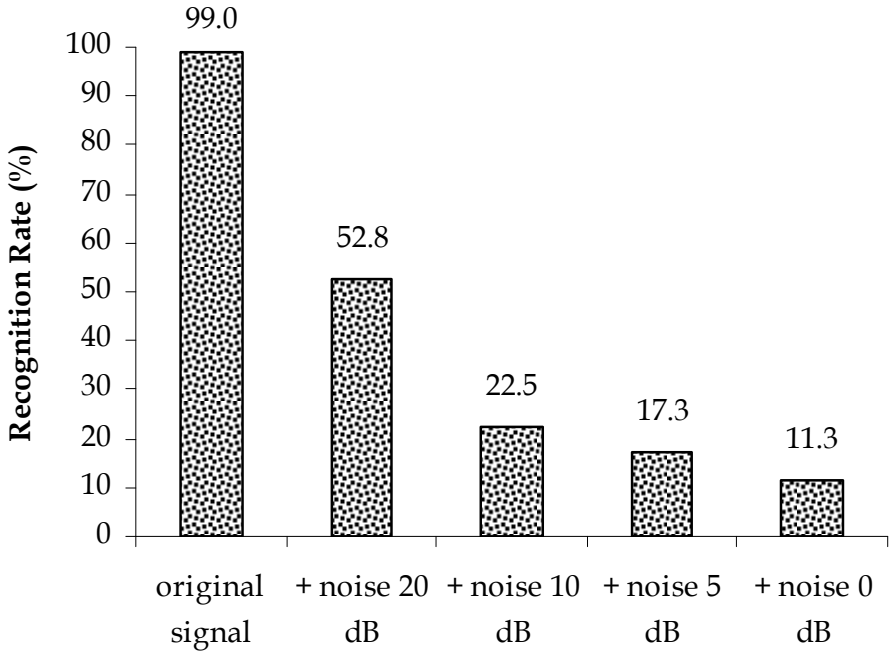


Fig. 11. The accuracy of the system for a variety of noise on the proportion of training data and test data 60:20

To improve the accuracy of the system, then we continue the experiment with Data set 3 (with noise removal or cancellation process, NC) and the results are presented in Figure 12. After going through the process of adaptive noise canceling, system performance increases, especially for signals with the addition of noise, as shown in the figure. For the original signal without adding noise, the NC system provides 96.6% accuracy, about 3% below the system without going through the NC. While for signals with the addition of noise, adaptive noise canceling improve system robustness against noise up to the level of 20 dB with an accuracy of 77.1%. For larger noise, the system failed to work properly.

Based on the above findings, we conducted further experiments using the bispectrum as input for the feature extraction stage. By using this bispectrum, it is expected effect of noise can be suppressed. Bispectrum for a given frame is a matrix with dimensions NxN, where N is the sampling frequency. In this research, we chose N=128, so that for one frame (40 ms) will be converted into a matrix of dimension 128x128. Therefore we perform dimension reduction using quantization techniques. This quantization results next through the process of wrapping and cosine transformation as done in the 1D-MFCC. To abbreviate, then we call this technique as 2D-MFCC.

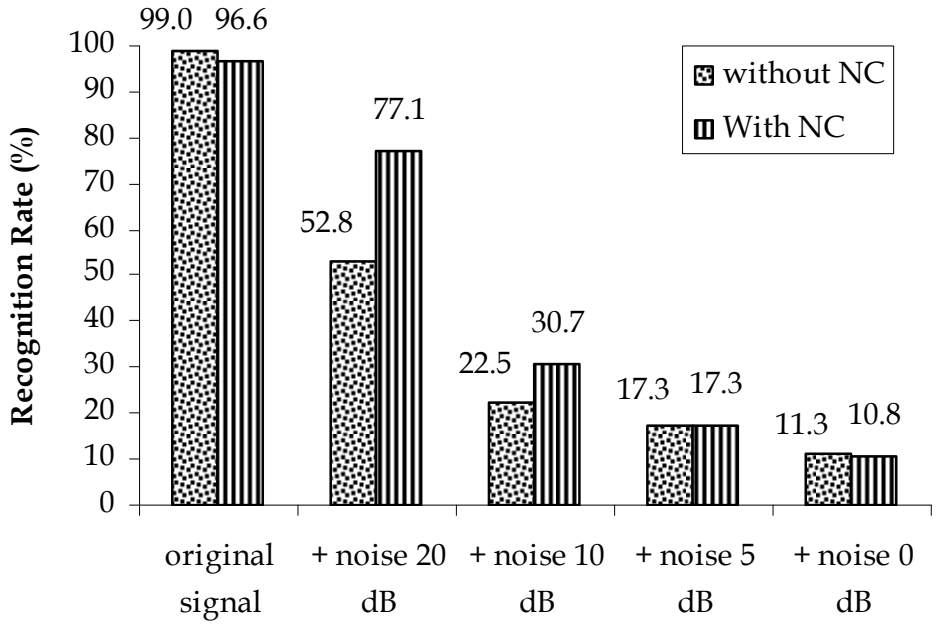


Fig. 12. Accuracy of the system with and without noise cancellation (NC)

4.3 2D-MFCC + HMM

Flow diagram of the experiments conducted in this section are presented in Figure 13. In general there are three parts of the picture, namely the establishment of the channel center (which would be required for quantitation of the bispectrum), the training of HMM models, and the testing model. The process of determining the center of the channel that carried out the research followed the procedure as described in (Fanany & Kusumoputro, 1998). In the training stage of HMM models, each voice signal in the training set is read frame by frame, is calculated its bispectrum values, quantized, and the process of wrapping and cosine transform, so that the feature is obtained. After the feature is obtained, then forwarded to the stage of parameter estimation of HMM with Baum-Welch algorithm. This is done for each speaker, thus obtained 10 HMM models. In testing or recognition phase, a voice signal is read frame by frame, then for each frame is calculated its bispectrum, quantized, followed by wrapping and cosine transform. After that, followed by the recognition process using a forward algorithm for each HMM model (which resulted in the training phase).

Channel center reconstruction

Due to the bispectrum is simetric, then we simply read it in the triangle area of the domain space bispectrum (two-dimensional space, F1xF2). Center channel is determined such that the point (f1, f2) with high bispektrum will likely selected as determination of the channel center. Therefore, the center will gather at the regional channels (f1, f2) with large bispectrum values and for regions with small bispectrum value will have less of channel center. With these ideas, then the center channel is determined by the sampling of points on F1xF2 domain. Sampling is done by taking an arbitrary point on the domain, then at that point generated the random number r∈[0,1]. If this random number is smaller than the ratio

of the bispectrum at these points with the maximum of the bispectrum, then the point will be selected as the determination point. For another thing, then the point is ignored. Having obtained a number of determination points, followed by clustering of these points to obtain the K cluster centers. Then, the cluster center as the channel center on the bispectrum quantization process. From the above explanation, there are three phases to form a center channel, namely the establishment of a joint bispectrum, bispectrum domain sampling and determination of the channel center.

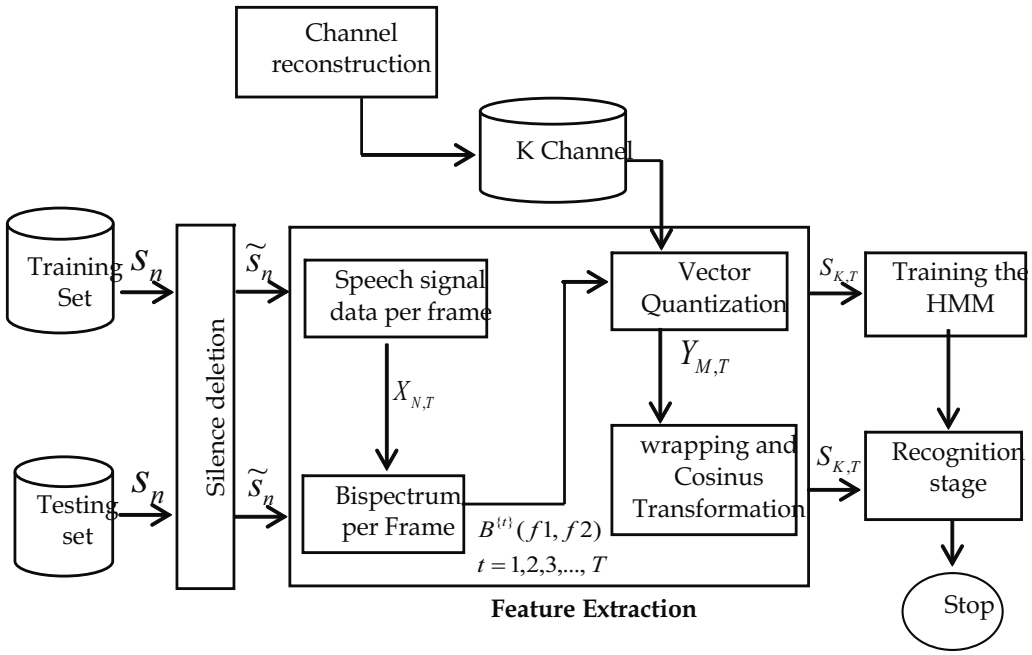


Fig. 13. Flow diagram of the experiments

Figure 14 presents the process of determining the combined bispactrum. a voice signal for each speaker is calculated its bispectrum frame by frame, and then averaged. After this process is done for all speakers, then the combined bispectrum is the sum of the average bispectrum of each speaker divided by the number of speakers (in this case 10).

After obtaining the combined bispectrum, the next is to conduct sampling of the points on the bispectrum domain. Figure 15 presents the sampling process in detail. The first time raised a point A (r1, r2) in the bispectrum domain and determined the point B (f1, f2) which is closest to A. Then calculated the ratio (r) between the combined bispectrum value at point B with the largest combined bispectrum value. After it was raised again a number r3. If r3<r, then inserted the point A into the set of point determination, G. If the number of points on the G already enough, followed by classifying the points on G into P clusters. Cluster centers are formed as the channel center. Next, the P channel's centers is stored for use in a quantization process of the bispectrum (in this research, the P value is 250, 400 and 600).

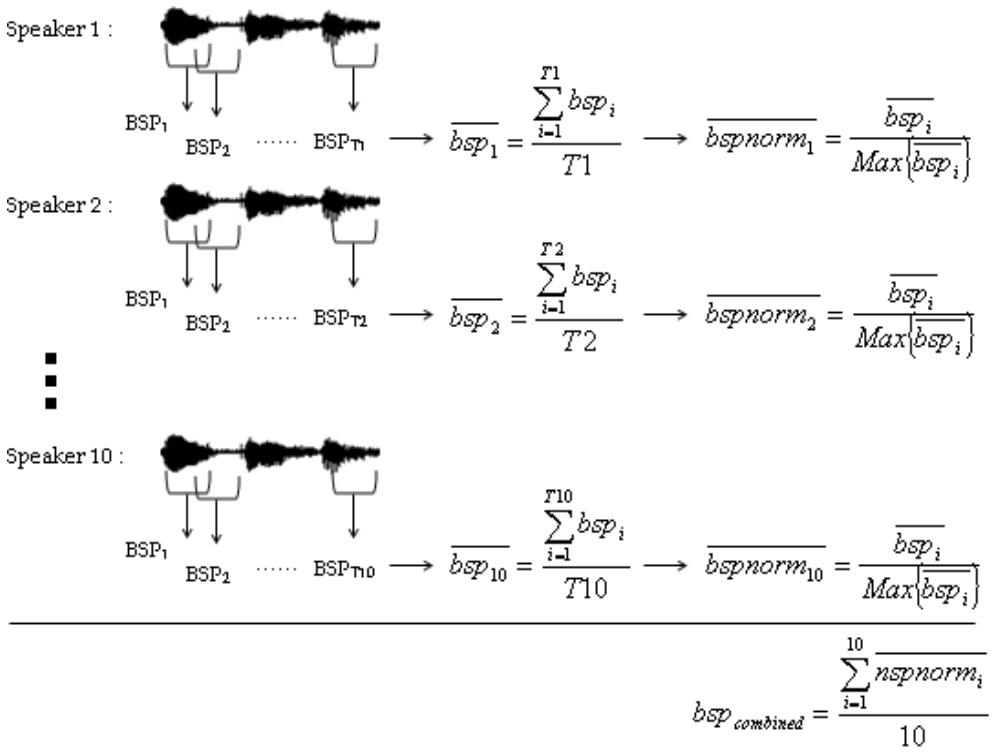


Fig. 14. The process of determining the combined bispectrum

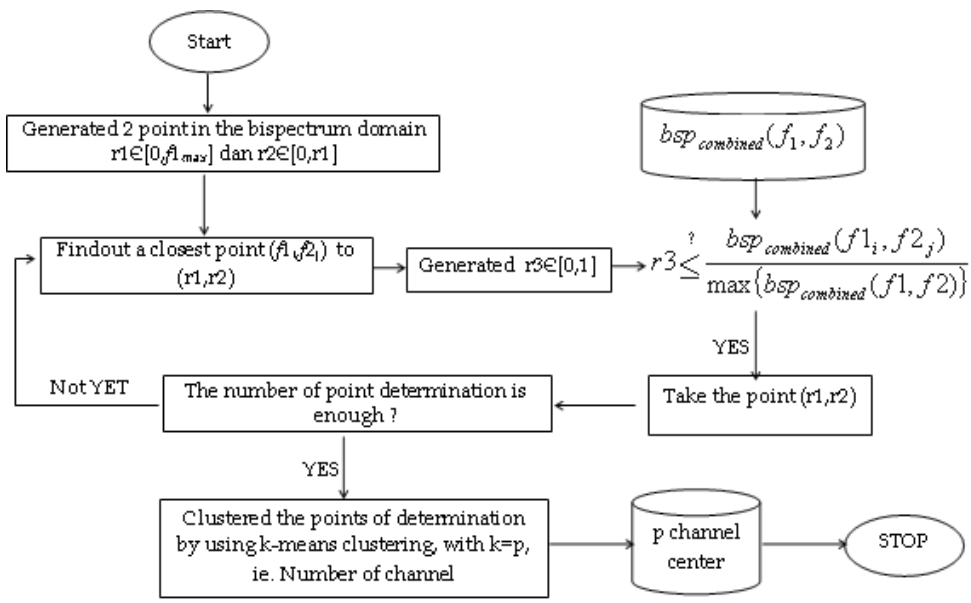


Fig. 15. Bispectrum domain sampling process

Having obtained the P channel's centers, next will be described the process of quantization the bisepctrum of a frame. Bispectrum is read only performed on half of the domain. Each point in the first half of this domain is labeled in accordance with the nearest channel center. Bispectrum values for each channel is obtained by calculating the bispectrum statistic.
The next stage of feature extraction is the process of wrapping. For this, the P channel are sorted based on the distance to the central axis. Wrapping process using a filter like that used in 1D-MFCC. Having obtained the coefficient for each filter, followed by a cosine transform. Output of the feature extraction process is then entered to the recognition stage.

Result and discussion

Figure 16. presents a comparison of the accuracy of the system using the number of channels 250, 400 and 600, followed by wrapping and cosine transform for the reduction of channel dimensions. From the figure, it seen that the 2D-MFCC as feature extraction system provides the average accuracy of 90%, 89%, 75% for the original signal, the original signal plus noise 20 dB and the original signal plus noise 10 dB. With level of noise 5 dB and 0 dB, the system has failed to recognize properly. From these images can also be seen that the number of channels did not provide significant differences effect.

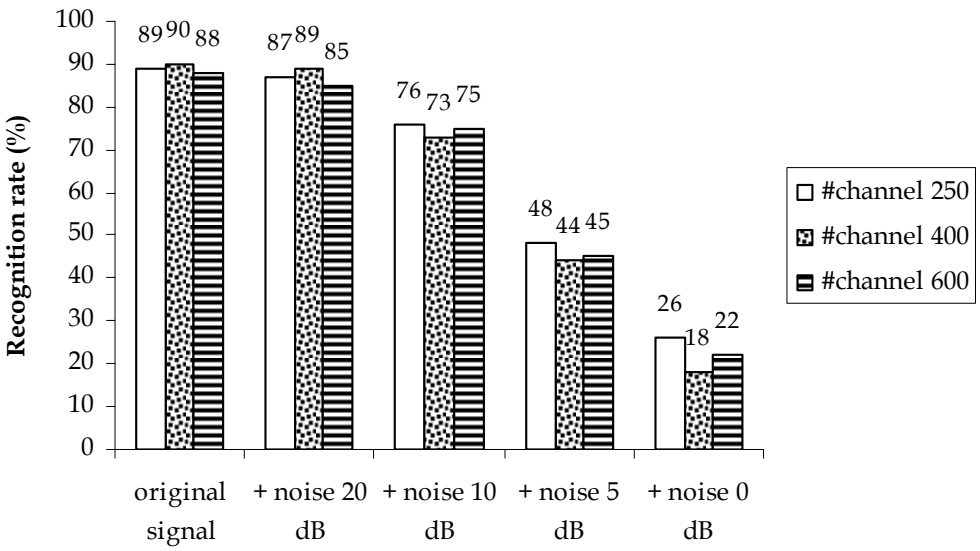


Fig. 16. Comparison of accuracy with different number of channels

When compared with previous techniques based on power spectrum (1D-MFCC) shows that the bispectrum-based technique is more robust to noise. This is as shown in Figure 17. Even if compared with the 1D-MFCC with the elimination of any noise, the 2D-MFCC technique still gives much better results. However, for the original signal, seen this technique still needs improvement. Some parts that can be developed is in the process of wrapping of the bispectrum which is quantized. In this case, there are several options, including whether to continue using the one-dimensional filter (as in the 1D-MFCC) with modifications on the shape and width filter. Or, by developing two-dimensional filter.

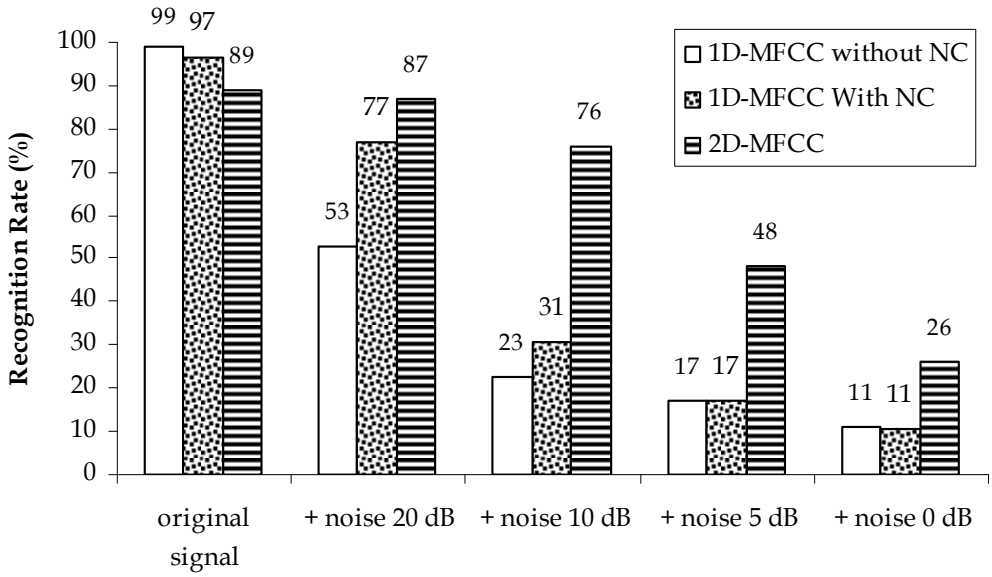


Fig. 17. Comparison of recognition rate between the 1D-MFCC with 2D-MFCC

5. Conclusion and future work

1. Conventional speaker identification system based on power spectrum can give results with an average accuracy of 99% for the original signal without adding noise, but failed to signal with the addition of noise, although only at the level of 20 dB. Noise removal technique is only capable of producing a system with sufficient accuracy (77.1%) up to 20 dB noise level. For larger noise, this technique can not work properly.
2. Bispectrum able to capture the characteristics of voice signals without adding noise or with the addition of noise, and visually it still looks up to levels above 0 dB. For noise level 0 dB, the shape of bispectrum has undergone significant changes compared with the one from original signal
3. In 2D-MFCC, the value bispektrum grouped on some channel that is formed by following bispektrum data distribution. Afterwards is the process of wrapping and cosine transformation. This technique is capable of providing accuracy to the original signal, the original signal plus noise 20 dB, 10 dB, 5 dB and 0 dB are respectively 89%, 87%, 76%, 48% and 26%.

From the experiments we have done, seen that the filter that is used for wrapping process contributes significantly to the level of accuracy. Therefore, further research is necessary to experiment using various forms of filters, such as those developed by Slaney (filter has a constant area, so the higher the filter is not fixed, but follow its width), also from the aspect of the number of filters (linear and logarithmic filters). In our research, we are just experimenting with the bispectrum (third order HOS), so we need further experiments using the HOS with higher order.There are Some disadvantages, (Farbod & Teshnehlab, 2005) with Gaussian HMM, especially in its assumptions, ie normality and independently, and constraints due to limited training data. Therefore it needs to do experiments that integrate 2D-MFCC (HOS-based) with the HMM model is not based on the assumption of normality, and do not ignore the fact that there is dependencies between observable variables.

6. References

Buono, A., Jatmiko, W. & Kusumoputro, B. (2008). Development of 2D Mel-Frequency Cepstrum Coefficients Method for Processing Bispectrum Data as Feature Extraction Technique in Speaker Identification System. *Proceeding of the International Conference on Artificial Intelegence and Its Applications* (*ICACIA*), Depok, September 2008

Nikeas, C. L. & Petropulu, A. P. (1993). *Higher Order Spectra Analysis : A Nonlinear Signal Processing Framework*, Prentice-Hall, Inc., 0-13-097619-9, New Jersey

Fanany, M.I. & Kusumoputro, B. (1998). *Bispectrum Pattern Analysis and Quantization to Speaker Identification*, Master Thesis in Computer Science, Faculty of Computer Science, University of Indonesia, Depok, Indonesia

Ganchev, T. D., (2005). *Speaker Recognition*. PhD Dissertation, Wire Communications Laboratory, Department of Computer and Electrical Engineering, University of Patras Greece.

Dugad, R., & Desai, U. B., (1996). *A Tutorial on Hidden Markov Model*. Technical Report, Departement of Electrical Engineering, Indian Institute of Technology, Bombay, 1996.

Rabiner, L., (1989). A Tutorial on Hidden Markov Model and Selected Applications in Speech Recognition. *Proceeding IEEE*, Vol 77 No. 2., pp. 257-286, 0018-9219, , Pebruari 1989

Boll, S. F., (1979). Suppression of Acoustic Noise in Speech Using Spectral Substraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-27, No. 2, April 1979, pp. 113-120, 0096-3518

Widrow, B. et. al., (1975). Adaptive Noise Canceling : Principles and Applications. *Proceeding of the IEEE*, Vol. 63. No. 12. pp. 1691-1716

Nilsson, M & Ejnarsson, M., (2002). *Speech Recognition using Hidden Markov Model : Performance Evaluation in Noisy Environment*. Master Thesis, Departement of Telecommunications and Signal Processing, Blekinge Institute of Technology

Reynolds, D., (2002). *Automatic Speaker Recognition Acoustics and Beyond*. Tutorial note, MIT Lincoln Laboratory, 2002

Farbod H. & M. Teshnehlab. (2005). Phoneme Classification and Phonetic Transcription Using a New Fuzzy Hidden Markov Model. *WSEAS Transactions on Computers*. Issue 6, Vol. 4.

Part 4

Improvements in the Transportation Industry

12

Ship's Hydroacoustics Signatures Classification Using Neural Networks

Andrzej Żak
Polish Naval Academy
Poland

1. Introduction

Classification is a procedure in which individual items are placed into groups based on quantitative information on one or more characteristics inherent in the items (referred to as traits, variables, characters, etc) and based on a training set of previously labelled items [Stąpor, 2005], [Zak, 2008].

Formally, the problem can be stated as follows: given training data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ produce a classifier $h : X \to Y$ which maps an object $x \in X$ to its classification label $y \in Y$. Classification algorithms are very often used in pattern recognition systems [Szczepaniak, 2004].

While there are many methods for classification, they are solving one of three related mathematical problems. The first is to find a map of a feature space (which is typically a multi-dimensional vector space) to a set of labels. This is equivalent to partitioning the feature space into regions, then assigning a label to each region. Such algorithms (e.g., the nearest neighbor algorithm) typically do not yield confidence or class probabilities, unless post-processing is applied. Another set of algorithms to solve this problem first apply unsupervised clustering to the feature space, then attempt to label each of the clusters or regions [Zak, 2008].

The second problem is to consider classification as an estimation problem, where the goal is to estimate a function of the form:

$$P(class|\vec{x}) = f(\vec{x};\vec{\theta}) \tag{1}$$

where: $\vec{x}$ is the feature vector input; $f(\cdot)$ is the function typically parameterized by some parameters $\vec{\theta}$.

In the Bayesian approach to this problem, instead of choosing a single parameter vector $\vec{\theta}$, the result is integrated over all possible thetas, with the thetas weighted by how likely they are given the training data D:

$$P(class|\vec{x}) = \int f(\vec{x};\vec{\theta}) P(\vec{\theta}|D) d\vec{\theta} \tag{2}$$

The third problem is related to the second, but the problem is to estimate the class-conditional probabilities $P(\vec{x}|class)$ and then use Bayes' rule to produce the class probability as in the second problem.

The most widely used classifiers are the Neural Network (Multi-layer Perceptron, Self Organizing Maps), Support Vector Machines, k-Nearest Neighbours, Gaussian Mixture Model, Gaussian, Naive Bayes, Decision Tree and RBF classifiers.
In this paper the hydroacoustics signals classification is understood as the process of automatically recognition what kind of object is generating acoustics signals on the basis of individual information included in generated sounds. Hydroacoustics signal classification is a difficult task and it is still an active research area. Automatic signal classification works based on the premise that sounds emitted by object to the environment are unique for that object. However this task has been challenged by the highly variant of input signals. The ship own noise is combined with technical environmental noise coming from remote shipping, ship-building industry ashore or port works. There exists also the noise of natural origin: waves, winds or rainfalls. Additional obstruction in the process of spectral component identification can be the fact that various ship's equipment may be the source of hydroacoustical waves of similar or same frequencies. The propeller is the dominant source of the hydroacoustical waves at higher vessel speeds. It generates the driving force that is balanced by the resistance force of the hull. It also stimulates the vibrations of the hull's plating and all elements mounted on it. It should be noticed that, sounds signals in training and testing sessions can be greatly different due to above mentioned facts and because of object sounds change with time, efficiency conditions (e.g. some elements of machinery are damaged), sound rates, etc. There are also other factors that present a challenge to signal classification technology. Examples of these are variations of environment conditions such as depth and kind of bottom of area were measured take place, the water parameters such as salinity, temperature and presence of organic and non organic pollutions.
Acoustic signatures have the great significance because its range of propagation is the widest of all physics field of ship. Controlling and classification of acoustic signature of vessels is now a major consideration for researchers, naval architects and operators. The advent of new generations of acoustic intelligence torpedoes and depth mines has forced to a great effort, which is devoted to classify objects using signatures generated by surface ships and submarines. It has been done in order to increase the battle possibility of submarine armament. Its main objectives are to recognize the ship and only attack this one which belongs to opponent.

2. Ship's hydroacoustics signatures

2.1 Transmission of acoustic energy

People, who has spent time aboard a ship known that vibration and related with them noise is a major problem there. First off all it should be proved that underwater radiated noise has its origin in vibration of ships mechanism [Gloza & Malinowski, 2001]. This can be done by simultaneous measurements of underwater noise and vibrations and then comparison of results using coherence function. Such result are gain over during research on stationary hydroacoustic range where a measured vessel is anchored between buoys which determine the area of range (see figure 1). In this form of measurements the array of hydrophones is positioned one meter above the sea bottom and under the hall of the ship. Accelerometers are installed inside important rooms of ship (engine and auxiliary rooms) to measure vibration. The points of positioning the accelerometers are such chosen to have adequate measurements of transmission vibration energy into water as sound energy. Mostly this points are the places of foundation of main engines, auxiliary engines or set of current generator.

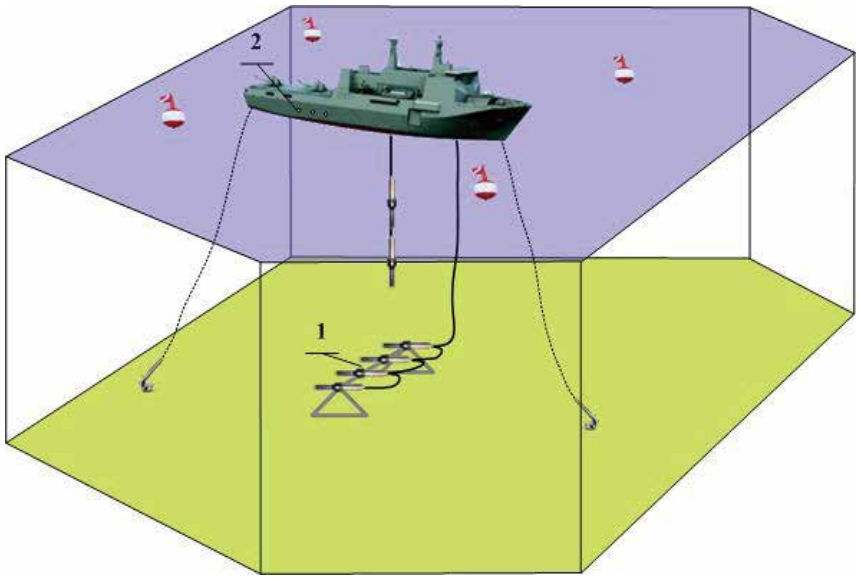


Fig. 1. Schema of hydroacoustics range during measurements using statical method; 1) sensors of acoustic signatures – array of hydrophones, 2) sensors of vibrations – accelerometers

The directional radiation from the vessel is injected into the water medium, where not only the source but also refraction and boundaries influence the acoustic propagation. At long ranges, the low frequency noise originates mainly from a very narrow sector [Gloza & Malinowski, 2002]. The ambient noise due to long – range shipping indicates that shipping noise constitutes a 20 to 30 dB elevation of the ambient noise levels in the low frequencies. What more the level of noise radiated to the sea environment in the all frequencies is increasing due to both the increased number of vessels at the sea and the increased engine power of the modern ships. Ship noise does not transmit acoustic energy uniformly in all directions, but has a characteristic directional pattern in the horizontal plane around the radiating ship as it is shown on figure 2. More noise is radiated in the aft direction, because of the working propellers and because the hull is screening in the forward direction and the wake at the rear.

It have to be determined how much total acoustic power is radiated by a running ship and how it compares with the power used by the vessel for propulsion through the water. This can be done by measuring vibration aboard the ship (inside the engine room) and compare it into the underwater sound. The similarities between the vibration signals of chosen elements within the hull and of the ship and the underwater acoustical pressure in the water are represented by the coherence function. For two signals of pressure $p(t)$ and vibration $v(t)$ the coherence function is described as follow [Gloza & Malinowski, 2001]:

$$\gamma_{pv}^2(f) = \frac{\left|G_{pv}(f)\right|^2}{G_p(f)G_v(f)} \tag{3}$$

where: G_p and G_v denote the corresponding spectral densities of signals $p(t)$, $v(t)$ respectively; G_{pv} denotes the cross spectral density.

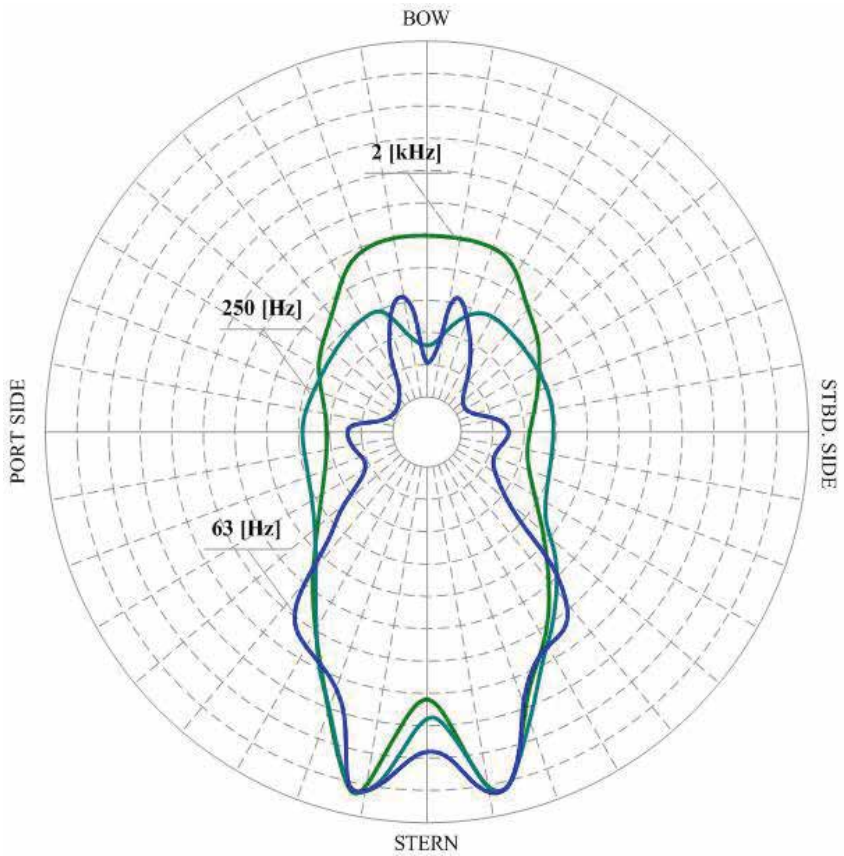


Fig. 2. Equal pressure level contours of noise around a ship

Coherence function is a real function accepting arguments from the range of:

$$O \le \gamma_{pv}^2(f) \le 1 \tag{4}$$

Therefore, the zero value occurs for signals that do not have the cause association and the one value for signals coming from the same source. Using the dependence (3) the coherence function between the signals can be determined. The components in the coherence spectrum determined this way reflect qualitative correlations associated with particular frequencies coming from a working piece of equipment.

Coherence coefficient function is convenient in this kind of research because it allows to determine the similarity between the spectra of particular signals. In the table 1 it can be seen a series of discrete components for which the coherence values are maximum that means from 0.8 to 1.

The interpretation of the underwater noise of a vessel was conducted by analyzing the spectra of consecutively powered up machines and comparing them with the corresponding underwater noise. In the first phase the measurements of vibration velocities and aggregate noise (primary engines not working) were carried out. Then, the measurements were continued for the left, right and both main engines.

Frequency [Hz]	Coherency function	Vibration on the hull [μm/s]
16.5	0.8	13
25	1	80
37.5	0.8	69
50	1	42
62.5	0.9	8.4
75	1	72
87.5	1	64
100	0.8	23
112.5	1	55
125	1	28
150	1	66
162.5	1	35
175	0.7	69
200	0.9	19

Table 1. Vibration and coherence function of hydroacoustics pressure and vibration

The comparison of vibrations velocities registered at the ship's hull and at the fundament of the power generators with the underwater noise were presented in table 2. Analogically, the research was conducted for the ship's main engine. The results of narrow-band spectral levels and the coherence function were shown on figure 3.

Vibration	Frequency		Harmonics
	Formula	[Hz]	
Unbalanced parts	$f_n = kf_0$	25	50, 75, 100, 125, 150, 175, 200, …
Diesel firing rate	$f_s = \frac{kz_c f_o s}{4}$	12.5	25, 37.5, 50, 62.5, 75, 87.5, 100,112.5, 125, 137.5, 150, …

Table 2. Basic frequencies and harmonics of vibration

where: $k = 1, 2, \ldots$ is the number of next harmonics; f_0 is the main frequency; s is the coefficient of stroke (equal 0.5 for four stroke engines); z_c is the number of cylinder;

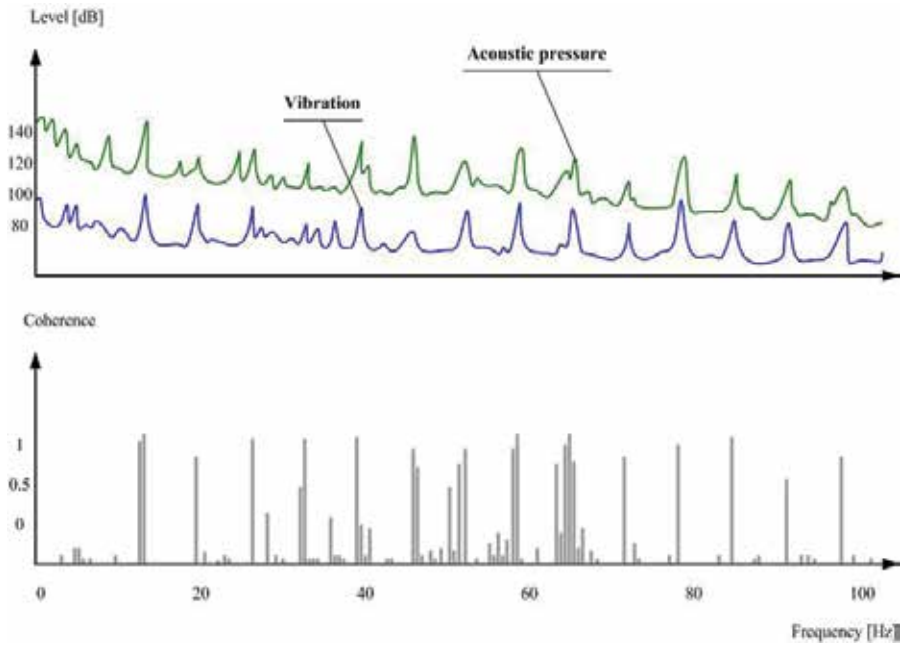


Fig. 3. Narrow-band spectra and coherence function of underwater acoustic pressure and vibration of a stationary ship

Relations between mechanical vibration and hydroacoustics field of a ship is presented by transmission coefficient of the mechanical vibration α :

$$\alpha = \frac{L_{1m,1Hz}}{\rho\, cv} \tag{5}$$

where: $L_{1m,1Hz}$ is sound pressure level relative to 1μPa at 1 m for 1 Hz; ρ is fluid density for sea water; v is vibration velocity; c is propagation velocity of sound wave.

$$L_{1m,1Hz} = L + 20\log R - 10\log \Delta f \tag{6}$$

where: L is acoustic pressure level under ship (dB re μPa); R is the distance between a ship and a sensor (m); Δf is the width of an applied filter (Hz).

The results of the acoustic levels, vibration speeds and coefficient α are shown in table 3.

f (Hz)	*L (Pa)*	*v (m/s)*	α
12.5	3.14	0.001	$2.2\ 10^{-3}$
25	6.3	0.00032	$1.4\ 10^{-2}$
37.5	14.1	0.00028	$3.4\ 10^{-2}$
75	56.2	0.0005	$7.7\ 10^{-2}$

Table 3. The energy transmission coefficient calculated for consecutive frequencies

The proportionality factor ρc is the acoustic resistance (specific impedance) of the fluid and for sea water is $1.5\ 10^5$ g/cm^2 s.

Though radiated sound is frequently expressed in spectrum levels, that is, in 1 Hz bands (shown in $L_{1m,1Hz}$), frequency analyses are more conveniently made in wider bands so the results are reduced to a band of 1 Hz. The results are reduced to a band of 1 Hz by applying a bandwidth reduction factor equal to 10 log of the bandwidth used. The distance in this case is the horizontal distance, while the actual source-to-receiver range, the radial distance, was used for these measurements. Therefore here should be calculated as 20 log range (spherical) spreading loss applies in the acoustic field at all frequencies.

2.2 Sources of ship noise and its deviations

Several sources of noise radiation from a ship exist. They have the characteristic frequency bands and are mainly dependent on speed. Among the main sources of ship noises are:

- propeller,
- machinery,
- hydrodynamic processes.

The sources of ship underwater sounds are diverse and a given source changes its sound output with ship speed. Therefore ship noises are variable complex and sound components are distributed through the entire frequency range.

The main source is the hull, which transmits the vibrations of the machinery into the water. The propellers also radiates high level of noise because of hydrodynamic streams and cavitations.

Machinery noise originates as mechanical vibrations of many devices inside a moving vessel. They create underwater noise in the following ways:

- rotating unbalanced shafts,
- repetitive discontinuities,
- explosions in cylinders,
- cavitation and turbulence in the fluid flow in pumps, pipes and valves,
- mechanical friction in bearings.

The first three of these sources radiate sounds of a discrete spectrum in which the noise is dominated by tonal components at the basic frequencies and their harmonics [Urick, 1975], [Gloza & Malinowski, 2002].

The harmonic structure of radiated noise is complex, and even a discrete component generated by a single source of noise is irregular and variable. With changing conditions of the ship it can be observed variations of level and frequencies.

There are various paths of sound transmission such as the mounting of the main engine or diesel generator, which connect the vibrating parts to the hull. Radiation at discrete components, caused by low frequency hull vibrations, excited by the machinery is easily detected. In the noise reduction control, it must be reduced as much as possible.

One of the methods of identification of underwater noises generated by moving ship is by investigation of its spectrum. Basing on the conducted analysis it is possible to isolate discrete components in the spectra associated with the work of mechanisms and equipment on board along with the broad band spectrum reflecting the work of the cavitating propeller, turbulent flow in piping and ventilators or bearing frictions.

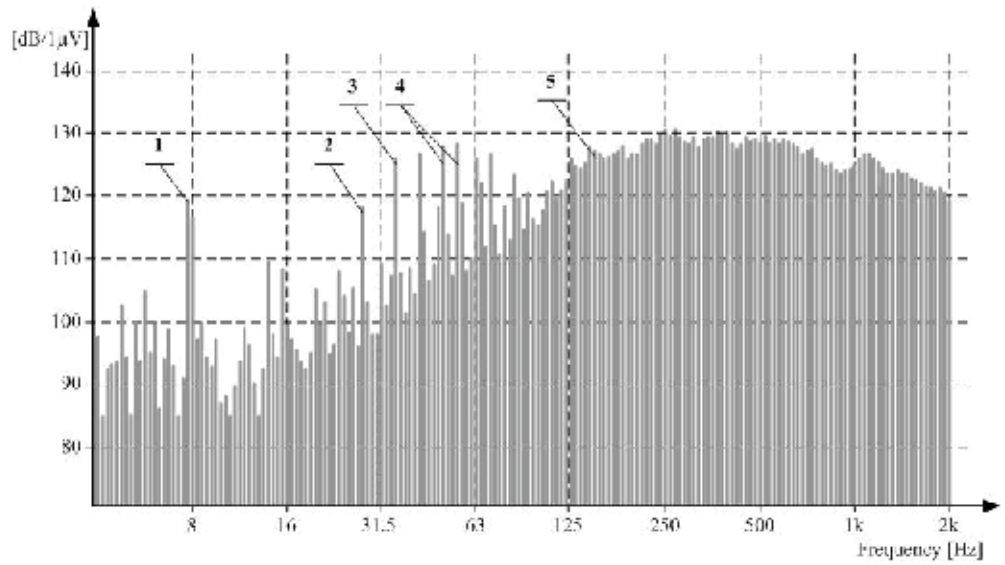


Fig. 4. The underwater noise spectrum or so called "acoustic portrait" of a moving ship; 1) shaft, 2) diesel generator, 3) propeller blades, 4) main engines, 5) propeller

Figure 4 shows a keel aspect narrow-band power spectrum in 0.5 Hz bands of a typical ship going with the speed of 3.8 knots. The radiated noise data show high-level tonal components which are from the ship's service diesel generator, main engine firing rate and blade rate. A ship's service diesel generator creates a series of harmonics which amplitudes and frequencies are independent of ship speed. Propellers generate cavitation especially at high speeds of a vessel (above 8 kn) which creates noise having a continuous spectrum. The cavitation is production and collapse of cavities and bubbles produced by the propeller action. Cavitation noise consists of a large number of random small bursts formed by bubble collapse. As it was mentioned earlier cavitation noise has a continuous spectrum. At the higher speed of the vessel the propeller noise increases and the main energy shifts to lower frequencies [Gloza & Malinowski, 2002].

The sound level spectrum constitutes a mixture of the continuous and discrete lines. The former are characterized by a maximum in the area from 50 to 200 Hz, which is a typical feature in ship noise spectra. At frequencies greater than 200 Hz, sound pressure level (SPL) falls by 6 dB, when the frequency is doubled. It means that SPL is inversely proportional to the square of the frequency. The discrete components are the most visible in a ship's spectra since they are detected even at low speeds (shown on figure 4). Moreover these discrete components of noise spectra are called "acoustic portrait", which is unique for each ship. This acoustic portrait is used to reveal the location and to identify the source of noise.

It can't be forgot that hydroacoustic signatures of ship is mainly generated by phenomena of vibrations of vessel working machinery. Therefore changing the speed of moving ship cause, first of all, the changes in sound volume which is described by sound pressure levels (shown on figure 5) what has the essential influence on the range of sound propagation.

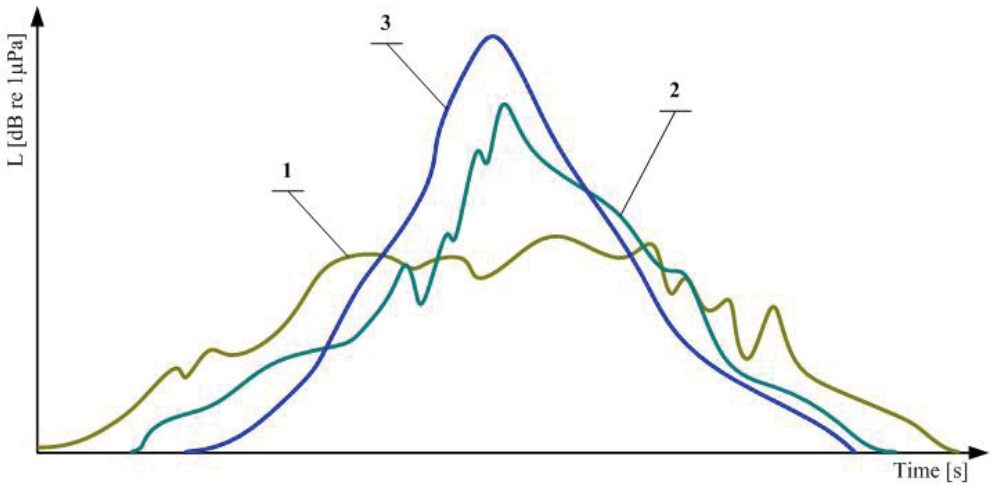


Fig. 5. The sound levels radiated by moving ship with different speeds; 1) 3.8 kn, 2) 8 kn, 3) 11 kn

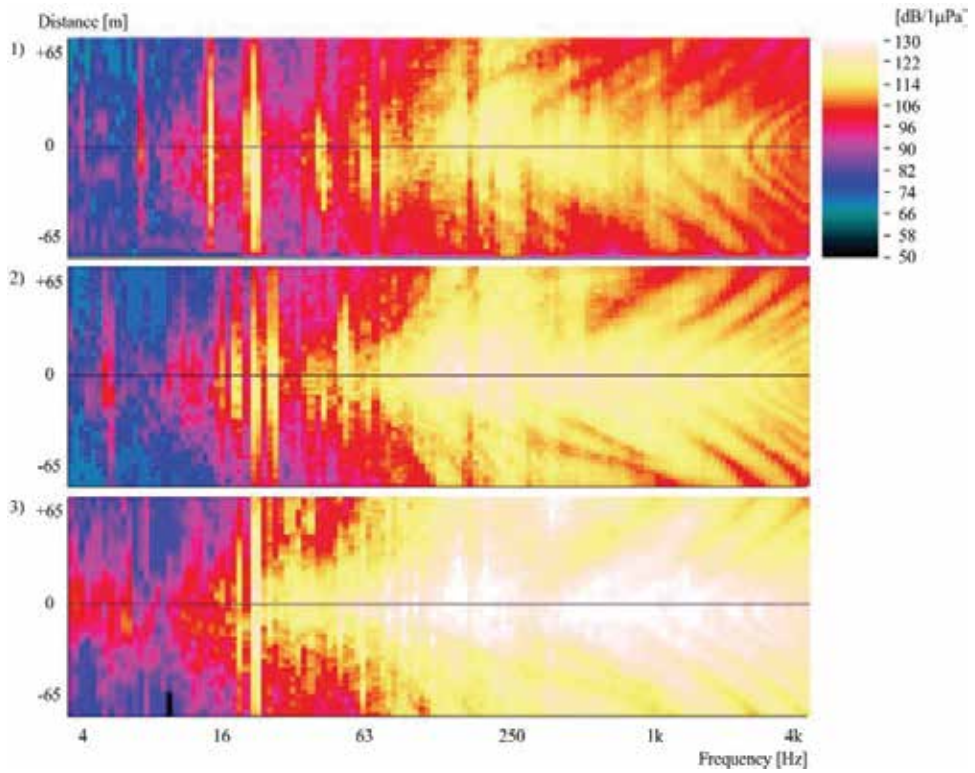


Fig. 6. The spectograms received during ship running over hydrophones with different speeds; 1) 3.8 kn, 2) 8 kn, 3) 11 kn

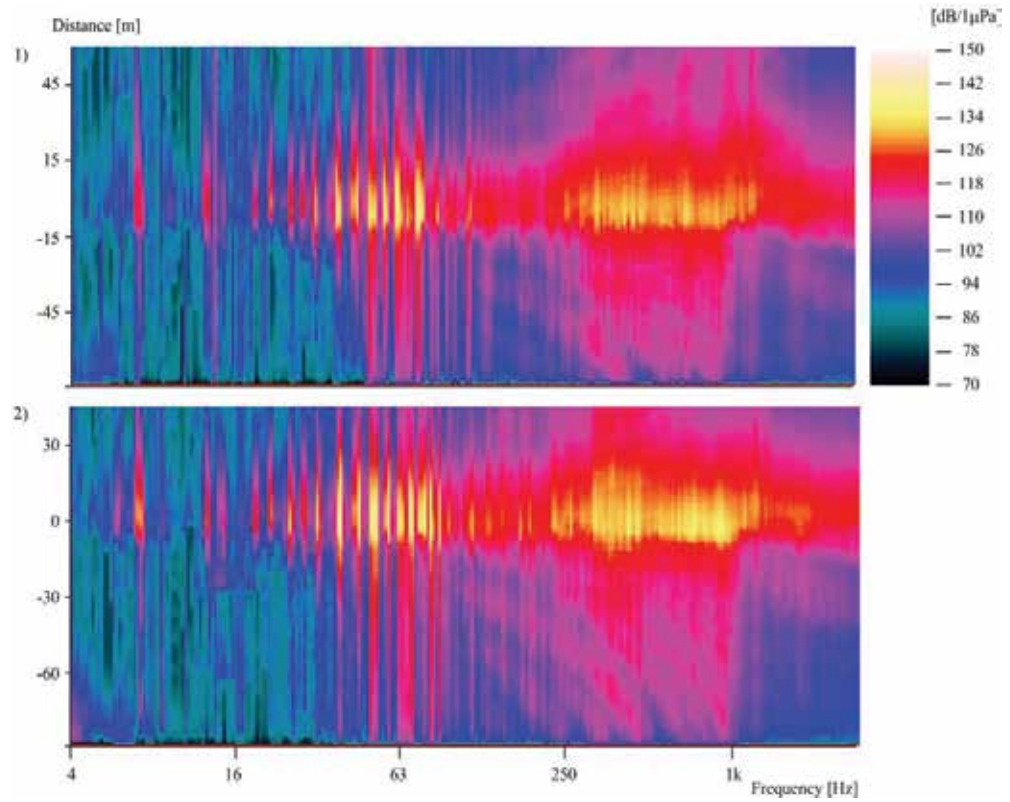


Fig. 7. The spectograms received during ship running over hydrophones in different phase of exploitation; 1) after general renovation, 2) 2 years after general renovation

But not only the sound level radiated by moving ship change with speed but also the distribution on frequency in hydroacoustic signature of ship is changing (shown on figure 6). Hydroacoustic signatures changes also with time (shown on figure 7). After few years of exploitation the conditions of mechanical elements of ship's mechanism aren't the same as after general renovation. Elements like bearings, pistons and other movable elements are using up. So it has influence on vibrations and the same the distribution of frequency in hydroacoustic signatures.

3. Feature extraction

The purpose of signal feature extraction module is to convert the sound waveform to some type of parametric representation for further analysis and processing. This is often referred as the signal-processing front end. A wide range of possibilities exist for parametrically representing the signals for the sound recognition task, such as Linear Prediction Coding (LPC), Mel-Frequency Cepstrum Coefficients (MFCC) [Zak, 2005], and others. Mel-Frequency Cepstrum Coefficients method will be discussed in this paper.
MFCC's are based on the known variation of the human ear's critical bandwidths with frequency, filters spaced linearly at low frequencies and logarithmically at high frequencies have been used to capture the phonetically important characteristics of speech. This is

expressed in the mel-frequency scale, which is a linear frequency spacing below 1000 [Hz] and a logarithmic spacing above 1000 [Hz].

A block diagram of the structure of an MFCC processor is given on figure 8. As been mentioned previously, the main purpose of the MFCC processor is to mimic the behavior of the human ears. In addition, rather than the speech waveforms themselves, MFFC's are shown to be less susceptible to mentioned variations.

First step of MFCC processor is the frame blocking. In this step the continuous sound is blocked into frames of N samples, with adjacent frames being separated by M where $M < N$. The first frame consists of the first N samples. The second frame begins M samples after the first frame, and overlaps it by $N - M$ samples. Similarly, the next frames are created so this process continues until all the sound is accounted for within one or more frames.

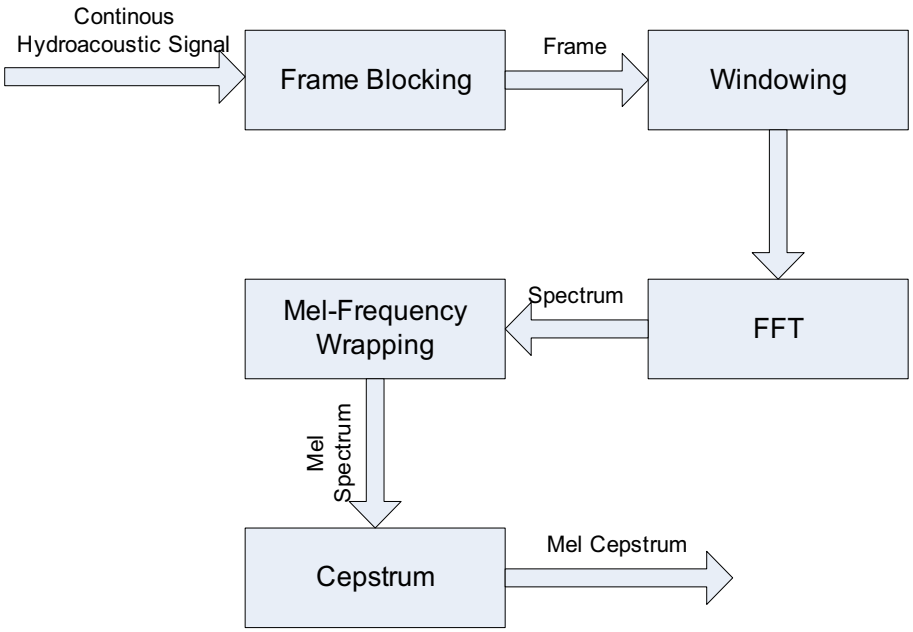


Fig. 8. Block diagram of the MFCC processor

The next step in the processing is to window each individual frame so as to minimize the signal discontinuities at the beginning and end of each frame. The concept here is to minimize the spectral distortion by using the window to taper the signal to zero at the beginning and end of each frame. If we define the window as: $w(n), \quad 0 \le n \le N-1$, where N is the number of samples in each frame, then the result of windowing is the signal:

$$y(n) = x(n)w(n), \quad 0 \le n \le N-1 \tag{7}$$

Typically the Hamming window is used, which has the form [Therrien, 1992]:

$$w(n) = 0.54 - 0.46\cos(\frac{2\pi n}{N-1}), \quad 0 \le n \le N-1 \tag{8}$$

The next processing step is the Fast Fourier Transform, which converts each frame of N samples from the time domain into the frequency domain. The FFT is a fast algorithm to implement the Discrete Fourier Transform (DFT) which is defined on the set of N samples, as follow [Therrien, 1992]:

$$X_n = \sum_{k=0}^{N-1} x_k e^{-2\pi jkn/N}, \quad n = 0,1,2,\ldots,N-1 \tag{9}$$

Next step in MFCC processor is the Mel-frequency Wrapping. As mentioned above, psychophysical studies have shown that human perception of the frequency contents of sounds for speech signals does not follow a linear scale. Thus for each tone with an actual frequency f, measured in [Hz], a subjective pitch is measured on a scale called the "mel" scale. The mel-frequency scale is a linear frequency spacing below 1000 [Hz] and a logarithmic spacing above 1000 [Hz]. As a reference point, the pitch of a 1 [kHz] tone, 40 [dB] above the perceptual hearing threshold, is defined as 1000 mels. Therefore we can use the following approximate formula to compute the mels for a given frequency f in [Hz]:

$$mel(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \tag{10}$$

One approach to simulating the subjective spectrum is to use a filter bank, spaced uniformly on the mel scale (figure 9). That filter bank has a triangular bandpass frequency response, and the spacing as well as the bandwidth is determined by a constant mel frequency interval.

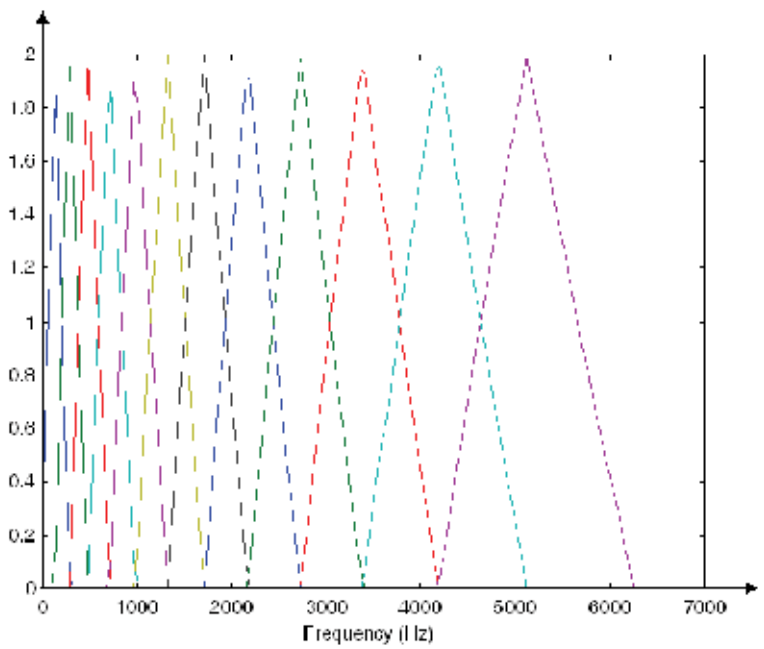


Fig. 9. An example of mel-spaced filterbank

In this final step, we convert the logarithmic mel spectrum back to time. The result is called the mel frequency cepstrum coefficients (MFCC). The cepstral representation of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame analysis. Because the mel spectrum coefficients, and so their logarithm, are real numbers, we can convert them to the time domain using the Discrete Cosine Transform (DCT). Therefore if we denote those mel power spectrum coefficients that are the result of the last step are S_k, $k = 1,2,\ldots,K$, we can calculate the MFCC's, as [Zak, 2005]:

$$c_n = \sum_{k=1}^{K} \log(S_k) \cos\left(\frac{n(k-0.5)\pi}{K}\right), \quad n = 1,2,\ldots,K \tag{11}$$

Note that we exclude the first component, c_0 from the DCT since it represents the mean value of the input signal which carried little speaker specific information.

4. Classification process

4.1 Self organizing maps

In literature there is no description of method of classification hydracoustic signatures. It is caused because very narrow group of scientists are interesting in this kind of problem. Most of these scientists are related with military scientific center because this problem from military point of view is very important, so their research works are mostly confidential. Therefore as method of classification of hydroacoustic signatures are used mostly general methods of classification like minimal-distance classifier, feature correlation, decision tree, Bayesian method or radial basis function classifiers. Another group establish methods such as hidden Markov's model where classification is bring to problem of determine the model of signal.

Because of similarity of hydroacoustic to acoustic there exists some basis to use methods of speech recognition as method of hydroacoustic signature's classification. To solve problems of speech recognition or widely acoustic signal recognition with successful are used linear predictive coding method or artificial neural networks.

Kohonen neural network, also known as The Self-Organizing Map (SOM) is a computational method for the visualization and analysis of high-dimensional data, especially experimentally acquired information [Fort, 2006], [Haykin, 1999].

One of the most interesting aspects of SOMs is that they learn to classify data without supervision. With this approach an input vector is presented to the network and the output is compared with the target vector. If they differ, the weights of the network are altered slightly to reduce the error in the output. This is repeated many times and with many sets of vector pairs until the network gives the desired output. Training a SOM however, requires no target vector.

For the purposes of this paper the two dimensional SOM will be discussed. The network is created from a 2D lattice of 'nodes', each of which is fully connected to the input layer. Figure 10 shows a very small Kohonen network of 4×4 nodes connected to the input layer (shown as rectangle) representing a two dimensional vector.

SOM does not need a target output to be specified unlike many other types of network. Instead, where the node weights match the input vector, that area of the lattice is selectively optimized to more closely resemble the data for the class the input vector is a member of.

From an initial distribution of random weights, and over many iterations, the SOM eventually settles into a map of stable zones. Each zone is effectively a feature classifier, so the graphical output can be treated as a type of feature map of the input space.
Training occurs in several steps and over many iterations [Kohonen, 2001]:

- Each node's weights are initialized.
- A vector is chosen at random from the set of training data and presented to the lattice.
- Every node is examined to calculate which one's weights are most like the input vector. The winning node is commonly known as the Best Matching Unit (BMU).
- The radius of the neighborhood of the BMU is now calculated. This is a value that starts large, typically set to the 'radius' of the lattice, but diminishes each time-step. Any nodes found within this radius are deemed to be inside the BMU's neighborhood.
- Each neighboring node's (the nodes found in step 4) weights are adjusted to make them more like the input vector. The closer a node is to the BMU, the more its weights get altered.
- Repeat step 2 for N iterations.

To determine the best matching unit, one method is to iterate through all the nodes and calculate the distance between each node's weight vector and the current input vector. The node with a weight vector closest to the input vector is tagged as the BMU.

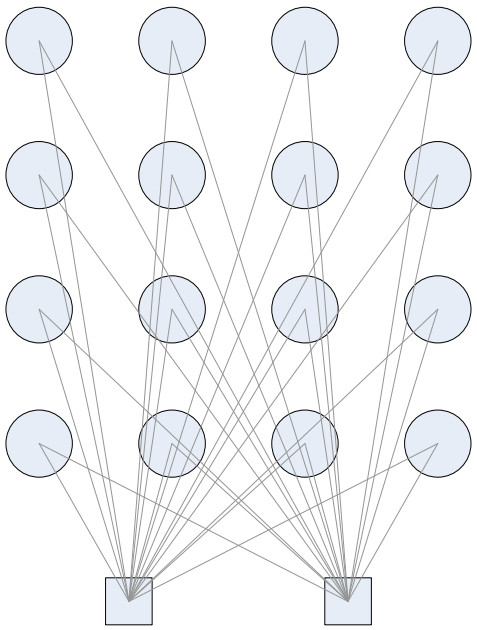


Fig. 10. A simple Kohonen network

There are many methods to determine the distance [Kohonen, 2001] but for our purpose we will use the most popular Euclidean distance which is given by:

$$d(x, w_i) = \|x - w_i\| = \sqrt{\sum_{j=0}^{N} (x_j - w_{ij})^2} \tag{12}$$

where: x is the current input vector; w is the node's weight vector.

Each iteration, after the BMU has been determined, the next step is to calculate which of the other nodes are within the BMU's neighborhood. All these nodes will have their weight vectors altered in the next step.
A unique feature of the Kohonen learning algorithm is that the area of the neighborhood shrinks over time. This is accomplished by making the radius of the neighborhood shrink over time.
To do this the exponential decay function can be used as follow:

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\lambda}\right) \quad t = 0,1,2,\ldots \tag{13}$$

where: σ_0 denotes the width of the lattice at time t_0; λ denotes a time constant; t is the current time-step (iteration of the loop).
Every node within the BMU's neighborhood (including the BMU) has its weight vector adjusted according to the following equation:

$$w_{ij}(t+1) = w_{ij}(t) + \theta(t)\eta(t)(x_j(t) - w_{ij}(t)) \tag{14}$$

where: t represents the time-step; η is a small variable called the learning rate, which decreases with time.
The decay of the learning rate is calculated each iteration using the following equation:

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\lambda}\right) \quad t = 0,1,2,\ldots \tag{15}$$

In equation (14), not only does the learning rate have to decay over time, but also, the effect of learning should be proportional to the distance a node is from the BMU. Indeed, at the edges of the BMUs neighborhood, the learning process should have barely any effect at all. Ideally, the amount of learning should fade over distance similar to the Gaussian decay according to the formula:

$$\theta(t) = \exp\left(-\frac{dist}{2\sigma^2(t)}\right) \quad t = 0,1,2,\ldots \tag{16}$$

where: $dist$ is the distance a node is from the BMU; σ is the width of the neighborhood function as calculated by equation (13).
Another method of learning Kohonen's neural networks is learning with strain. The learning with strain is special modification of concurrent learning. This learning method allows to use Kohonen's network in cases when the vectors of desired output signals of neural networks z_j are known. This learning method has the character of straining the correct answers of network despite of what network want to do.
This method needn't to calculate the values of errors made by neural network as it has place in classic feedforward networks, what makes possible to speed up the learning process. The following methods of learning with strain can be pointed [Fort, 2006]:

- method of autoassociation:

$$w_{ij}(t+1) = w_{ij}(t) + \Theta(t)\eta(t)(x_j(t)z_j(t)) \tag{17}$$

- method of incremental autoassociation:

$$w_{ij}(t+1) = w_{ij}(t) + \Theta(t)\eta(t)\cdot(x_j(t) - x_j(t-1))(z_j(t) - z_j(t-1)) \tag{18}$$

- method of bringing nearer the weight's vector to the desired output vector:

$$w_{ij}(t+1) = w_{ij}(t) + \Theta(t)\eta(t)(z_j(t) - w_{ij}(t)) \tag{19}$$

Each time the choice of presented above method must be done basing on usefulness in concrete task. It must be noticed that because of lack of general theory in this case there are necessary the experiments and research leaning on empirical investigations.

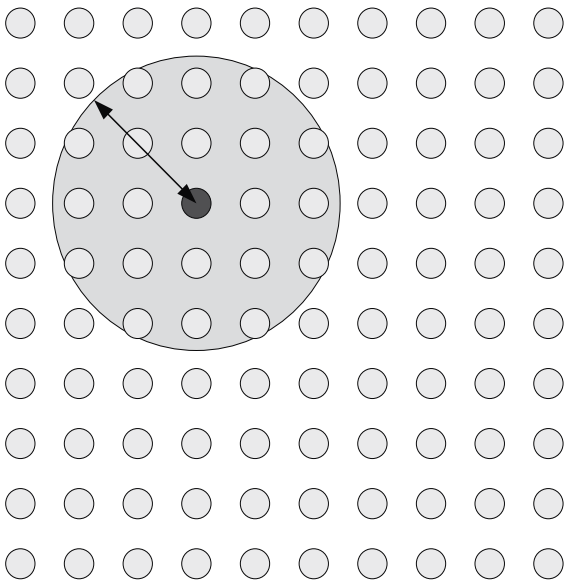


Fig. 11. The BMU's neighborhood

4.2 End-user classification

We must notice that using Kohonen's neural networks in classification as a results gives maps of membership in which each activated region is connected with particular ship. But basing of these results we can't say anything about type of ship because we don't know in which region of membership maps will be particular ship associated. Of course after many cycles of calculations person which is supervising classification will be able to say which type of ship is connected to given region of maps. But on introduction we assume that classification will be made automatically so we use feedforward neural network to connect activated regions in membership maps acquired form Kohonen's network with particular name of type of ship. Let's see how to do this.

Feedforward neural networks are the most popular and most widely used models in many practical applications. Neural networks consist of artificial neurons which are the systems

with many inputs and one output. Each neuron performs a weighted summation of the inputs, which then passes a nonlinear activation function, also called the neuron function. Mathematically the functionality of a neuron is described by [Osowski, 1996]:

$$y_i = f\left(\sum_{j=1}^{N} w_{ij} x_j\right) \tag{20}$$

where: y is the output of i-th neuron, w denotes the weight vector, x is the input vector, $f()$ denotes activation function.

The activation functions can be any differential function. Most common is used standard sigmoid function. The variables w for each neuron are the parameters of the network model that can be represented collectively for whole neural network by the parameter vector Θ.

The network is divided into layers. The input layer consists of just the inputs to the network. Then follows a hidden layer or layers, which consists of any number of neurons. The network output is formed by the output layer. Generally, the number of inputs depend on length of input vector and number of neurons in output layer equals the number of outputs of the approximation problem. During creating architecture of feedforward neural network the problem is to determine numbers of hidden layers and number of neurons in each hidden layer. This problem can be solved using Vapnik-Chervonenkins rules [Osowski, 1996].

Given a fully specified network, it can be trained using a set of data containing N input-output pairs (x, z) where x denotes input vector and z desired values of output of neural network. With this data the mean square error (between calculated output of neural network and desired values) is defined by [Osowski, 1996]:

$$E(\Theta) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i(k) - z_i(k)\right)^2 \tag{21}$$

Then, a good estimate for the parameter is one that minimizes the MSE that is:

$$\hat{\Theta} = \arg\min_{\Theta} E(\Theta) \tag{22}$$

The various training algorithms that apply to feedforward networks have one thing in common—they are iterative. They start with an initial parameter vector Θ_0, which is generated using random function. Starting at Θ_0, the training algorithm iteratively decreases the MSE by incrementally updating along the negative gradient of the MSE, as follows [Osowski, 1996]:

$$\Theta(k+1) = \Theta(k) - \eta R \nabla_{\Theta} E(\Theta) \tag{23}$$

where: the matrix R may change the search direction from the negative gradient direction to a more favorable one, η is the learning rate.

The purpose of parameter η is to control the size of the update increment in Θ with each iteration i, while decreasing the value of the MSE.

There are few algorithms of training neural networks for example the most popular are: Backpropagation, Levenberg-Marquardt, Gauss-Newton, Steepest-descent. The basis method is backpropagation algorithm which is similar to the steepest descent algorithm

with the difference that the step length η is kept fixed during the training. Hence the backpropagation algorithm is obtained by choosing $R = I$ in the parameter update in (23). The MSE calculated for output layer if propagated back from the output layer through hidden layers to the input layer, and become basis to determine changes of neural network parameters Θ. The training algorithm may be augmented by using a momentum parameter μ. According to this the new algorithm is [Osowski, 1996]:

$$\Theta(k+1) = \Theta(k) - \eta R \nabla_{\Theta} E(\Theta) + \mu(\Theta(k) - \Theta(k-1)) \tag{24}$$

The idea of using momentum is motivated by the need to escape from local minima, which may be effective in certain problems. In general, however, the recommendation is to use one of the other, better, training algorithms and repeat the training a couple of times from different initial parameter initializations.

5. Results of research

5.1 Research conditions

During research the five ships were measured on the Polish Navy Test and Evaluation Acoustic Ranges which schema was presented on figure 1. Ships No. 1 was minesweeper project 206FM, ship No. 2 was minesweeper project 207D, ship No. 3 was salvage ship project 570, ship No. 4 was minesweeper project 207P, and ship No. 5 was racket corvette project 1241RE.

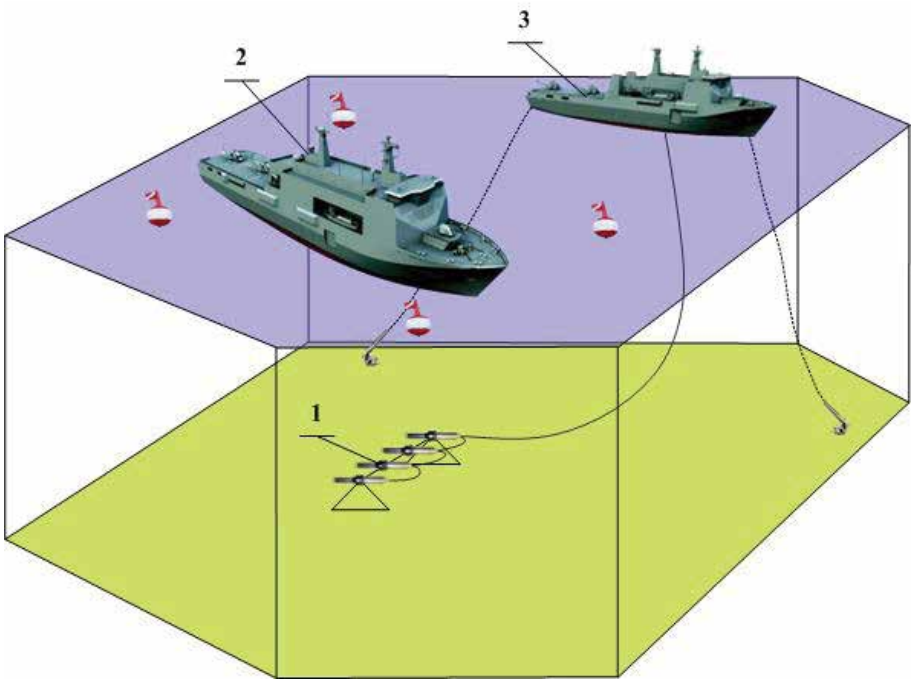


Fig. 12. Schema of hydroacoustic range during measurements; 1) sensors of acoustic signatures – array of hydrophones, 2) measured ship, 3) ship – base with mounted hydroacoustic measuring system

The recordings were carried out by means of the array of hydrophones. Several hydrophones were strung in a line along the bottom in shallow water. The depth was about 10 m. During the ship measurements, the average see wave height was less than 1 m and wind speeds less than 5 m/s, so the ambient noise level was low. At the time of the measurements the sound velocity profile was typical for the summer. This curve was smooth with gradually decreasing gradient without mixed layers. The ship under test was running at a constant speed and course during cross over hydrophones. The array of hydrophones was mounted about 1 m above sea bottom on tripod. The bottom-mounted hydrophones range is very useful for measuring the noise of surface ships. What more when they are used bottom-fixed hydrophones the irrelevant low-frequency wave-induced noise is also eliminated. Throughout this measurement, the signal-to-noise ratio for the spectrum data was greater than 28 dB.
All of investigated ships were measured at the similar hydrological and metrological conditions. Every ship was measured with few, various speed of crossing.
Data from hydrophones were recorded on digital recorder designed by crew of Hydroacoustics Institute of Polish Naval Academy. This system has possibility to simultaneous recording in 16 channels with resolution of 16 bits and sampling frequency up to 250 kHz per channel. Digital recorder has possibility to make in real time transformation and analysis of acquired data. More over it is possible to create own programs for special use. As a sensors of acoustic field of moving ship were used hydrophones produced by Reson model TC4032. This hydrophones has omnidirectional characteristic in horizontal directivity so they were positioned parallel to the plane of sea bottom. Other parameters which cause that these sensors are proper to acquire data for classification systems are: high sensitivity equal -170 dB re 1V/ μ Pa, preamplifier gain of 10 dB and broad usable frequency range from 5 Hz to 120 kHz. Mentioned above digital recorder has possibility to direct connections of hydrophones TC4032.

5.2 Parameters setup

The best solutions to detect a ship are the discrete components in the low frequency part of the ship's noise spectrum and that only narrow band filters can be used. This must be done because there are no components discrete lines at frequencies range greater than 200 Hz in the modern submarines and surface warships. In the Baltic's shallow waters and the conditions under which the measurements were made, the area of optimal frequencies for the propagation of sound lies in the band from several Hz up to 5 kHz.
Recorded during research signals were sampled on digital recorder with frequency of 250 kHz. From the theoretical point of view (Shanon-Kotielnikow Law) it is enough for used sensors which has the upper band of frequency equal 120 kHz. From the practical point of view it is advisable to have 10 samples per period of highest frequency of analyzed signals. In this case we have usable band of signals up to 25 kHz. In research we need signal of band frequency from 5 Hz (because of used hydrophones) up to 200 Hz (because of existence of discrete lines in spectrum). So used measured system is suitable for this research.
To cut off signals above 200 Hz it can be used some digital or analog filters. In other hands using filters may cause to raise the noise-to-signal ratio. Therefore in research we do not use filters but after calculation of spectrum we will use only data which are above 5 Hz and below 200 Hz.

5.3 Results

Classification is made by neural classifier in which first layer was Kohonen network which has two dimensional architecture and second layer was feedforward neural network, both described above. Characteristic parameters of used Kohonen network are: number of neurons, beginning size of area of the neighborhood, beginning learning rate and methods to determine the distance between neuron weights and input vectors. Because there is no theory about beginning setup of mentioned above neural network's parameters there were made few experimental research. For this case because of speed of learning, possibilities to classify data and possibilities to generalize the knowledge it seems that follows values are the best: number of neurons: 30x30 neurons map, beginning size of area of neighborhood: 3, beginning learning rate: 0.35 and method to determine the distance: Euclidean distance.

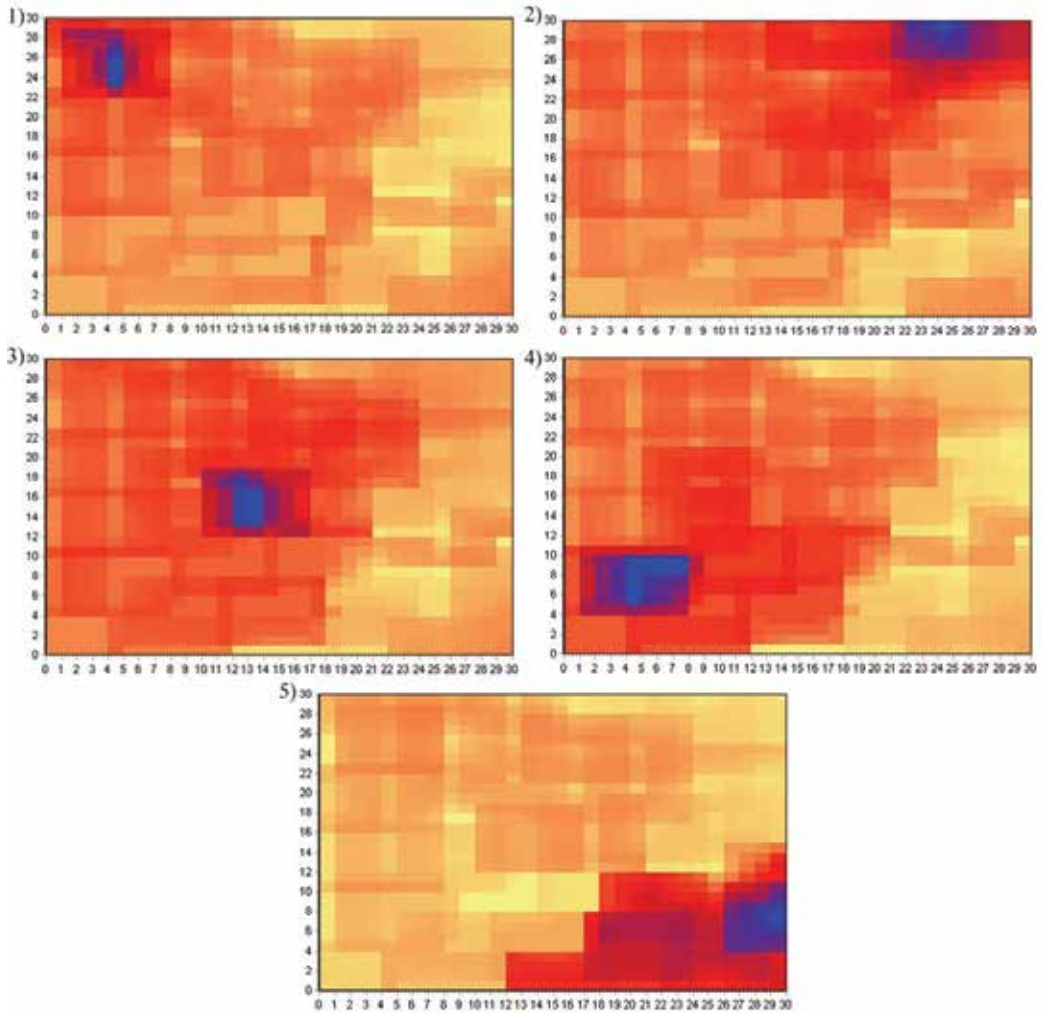


Fig. 13. The results of classifier work out - maps of memberships 1) for ship no. 1, 2) for ship no. 2, 3) for ship no. 3, 4) for ship no. 4, 5) for ship no. 5

As it was said second layer of neural classifier was feedforward neural network which characteristic parameters are: number of layers, number of neuron in each layer, learning rate, momentum rate, methods of learning. In this experiment it seems that using 3 layers feedforward neural network (one input layer, one hidden layer and one output layer). The value of learning rate was set up at 0.3 and momentum rate at 0.65. With this parameters the possibility to classification (number of correct answers, speed of learning and minimum number of neurons) was the best.
After about 35 000 cycles of neural network learning, from the first layer on neural classifier (Kohonen network) was obtained the map of memberships for every presented ship as it is shown on figure 13.
All areas activated by signals generated by considered ships were clearly separated as it was shown on figure 14.

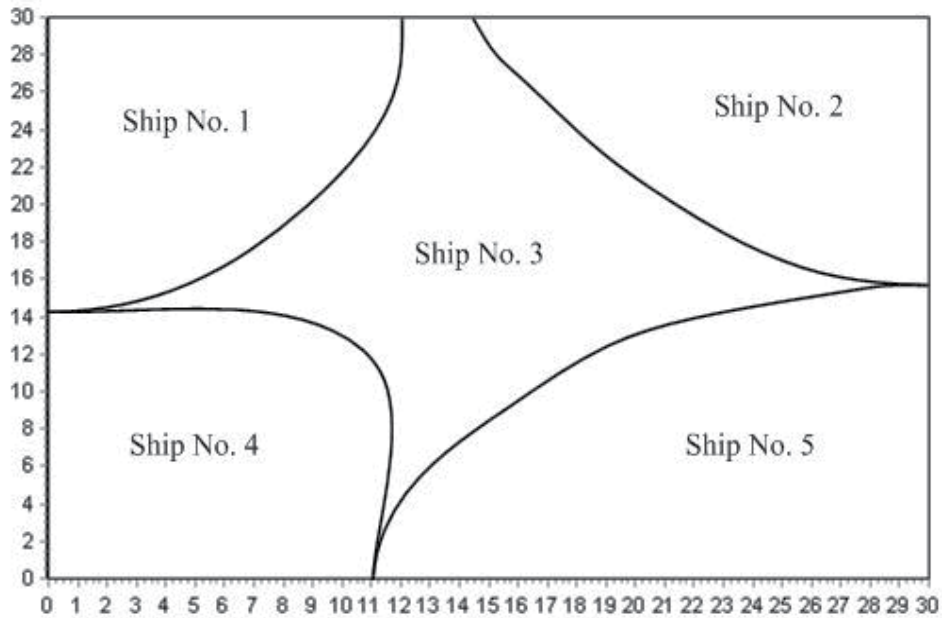


Fig. 14. The map of partition for area of activation for researched ships

Now to assign for each active region of membership maps to the name of ship's class we use the second layer of neural classifier (feedforward network). Using supervised learning we create fully usefulness classifier.
Firstly to check if the classifier works properly we make test on data which were presented during neural network learning process. To find out if the building classifier is properly configured and learned some data which weren't presented before were calculated. The table 3 shows number of correct classification of presented data relatively to the type of ship. The number of correct answer is presented as percent of all answers. The research was made for data which were presented during learning process and data which weren't presented before.
After this part of researches the new ship No. 6 which was rocket corvette project 1241.1MP was presented. In few first presentations it was classified as ship No. 5 what was

comprehensible because ship No. 5 is the oldest version of this vessel. Next the new group was created, which was separated from the area activated before by ship No. 5. The new map of partition for area of activation looks like is presented on figure 15.

Ship no. / Data	1	2	3	4	5
presented before	94.5%	96.0%	92.3%	95.3%	92.8%
not presented before	72.1%	69.4%	75.8%	73.5%	77.2%

Table 3. The number of correct classifications

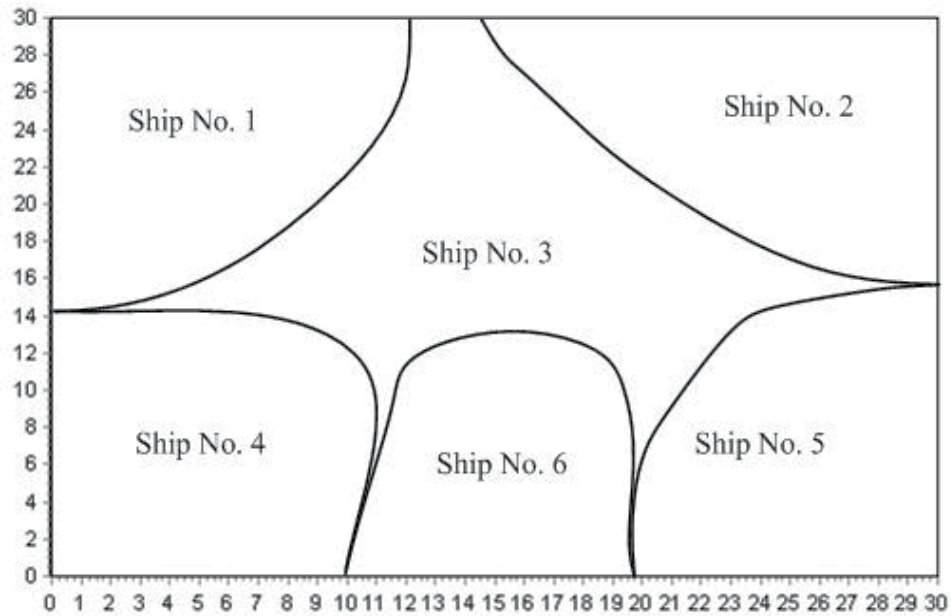


Fig. 15. The new map of partition for area of activation for researched ships after introducing new ship

6. Conclusion

As it is shown on results the used Self-Organizing Map is useful for ships classification based on its hydroacoustics signature. Classification of signals that were used during learning process, characterize the high number of correct answer (above 90%) what was expected. This result means that used Kohonen network associated with feedforward network has been correctly configured and learned. Presentation of signals that weren't used during learning process, gives lowest value of percent of correct answer than in previous case but this results is very high too (about 70 % of correct classification). This means that neural classifier has good ability to generalize the knowledge. More over after

presentation of new ship which weren't taking into account during creating classifier, the Kohonen networks was able to create new group dividing the group which belongs to the similar type of ship. After few cycles used neural networks expand its output vector or in other words map of membership about new area of activation. This means that used Kohonen networks has possibility to develop its own knowledge so it cause that presented method of classification is very flexible and is able to adaptation to changing conditions.

Presented case is quite simple because it not take into account that object sounds change with time, efficiency conditions (e.g. some elements of machinery are damaged), sound rates, etc. It doesn't consider the influence of changes of environment on acquired hydroacoustics signals. In next step of research the proper work of this method will be checked for enlarged vector of objects. The hydroacoustics signatures of ships were acquired in different environmental conditions and in different stage of ship operating. Therefore the cases of changing hydroacoustics signatures which were mentioned before should be investigated too.

In future research the influence of network configuration on the quality of classification should be checked. More over some consideration about feature extracting from hydroacoustics signature should be made.

Described method after successful research mentioned above and after preparation for work in real time will be extended and its application is provided as assistant subsystem for passive hydrolocations systems of Polish Naval ships.

The aim of presented method is to classify and recognize ships basing on its acoustic signatures. This method can found application in intelligence submarine weapon and in hydrolocation systems. In other hand it is important to deform and cheat the similar system of our opponents by changing the "acoustic portrait" of own ships. From the point of ship's passive defense view it is desirable to minimize the range of acoustic signatures propagation. Noise isolation systems for vessels employ a wide range of techniques, especially double-elastic devices in the case of diesel generators and main engines. Also, rotating machinery and moving parts should be dynamically-balanced to reduce the noise. In addition, the equipment should be mounted in special acoustically insulated housings (special kind of containers). One of the method to change the hydroacoustics signatures is to pump the air under the hull of ship. It cause the offset of generated by moving ship frequency into the direction of high frequency, the same the range of propagation become smaller.

7. References

Fort J. C., SOM's mathematics, *Neural Networks*, 19: 812–816, 2006

Gloza I., Malinowski S. J., Underwater Noise Radaited by Ships, Their Propulsion and Auxiliary Machinery and Propellers, *Hydroacoustics vol*. 4, pp. 165-168, 2001

Gloza I., Malinowski S. J., Underwater Noise Characteristics of Small Ships, *Acta Acoustica United with Acustica*, vol. 88 pp. 718-721, 2002

Haykin S., *Self-organizing maps, Neural networks - A comprehensive foundation*, 2nd edition, Prentice-Hall, 1999

Kohonen T., *Self-Organizing Maps*, Third, extended edition, Springer, 2001

Osowski S., *Neural Networks*, Publishing House of Warsaw University of Technology, 1996

Stąpor K., *Automatic object classification*, Publishing House EXIT 2005

Szczepaniak P. S., *Intelligence Calculations, Fast Transforms and Classification*, Publishing House EXIT, 2004

Therrien Ch. W., *Discrete Random Signals and Statistical Signal Processing*, Prentice Hall International, Inc. 1992

Urick R. J., *Principles of Underwater Sounds*, McGraw-Hill, New York 1975

Zak A., Creating patterns for hydroacoustics signals, *Hydroacoustics* Vol. 8, pp. 265- 270, 2005

Zak A., Neural Classification Of Ships Hydroacoustics Signatures, Proceedings of the 9th European Conference on Underwater Acoustics ECUA 2008, Paris, France 2008, pp. 829-834

13

Dynamic Vehicle Routing Problem for Medical Emergency Management

Jean-Charles Créput[1], Amir Hajjam[1],
Abderrafiãa Koukam[1] and Olivier Kuhn[2,3]
[1]*Systems and Transportation Laboratory, U.T.B.M., 90010 Belfort Cedex,*
[2]*Université Lyon 1, LIRIS, UMR5205, F-69622 Villeurbanne,*
[3]*Université de Lyon, CNRS*
France

1. Introduction

Nowadays telemedicine applications are more and more present in the state-of-the-art medicine. Telemedicine is a good way to improve access to healthcare, quality of care, reduce isolation and also costs. In that way we can now safely perform surgery between two places separated by several thousand km, navigate in 3D models of blood vessels or generate 3D models from Nuclear-Magnetic Resonance Imaging (MRI). But there is currently a lack of tools for all day medical acts which could improve medical system efficiency especially for medical emergency services.
In order to help medical emergency services, the project MERCURE (Mobile and Network for the Private clinic, the Urgency or the External Residence) has been launched in order to create tools that optimize, follow and manage emergency interventions. The current problem is that the choice of the doctor for a patient is done by hand.The call center is neither aware of the exact location nor the current state of the doctors. Thus it is rarely the best located doctor who is chosen and moreover he may not have correct equipments to heal the patient. To optimize that aspect, we have developed software allowing the optimized management of human and material medical resources.
This problem, conventionally called vehicle routing problem (VRP), is one of the most widely studied problems in combinatorial optimization. In the standard VRP, a fleet of vehicles must be routed to visit a set of customers at minimum cost, subject to vehicle capacity constraint and route duration constraint. In the static version of the problem, it is assumed that all customers are known in advance to the planning process. In the case of medical emergency management, it includes some dynamic elements. The information data often tends to be uncertain or even unknown at the time of the planning. It may be the case that patients, driving times or service times, are unknown before the day of operation has begun, but become available in real-time. Due to the recent advances in information and communication technologies, such as geographic information systems (GIS), global positioning systems (GPS) and mobile phones, companies are now able to manage vehicle routes in real-time. Hence, with the increased access to these services, the need for robust real-time optimization procedures will be of critical importance, for small to big distribution companies, whose logistics are based on a high reactivity to the customer demand.

As for static vehicle routing problems, a lot of versions of the dynamic problem exist depending on application areas. For an overview and classification of the numerous versions of real-time routing and dispatching problems, we refer the reader to the general surveys and classifications given in (Ghiani & al., 2003), (Larsen, 2000), (Larsen &al., 2008), (Gendreau &Potvin, 1998) and (Psaraftis, 1995), (Psaraftis, 1998). One of the simplest versions is the standard dynamic VRP with capacity and time duration constraints (Kilby & al., 1998), called "dynamic VRP" in this paper, which is a straightforward extension of the classical static VRP (Christofides & al., 1979). In this problem, the customers are the only elements which have a dependence on time. Customers are not known in advance but arrive as the day progresse. The system has to incorporate them into the already designed routes in real time. Problems fitting this model appear frequently in industry.
A lot of different versions of the dynamic VRP have been studied, whereas very few dynamic routing problems except the dynamic VRPTW or dynamic PDPTW are recognized as standard problems well suited to allow comparative evaluations of heuristics and metaheuristics on a common set of benchmarks. For example, only two papers on the dynamic VRP that shared detailed results on a common test set have been found. They are first an adaptation of the ant colony approach MACS-VRPTW Gambardella & al., 1999) by (Montemanni & al., 2005), and second a genetic algorithm (Goncalves & al., 2007). They share results (Kilby & al., 1998), test set with 22 problems of sizes from 50 with up to 385 customers. This paper tries to go one step further in that direction considering the dynamic VRP as a standard dynamic problem, and yielding a comparative study with these two methods on the Kilby et al. test set. Then, we restrict the scope of our work to the dynamic VRP, with capacity and time duration constraints.
In the following section, the MERCURE project will be presented. In section 3 we shall introduce our optimization system with implementation details. Then, section 4 reports experiments carried out on the Kilby at al. benchmark and the comparisons made with a state-of-the-art ant colony approach and a genetic algorithm already studied on these benchmarks. Finally, last section is devoted to the conclusion and further research.

2. Project MERCURE

2.1 Aim of the project

The project MERCURE takes part in the French pole of competitiveness therapeutic innovations. The aim of the project is to give, thanks to information technologies, an optimized and dynamic management of resources used in the scope of urgentist interventions like material and human resources. The system gives a real-time tracking of current interventions, from the reception of the call to the closure of the medical record. It optimizes resources, travel times and takes care of whole constraints relative to the domain: emergency level, pathology, medical competences, location and other specific aspects related to this profession.
The platform exploits satellite location system associated with a geographical information system (GIS) and is based on results coming from works on vehicle routing problems (Creput & al., 2007). With present technologies we can have accurate current location of patients and doctors via GIS and A-GPS1 respectively. The A-GPS system has 3 main uses.

1. Know the position of each medical team.
2. Help the doctor to reach quickly the intervention point.
3. Track in real-time medical teams and resources.

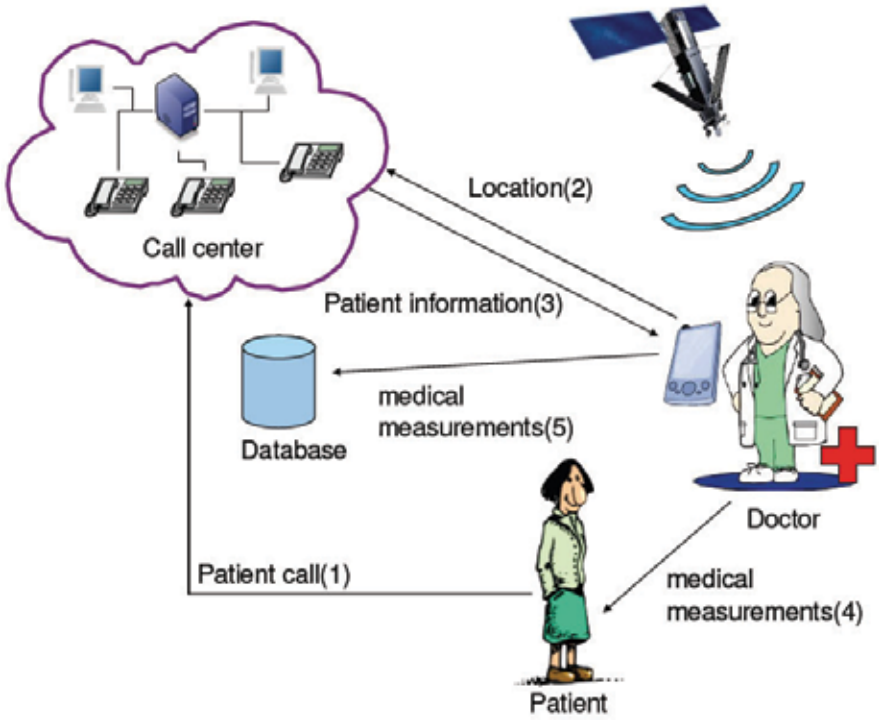


Fig. 1. Data exchanges after a patient call

Here is a basic scenario when an emergency call arrives (see figure 1). Call center point of view:

- Information about the patient (name, address, pathology…) is inputted in the software.
- Patient's information are processed, a set of doctors which suit to the patient's needs is created (depending of the pathology, the intervention area…).
- The selection of a doctor in the previously created set is done via an optimization algorithm. Here we focus on optimizing several criteria like distance, reaction time. . . The patient is inserted in the doctor's road.
- The selected doctor is warned by a message on his PDA2.

Now from a doctor point of view:

- The doctor receives a patient request on his PDA and he is geo-guided to the patient's location via A-GPS.
- As soon as he arrives, all information about the patient are shown: previous diseases, his allergy, current treatments. . . Those information are transferred from the database via radio link like GPRS3 or UMTS4 for example.
- When the auscultation is finished, he inputs results and notes that are immediately transferred to the central database. Then he goes on with the next patient.

2.2 Improvements

This whole process improves reaction time of emergency services and thus save lives. It also provides an unique database gathering up-to-date information about patients and so facilitate the follow-up of patients. Another main improvement is that the answer fits to the

patient's needs. In other words, the call is answered by a doctor-regulator who is able to help the patient to describe and specify his illness. This is a real telemedicine act and thus the software is able to select the appropriate doctor or send an ambulance. Moreover this system may suit to other emergency services like fire brigade or police department with some adaptations. There are some papers about ambulances location and relocation models written by (Gendreau & al., 1997), (Gendreau & al., 1999), (Gendreau & al., 2001) and (Brotcorne & al., 2003). But currently we are not aware of other tools for such size of emergency services. This project is realizable thanks to recent new technologies like A-GPS, wireless data communication and improvements in artificial intelligence and operations research for dynamic problems.

3. Dynamic optimization system for urgentist

In the MERCURE project, we are in charge of the optimization part for the selection of doctors and assignment of patients. We have tackled this problem as an operations research problem named Vehicle Routing Problem (VRP) (Toth & Vigo, 2001).

3.1 Problem statement

Allan Larsen stated in his PhD report (Larsen, 2000) that emergency services have 2 major criteria (see figure 2):

- They are highly dynamic: most or all requests are unknown at the beginning and we have no information about their arrival time.
- The response time must be very low because lives can be in danger.

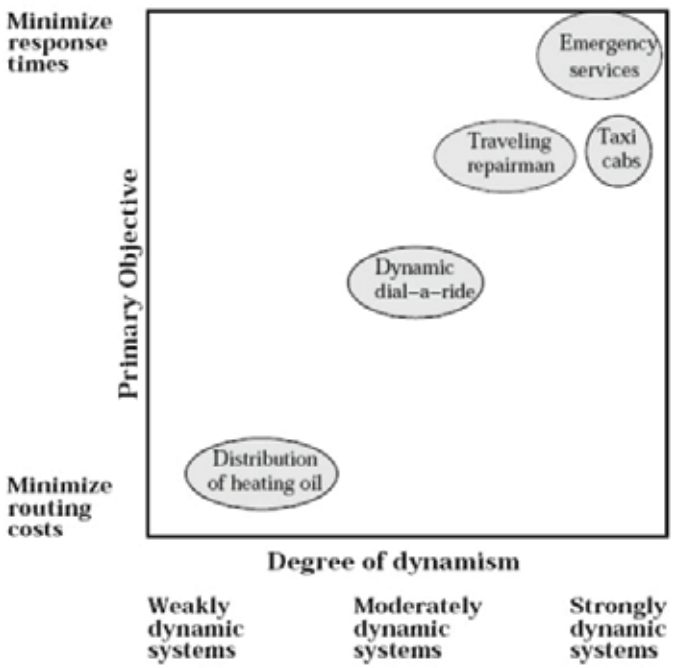


Fig. 2. Framework for classifying dynamic routing problems by their degree of dynamism and their objective

That is why we have chosen to represent the emergency problem as a Dynamic Vehicle Routing Problem with Time Window (DVRPTW) which is presented in the next paragraph. This extension of the well known VRP suits very well to this kind of problem because it takes care of the 2 criteria previously stated. Time windows are perfect to consider response time and the dynamic aspect allows the system to receive requests during the optimization process.

1) DVRPTW presentation: A Dynamic Vehicle Routing Problem with Time Windows is a specialization of the well known Vehicle Routing Problem. The static VRP is defined on a set $V = \{v_0, v_1, ..., v_N\}$ of vertices, where vertex v_0 is a depot at which are based m identical vehicles of capacity Q, while the remaining N vertices represent customers, also called requests, orders or demands. A non-negative cost, or travel time, is defined for each edge $(v_i, v_j) \in V \times V$. Each customer has a non-negative load $q(v_i)$ and a non-negative service time $s(v_i)$. A vehicle route is a circuit on vertices. The VRP consists of designing a set of m vehicle routes of least total cost, each starting and ending at the depot, such that each customer is visited exactly once by a vehicle, the total demand of any route does not exceed Q, and the total duration of any route does not exceed a preset bound T (see figure 3).

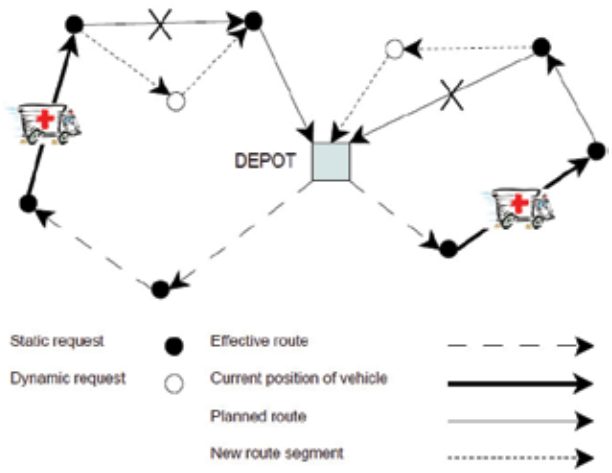


Fig. 3. Example of dynamic vehicle routing problem with 7 static requests and 2 immediate requests

As it is the mostly done in practice (Cordeau & al., 2005), we address the Euclidean VRP where each vertex v_i has a location in the plane, and where the travel cost is given by the Euclidean distance $d(v_i, v_j)$ for each edge $(v_i, v_j) \in V \times V$. Then, the objective for the static problem is the total route length (*Length*) defined by

$$Length = \sum_{i=1,..,m} \left(\sum_{j=1,...,k_i-1} d\left(v_j^i, v_{j+1}^i\right) + d\left(v_0, v_1^i\right) + d\left(v_{k_i}^i, v_0\right) \right) \quad (1)$$

where $v_j^i \in V$, $0 \le j \le k_i$, $0 \le k_i \le N$, are the ordered set of demands served by the vehicle i, $1 \le i \le m$, *i.e.* the vehicle route. The capacity constraint is defined by

$$\sum_{j=1,...,k_i} q\left(v_j^i\right) \le Q\,, \; i \in \{1,...,m\} \quad (2)$$

then, assuming without loss of generality that the vehicle speed has value 1 the time duration constraint is given by

$$\sum_{j=1,\ldots,k_i} s\left(v_j^i\right) + \sum_{j=1,\ldots,k_i-1} d\left(v_j^i, v_{j+1}^i\right) + d\left(v_0, v_1^i\right) + d\left(v_{k_i}^i, v_0\right) \leq T \ , \ i \in \{1,\ldots,m\} \tag{3}$$

The problem is NP-hard. Thus, using heuristics is encouraged in that they have statistical or empirical guaranty to find good solutions for large scale problems with several hundreds of customers. For example, the most powerful Operations Research (OR) heuristics for the VRP, referred in the extensive surveys (Gendreau & al., 2002), (Cordeau & al., 2005), are based on metaheuristic frameworks as the Tabu Search, simulated annealing, and population based methods, such as evolutionary algorithms, adaptive memory and ant algorithms. Other methods can hybridize several metaheuristics principles, such as for example the very powerful active guided local search (Mester & Bräysy, 2005), which is maybe the overall winner approach considering both quality solution and computation time.

In the static VRP, vehicles must be routed to visit a set of customers at minimum cost, assuming that all orders for all customers are known in advance. However, in the dynamic VRP, new tasks enter the system and must be incorporated into the vehicle schedules and served as the day progresses. In real-time distribution systems, demands arrive randomly in time and the dispatching of vehicles is a continuous process of collecting demands, forming and optimizing tours, and dispatching requests to vehicles in order to process requests at the required geographic locations. In the case of the static VRP, the three phases of demands reception, routes optimization and vehicles travelling are clearly separated and sequentially performed, the output of a given phase being the input of the subsequent one. At the opposite, we can see the dynamic VRP as an extension of the static VRP where these three time-dependent processes are merged into an approximately same period of time. This period of time is called the working day or planning horizon of length D. Here, we precisely define the working day length D as the length of the collecting period, knowing that the optimization period and the vehicle travelling period would have to be of approximately the same length.

It is often the case that in real life situations the objective function consists of a trade-off between travel costs and customer waiting time *i.e.* the delay between the occurrence time of a demand and the instant the service of the demand begins, often called system response time in the literature. Hence, we define the dynamic VRP as a bi-objective problem by adding to the classical objective and constraints of the standard VRP a supplementary objective which consists of minimizing the average customer waiting time. In a dynamic setting the waiting time can be more or less important depending on the application at hand. Examples of applications where the waiting time is the important factor include the replenishment of stocks in a manufacturing context, the management of taxi cabs, the dispatch of emergency services, geographically dispersed failures to be serviced by a mobile repairman. It is then necessary to identify the many trade-offs between these two objectives. Hence, to gauge the reactivity and the dynamism of the system, a real-time objective consists in minimizing the average customer waiting time (WT) :

$$WT = \sum_{i \in \{1,\ldots,N\}} W_i \Big/ N \tag{4}$$

where W_i is the waiting time of demand i, *i.e.* $W_i = st_i - t_i$ where $t_i \in [0, D]$ is the demand occurrence time, and st_i is the time when the service starts for that demand.

It is worth noting that the total route length and the classical constraints of capacity and time duration are evaluated exactly the same way as for the static problem case. This is done in order to be the closest as possible to the standard problem formulation and to allow comparisons between the solutions generated in both the dynamic and static cases. Hence, a route remains a simple schedule of demands. Whereas, in order to evaluate the customer waiting time we need to consider travel distances and service times, but also consider the "real time" at which the service is really performed, thus taking care of the possible extra times during which the vehicle may be waiting, or driving back to the depot before some new requests are dispatched to it. It should be noted also that we assume that no information is available about the future locations of the demands.

Also, it may be possible that a vehicle will finish its work and go back to the depot after the period D has finished. Hence, in order to gauge what is the real part of the services that are performed within the working day in real-time or after the day has finished once all demands are already known, it may be useful to compute an auxiliary criterion that we define as the real finishing time of the vehicle services, *i.e.* the date when all the vehicles have finished their service and have returned to the depot. In this way, looking both at the vehicle lengths and at the finishing time will give another intuitive light about the dynamicity of the system. Thus, we define the maximum vehicle finishing time (MT) as

$$MT = \underset{k\in\{1,\ldots,m\}}{Max} \{FT_k\} \tag{5}$$

where FT_k is the vehicle finishing time of vehicle k, that is, the occurrence time at which the vehicle arrives to the depot once it has served its last customer for the day. We will see that this finishing time can be maintain in adequate bounds even when introducing some delay to the departure of the vehicles, thus drastically and simultaneously reducing the total route length.

Clearly, only the evaluations of equations (4) and (5) depend on a real-time realization, whereas the evaluations of (1)-(3) only depend on the scheduling of the demands the same way as for the standard VRP. Then, to empirically evaluate a given real-time optimization approach, we need to embed its execution in a real-time simulator.

Between a VRP and a DVRPTW, 2 constraints are added:

- Usually, relevant information, such as new patient requests and cancelled requests can occur all the time, even after the optimization process has started. The dynamism consists in receiving several requests during the evolution of the simulation. These dynamic variations can be very important to really reduce the costs in vehicle routing problems. The date when the request i arrives is noted g_i as the generation date of request i.
- The time windows constraint which consists in having 2 time limits associated with each request i: $[a_i, b_i]$. The vehicle must start the customer service before b_i, but if any of them arrives at customer i before a_i, it must wait. So the smaller the time window of a request is the harder will it be to find a good insertion place in a vehicle road.

To these 2 constraints, a third one can be added depending of the instance of the problem. It comes from the fact that all doctors may not start from the same location so we must manage multi-depots instances of DVRPTW.

2) Matching to DVRPTW: We have to affect each real entity (call center, resources and patients) to one in routing problem which are vehicles, requests and the company. The most

logical way to make them correspond is to match the doctors to the vehicles, the patients to the requests and the call center to the company.
But there is some specificity that we must consider in the problem.
First the patient may need a specialist for his illness. So not all vehicles can serve this request. It is the same thing for ambulances. In the same way, we must avoid sending a woman into a district with bad reputation. We need to have in our application different types of vehicle which is not managed in classical VRP instances where all vehicles are identical. So in our DOS a request can be dedicated to a vehicle and only this vehicle can serve it.
We must also take care of the loading of the system. We have a time constraint that is specific to emergency services. In classical VRPTW, when some requests are not served at the end of the day they are deferred to the next day. Here when the system is overloaded, we must serve most urgent requests and redirect less urgent ones to a classical doctor if possible.

3.2 Dynamic optimization system

In order to solve DVRPTW, we have developed a simulator that we have called Dynamic Optimization System (DOS). You can find some screenshots in figure 4.

Fig. 4. Evolution of a simulation in DOS on a static benchmark. Dotted lines represent the road segments that have been completed

1. Architecture of the simulator: This simulator is divided in 2 distinct parts.

On the one hand we have a multi-agents simulator. Its role is to schedule main entities present in a VRP. Each entity is represented by a process.

- The environment process is dedicated to generate events during the simulation.
- The company process simulates a real company. It receives requests and plans vehicles roads.
- Vehicles processes follow roads given by the company and serve requests.

All these processes are synchronized on a same clock owned by the scheduler so they advance in time simultaneously. One simulation step lasts *To* milliseconds in real time and the corresponding simulation time depends on a ratio to suit the problem. As our application domain is in real time, the ratio will be 1. So each step will last *To* millisecond in the simulation.
During a step, each process is called once to make a short action and so share CPU time as shown in figure 5. Actions that need a lot of time must be divided in several shorter actions with small execution time.

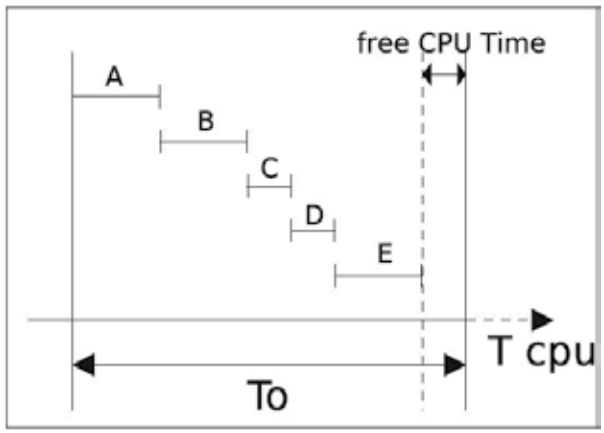


Fig. 5. CPU sharing between 5 processes (A to E) during To ms

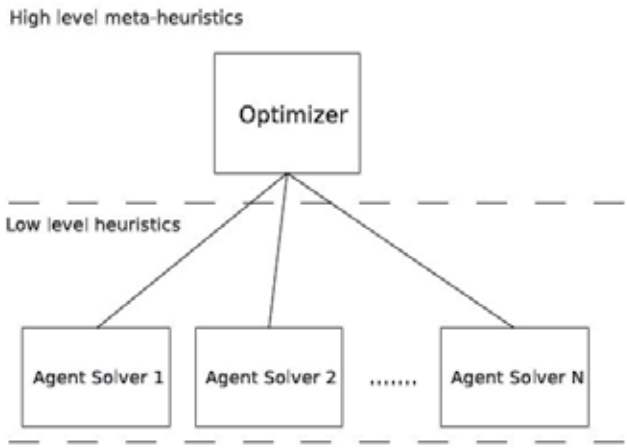


Fig. 6. Architecture of the optimization part of the simulator

One the other hand we have the optimization part.

The optimization process can be viewed as a black box, receiving the current solution (a set of vehicles) and the known requests and giving back a better solution if possible. This part of our DOS is explained more precisely in 4. The company has the role of asking the optimizer to optimize current solution. After a fixed time, the company reads the solution and gives new plans to the vehicles or confirms the current one. The exchanges between the two parts are done via a letterbox with exclusive access. This ensures the data transfers between two unsynchronized threads and prevents data overriding.

2. Additional features: Through this system we can also gather lots of interesting information that can be processed in order to extract some statistical data. We can imagine optimizing the number of doctors depending of the date, the specialization the most needed and so on. Once enough requests are stored in the database, we can extract main trends and optimize human and logistic resources.

Moreover we can use that probabilistic information on future events to route doctors to their next patient by making them pass close to area with high probability of new requests.

(Bertsimas & al., 1990) describe this kind of problems and call them Probabilistic Vehicle Routing Problem (PVRP).

4. Optimization approach

We are now facing a DVRPTW that we must solve relatively quickly in order to be able to warn doctors of a new patient to see urgently. In a VRP problem, finding one of the best solutions requires a lot of time. Here we prefer having a relatively good solution quickly and then improve it.
To do that, our optimizer has a 2-level architecture. The top level uses a global meta-heuristic strategy and controls several solver agents. In the lower level we can find previously mentioned solver agents which represent different heuristics for solving VRP (see figure 6).

A. Global Meta-heuristic

This level aims at finding the best solution by using several solver agents. Each of these agents represents a heuristic for solving VRP (see 4-B). That can be seen as a worker with different tools (the solver agents) at his disposal for doing his job, here optimizing vehicles routes. It has to choose the strategy which suits the best to the problem for example creating the first solution. To do that, the optimizer initializes a set of selected solver agents and tells them to do the job separately. Then before a defined generation time (Tg) it gathers all solutions from the agents and makes a selection to keep most interesting ones depending on the strategy and then gives the best solution to the company via the letterbox. So we manage a population of solution where we keep or replace individuals like in genetic algorithms. This allows exchanging solutions between different heuristics and so discovering new ones and getting out local optima.
The solver agents are scheduled by the optimizer like the processes in the simulation part. When all used solver agents have been activated once, one step is done. So we can have several different optimization methods in parallel. The specifics of our solver agents are approached in the next part.

B. Low-level heuristics

We shall now analyze the lower level where solver agents are located. Their aim is to solve a type of VRP thanks to a specified heuristic. Each agent uses one or more heuristics which can be very basic like a 2-opt which consists in exchanging 2 roads (see figure 7) or more complicated like neural networks or other artificial intelligence methods. The optimizer is aware of features of all solver agents.

1. **Memetic SOM:** The main optimization algorithm we are using is based on local search (Rochas & Taillard, 1995) and selforganizing maps (SOM) (Kohonen, 24), (Ghaziri, 1996), (Modares & al., 1999), by embedding them into an evolutionary algorithm. This approach is called memetic SOM (Creput & al., 2007).

One way to explain the "philosophy" of the approach may be by referring the reader to some well known concepts in the Artificial Intelligence domain like emergent computation, bio-inspired methods, and soft-computing concepts including neural network, evolutionary algorithms, or hybrid systems. The approach can be seen as following a biologic metaphor where customers constitute external stimuli to which a "biologic organism", may respond dynamically adapting its shape continuously to absorb, neutralize or satisfy the external

stimuli. More generally, we can exploit this metaphor to address a large class of spatially distributed problems of terrestrial transportation and telecommunications, such as facility location problems, vehicle routing problems or dimensioning mobile communication networks (Creput & al., 2005), (Creput & Koukam, 2007). These problems involve the distribution of a set of entities over an area (the demand) and a set of physical systems (the suppliers) which have to respond optimally relatively to the demand. This optimal response constitutes the solution to the optimization problem. Thus, a distributed bio-inspired heuristic to address such problems is a simulation process of such spatially distributed entities (vehicles, antenna, customers) interacting in an environment which produces the "emergence" of a solution by the many local and distributed interactions

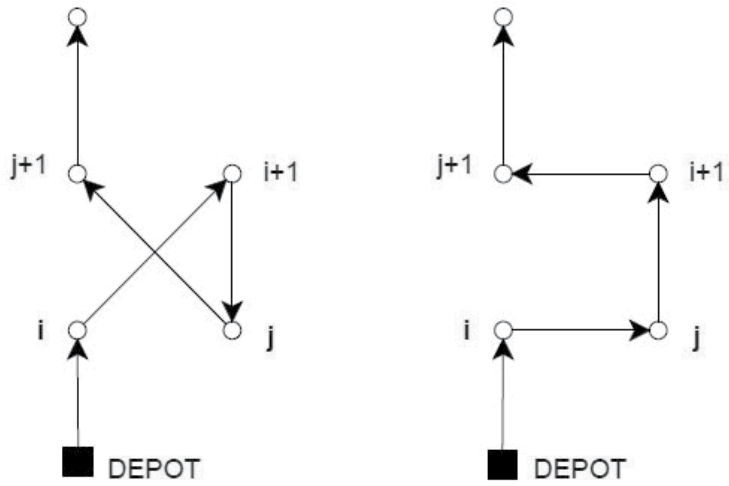


Fig. 7. Example of a 2-opt operation

Here, we generalize the SOM algorithm giving rise to a class of "closest point findings" based operators that are embedded into a population based metaheuristic framework. The structure of the metaheuristic is similar to the memetic algorithm, which is an evolutionary algorithm incorporating a local search (Moscato & Cotta, 2003). The SOM is a (long) stochastic gradient descent performed during the many generations allowed, and used as a "local search" similarly as in a classical memetic algorithm. This is why the approach has been called memetic SOM (Moscato & Cotta, 2003) in previous work and we will maintain the name in this paper. The approach follows two types of metaphors. It follows a self-organization metaphor at the level of the interacting problem components, or heuristic level, and an evolution based metaphor at the population based metaheuristic level. Since demands are conceptually separated from the routes representation, which is an independent network or graph in the plane which continuously adjusts itself to the data, this leads to a straightforward application from a static to a dynamic setting. As they arrive, new demands are simply inserted on-line in a buffer of demands, in constant time, leading to a very weak impact on the course of the optimization process.

The evolutionary algorithm embedding SOM is based on memetic loop which applies at each iteration (called a generation) a set of operators to a population of individuals. The construction loop starts its execution with solutions having randomly generated vertex

coordinates, into a rectangle area containing cities. The improvement loop starts with the single best previously constructed solution, which is duplicated in the new population. The main operator is the SOM algorithm applied to the graph network. At each generation, a predefined number of SOM basic iterations are performed letting the decreasing run being interrupted and combined with application of other operators, which can be other SOM operators with their own parameters, mapping and fitness evaluation, and selection. Each operator is applied with probability prob. Details of operators are the followings:

1. Self-organizing map operator. It is the standard SOM applied to the ring network. One or more instances of the operator can be combined with their own parameter values. A SOM operator is executed performing η_{iter} basic iterations by individual, at each generation.
2. SOM derived operators. Two problem specific operators are derived from the SOM algorithm structure for dealing with the VRP especially. The first, denoted SOM VRP, is like a standard SOM but restricted to be applied on a randomly chosen vehicle, using requests already assigned to that vehicle. While capacity constraint will be considered in the mapping operator below, a SOM based operator, denoted SOM DVRP, deals with the time duration constraint. It performs a greedy insertion move.
3. Fitness/assignment operator. This operator, denoted FITNESS, generates a VRP solution and modifies the shape of the ring accordingly. The operators greedily maps customers to their nearest neuron, considering only the neurons not already assigned to a customer, and where vehicle capacity constraint is satisfied. The capacity constraint is then greedily tackled through the requests assignment. Once the assignment of requests to routes has been performed for each individual this operator evaluates a scalar fitness value that has to be maximized and which is used by the selection operator. Taking care of time duration constraint the fitness value is computed sequentially following routes one by one and removing a request from the route assignment if it leads to a violation of the time duration constraint.
4. Selection operators. Based on fitness maximization, the operator denoted SELECT replaces worst individuals, which have the lowest fitness values in the population, by the same number of bests individuals, which have the highest fitness values in the population.

The memetic SOM is very interesting because of its adaptability and flexibility due to its neighbourhood search capabilities and simple moves performed in the plane. We can easily add or remove requests without having to relaunch an optimization from the beginning because they are immediately inserted at a good position.

We are currently working on the integration of this algorithm in the optimization system.

2. **Classical optimizations:** Moreover we agentified several classical optimization heuristics to make them work in our multi-agents optimization architecture. We have chosen some intra-route and inter-route heuristics like 2-opt (see figure 7) or 1-1 exchange (see figure 8), to improve solutions obtained by memetic SOM and also explore new solutions by mutating some of them.
3. **Similar approach:** Our approach is similar to that of (Kytjoki & al., 2007) called variable neighbourhood search (VNS) where they create an initial solution by a cheapest insertion heuristic that is improved with a set of improvement heuristic. In a second phase they improve the solution with the same set of heuristic until there are no more improvements. With this approach they can solve very large scale VRP, up to 20,000

customers within reasonable CPU times. But their solution does not address time windows VRP and was not tested on dynamic sets.

We also think that the mixing of artificial intelligence approach with several operations research approaches can give better results than focusing on a unique one. That is why we have chosen such architecture for the optimization part to be able to add different methods and see which ones work well together.

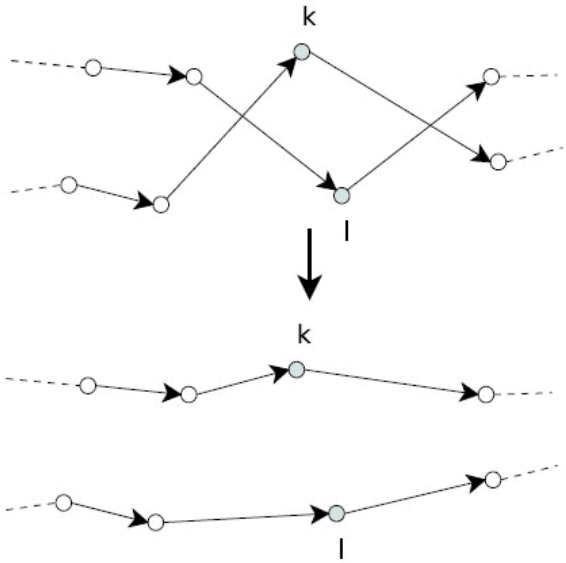


Fig. 8. Example of 1-1 exchange move

5. Experimentation

In this section, we will present an analysis of the trade-off between length optimization and customer waiting time as a function of different degrees of dynamism of the optimization system, and will report results for a benchmark test set for which some already performed experiments exist, even if partials and incomplete. Results reported in the literature and examined in this paper were also obtained considering a medium degree of dynamism, but by modifying the instance by hand, by treating demands with an available time after the half of the day as if they arrived the day before. We prefer in this paper to operate by delaying vehicle starts, in order to report the control of dynamism to the optimization system, rather than to the different ways of managing and using the benchmark test set. In that way, we emphasize to the logical continuity that arises from the dynamic case problem to the static case problem, the latter being a particular case of the former with vehicle delay starts exceeding the working day. In other words, we consider the degree of dynamism as a property of the optimization system, rather than of instances, in order to discriminate algorithms and not the instances.

It is worth noting that at the moment of writing this paper very few approaches to the dynamic VRP were found sharing experiments on a same benchmark. The dynamic problems adopted in this paper are the only set of benchmarks for the dynamic VRP we

have found in the literature on which some metaheuristic approaches are effectively evaluated, that is, the 22 test problems originally proposed by (Kilby & al., 1998).
The proposed memetic SOM was programmed in Java and has been ran on a AMD Athlon 2 GHz computer. All the tests performed with the memetic SOM are done on a basis of 10 runs per instance. For each test case is evaluated the percentage deviation, denoted "%*Length*", to the best known route length, of the mean solution value obtained, *i.e.*

$$\%Length = (\text{mean } Length - Length^*) \times 100 / Length^* \tag{8}$$

where *Length** is the best known value taken from the VRP Web, and "mean Length" is the sample mean based on 10 runs. The average computation times are also reported based on 10 runs. The average customer waiting tine (4) and the maximum vehicle finishing time (5) are expressed as a fraction of the working day in order to compare data with different working days. The waiting time is expressed as a percentage of the working day length *D* by

$$\%WT = \text{mean } WT \times 100 / D, \tag{9}$$

whereas, the maximum finishing time is expressed as an excess deviation to the working day by

$$\%MT = (\text{mean } MT - D) \times 100 / D. \tag{10}$$

While originally, Kilby et al. have set the number of vehicles to 50 for each problem, we prefer to set the number of vehicles according to the overall load of each problem. We think that it looks reasonable to not over-dimension the vehicle resources since it is generally the case in concrete situations that a limited amount of resources are available. Hence the maximum number of vehicles *m* available to perform the tasks for a given problem is set to

$$m = \sum_{i=1,\ldots,N} q(v_i)/Q + 0.1\left(\sum_{i=1,\ldots,N} q(v_i)/Q \right), \tag{11}$$

with $q(v_i)$ the load of demand v_i and Q the vehicle capacity.
This setting also guarantees that it is possible to serve all the demands for the problems considered. Finally, to make things concrete and realistic, the vehicle speed defined in the benchmarks of 1 distance-unit by 1 time-unit can be seen as a vehicle speed of 1 km/mn, or equivalently of 60 km/h. In order to be concrete, we will express the real-time in minutes and the distances in km when reported by their absolute values in some graphics. The working days are roughly between 4 hours to 17 hours, with an exception of a single test case having a 195 hours working day. It is worth noting that the parameter *N* and the total load of the demands are known before optimization in order to adequately dimension the system. Hence, the working day *D* can be decomposed into the many required time-slices. We assume that such values are necessarily known in advance in order to model a concrete real-life situation where a limited number of vehicles are intended to serve a maximum amount of demands, and to reasonably dimension the real-time simulator memory and the optimization system.
We report detailed results of the experiments performed on the (Kilby & al., 1998) benchmarks in Table 1. Here, such results are mainly given in order to allow further comparisons with heuristic algorithms for the dynamic VRP. In table 1, results are presented against the two other approaches found in the literature (Montemanni & al., 2005),

(Goncalves& al., 2007), that have used the benchmark set with a medium degree of dynamism, considering that half of the demands were known in advance. It is worth noting that we simulate the same degree of dynamism by a vehicle delay start time at $D/2$. As we argued along this paper, we consider the degree of dynamism as a property of the system rather than a property of the instance. The first column "Name-size" of the table indicates the name and size of the instance. The second column "D" indicates the working day length, and the third column the best known value obtained for the static problem. Then, results are given within five columns for a given algorithm configuration. The columns "%Length", "%WT", and "%MT" are respectively defined by equations (8), (9), and (10), as the percentage routes length, percentage average customer waiting time, and percentage maximum finishing time. The column "±%CI" is the 95% confidence interval for the routes length. Finally, the column "Sec" reports the computation times in seconds. Two algorithm configurations are considered respectively with fast (To=30ms) and long (To=200ms) computation times. The metaheuristic population size was set to Pop = 10.

When looking at the results of table 1, one should observe the different tradeoffs between route lengths (%*Length*) and waiting times (%*WT*). Then, a medium degree of dynamism will favor the drivers working period to be smaller, but at the expense of the customer waiting time. In the table 1, the approach is compared with an ant colony approach, that is, an adaptation of the well known MACS-VRPTW approach of (Gambardella & al., 1999) that is considered as one of the best performing approaches to the static VRP. The application to the dynamic VRP is due to (Montemanni & al., 2005). Also, it is compared with the genetic algorithm of (Goncalves & al., 2007).

Considering that materials used are quite similar, the memetic SOM yields a better solution quality than the two approaches for less computation time spent. In order to evaluate how the memetic SOM performance behaves as the computation time diminishes, we performed a supplementary set of experiments with a timer-clock at To = 20 ms. The memetic SOM clearly outperforms the ant colony approach in all cases, being roughly an hundred times faster. It also outperforms the genetic algorithm approach being roughly ten times faster. It is worth noting that none of the two approaches report the customer waiting times, this point being a clear drawback of the results presented in the two papers. The authors only claim that the experiments were done with a medium degree of dynamism, half of the demands being considered as known in advance. It is a goal of this paper to be more precise when evaluating a dynamic system, by explicitly considering the tradeoffs between the length and waiting time minimization, as well as the computation time spent.

6. Conclusion

The MERCURE project is helpful for emergency services by giving them appropriate tools to do their job in better conditions. By representing medical emergency services by a Dynamic Vehicle Routing Problem with Time Windows, we are able to optimize human and material resources and so reduce costs, reaction time and maybe save lives.

We have presented the dynamic VRP as a straightforward extension of the classic and standard VRP, and a hybrid heuristic approach to address the problem using a neural network procedure as a search process embedded into a population based evolutionary algorithm, called memetic SOM.

			memetic SOM (fast, start time D/2, Pop=10)					memetic SOM (long, start time D/2, Pop=10)					Montemanni et al. (2005)		Goncalves et al. (2007)	
Name-size-veh	D	Best	%Length	±%CI	Sec[a]	%WT	%MT	%Length	±%CI	Sec[a]	%WT	%MT	%Length	Sec[b]	%Length	Sec[c]
c50	351	524.61	18.58	2.48	16.52	45.49	50.37	17.82	2.14	110.82	46.45	55.19	30.00		13.99	
c75	346	835.26	24.59	2.88	17.00	39.59	56.99	22.65	2.20	106.01	39.53	50.58	24.75		15.89	
c100	399	826.14	14.33	2.68	18.44	42.14	47.69	13.39	2.06	118.60	42.13	46.12	29.03		24.07	
c100b	468	819.56	9.76	4.19	19.49	34.82	33.06	12.88	2.02	127.54	34.92	33.97	24.95		13.60	
c120	794	1042.11	9.47	2.22	29.33	29.26	18.11	7.95	2.62	189.07	29.32	17.12	46.34		37.22	
c150	399	1028.42	32.05	2.20	17.89	36.98	43.23	32.14	1.54	114.52	36.50	41.08	41.58		30.59	
c199	399	1291.29	33.23	1.72	17.22	35.71	37.89	30.29	2.18	113.69	35.35	40.05	42.88		30.18	
f71	211	237	30.89	3.99	9.73	48.76	47.20	27.03	2.94	63.19	49.07	47.01	47.26		19.41	
f134	11741	11620	53.14	8.37	48.38	18.12	18.60	46.98	5.34	318.72	18.59	20.21	38.42		35.29	
tai75a	769	1618.36	19.99	3.02	17.10	31.41	42.08	15.40	2.07	108.87	31.21	39.17	20.18		15.18	
tai75b	905	1344.62	23.93	4.01	18.76	30.75	32.49	22.54	3.69	123.42	31.07	34.08	26.73		13.86	
tai75c	782	1291.01	20.67	2.93	16.72	34.55	36.57	21.93	2.68	108.63	34.70	36.55	28.12		25.64	
tai75d	789	1365.42	17.26	2.94	17.13	33.10	38.71	14.89	3.28	112.53	33.54	40.20	11.98		10.22	
tai100a	897	2041.34	15.98	3.67	40.12	36.97	43.04	13.40	2.21	262.21	37.35	43.81	18.94		18.65	
tai100b	799	1940.61	17.13	2.78	36.35	39.01	45.48	15.19	2.13	239.28	39.05	47.32	20.99		15.48	
tai100c	905	1406.2	23.39	2.99	34.92	25.74	23.38	24.58	3.11	229.39	26.44	24.69	17.76		24.02	
tai100d	782	1581.25	28.21	2.73	33.90	31.78	38.61	24.71	3.68	221.12	31.75	39.09	30.34		20.66	
tai150a	1062	3055.23	19.75	2.52	44.73	34.89	34.71	20.72	3.05	291.58	34.70	35.08	25.69		20.51	
tai150b	988	2656.47	21.36	3.22	38.83	32.79	25.69	20.89	2.38	254.67	32.69	26.81	25.24		24.49	
tai150c	1081	2341.84	21.30	1.44	42.40	27.87	25.40	18.61	2.01	277.16	27.53	26.14	28.79		24.43	
tai150d	1025	2645.39	20.16	3.06	40.44	35.30	26.15	16.22	1.87	269.60	35.20	29.40	21.12		20.55	
tai385	4816	24431.44	38.03	2.21	95.76	29.33	27.03	38.86	1.65	623.00	28.64	27.33	-		-	
Average without tai385			**22.63**	0.63	**27.40**	34.52	36.45	**19.82**	0.53	**179.08**	34.62	36.84	28.62	1500	21.62	1500
Average all			23.33	0.66	30.51	34.29	36.02	20.58	0.51	199.26	34.35	36.41				

[a] Time per run in AMD Athlon (2 GHz) seconds, Java program.
[b] Time per run in Pentium IV (1.5 GHz) seconds, C program.
[c] Time per run in Pentium IV (2.4 GHz) seconds, Java program.

Table 1. Comparative evaluation on the 22 instances of Kilby et al (1998) with medium dynamism

The results given by our simulator look encouraging in that the approach clearly outperforms the few heuristic approaches already applied to the dynamic VRP and evaluated in an empirical way on a common benchmark set. We claim that the memetic SOM is simple to understand and implement, as well as flexible in that it can be applied from a static to a dynamic setting with slight modifications. Also, we think that the memetic SOM is a good candidate for parallel and distributed implementations at different levels, at the level of the population based metaheuristic and at the level of the cellular partition of the plane.
Another interesting aspect of our simulator is that it currently focuses on medical emergency services but it could be extended to address several kinds of emergency services problems.

7. References

Bertsimas, D.; Jaillet, P. & Odoni, A. R. (1990).A priori optimization, in *Operations Research*, vol. 36, no. 6, pp. 1019–1033.

Brotcorne, L .; Laporte, G. & Semet, F. (2003). Ambulance location and relocation models, in *European Journal of Operational research*, 2003, vol. 147, pp. 451–463.

Christofides, N.; Mingozzi, A. & Toth P. (1979). The vehicle routing problem, In: Christofides N et al. editors. *Combinatorial Optimization*, Wiley, pp. 315-338.

Cordeau, J.F.; Gendreau, M.; Hertz, A.; Laporte, G. & Sormany, J.S. (2005). New Heuristics for the Vehicle Routing Problem, In: Langevin A. & Riopel D. (Ed.), *Logistics Systems: Design and Optimization*, Springer, New York, 2005. p. 279-297.

Creput, J.C.; Koukam, A.; Lissajoux, T. & Caminada, A. (2005). Automatic Mesh Generation for Mobile Network Dimensioning using Evolutionary Approach, In: *IEEE Transactions on Evolutionary Computation*, vol. 9, no 1, pp.18-30.

Creput, J.C.; Koukam, A. & Hajjam, A. (2007). Self-organizing maps in evolutionary approach for the vehicle routing problem with time windows, In: *International Journal of Computer Science and Network Security*, vol. 7, no. 1, pp. 103–110.

Creput, J.C. & Koukam, A. (2008). Self-Organization in Evolution for the Solving of Distributed Terrestrial Transportation Problems. In: *Soft Computing applications in industry*, Prasad B., (Ed.), Series : Studies in Fuzziness and Soft-Computing, vol. 226. Springer-Verlag, pp. 189-205.

Gambardella, L.M.; Taillard, E. & Agazzi, G. (1999). MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows, In: , *New Ideas in Optimization*, Corne D, Dorigo M & Glover F editors McGraw-Hill, UK, pp. 63-76.

Gendreau, M.; Laporte, G. & Semet, F. (1997). Solving an ambulance location model by tabu search, In: *Location Science*, vol. 5, no. 2, pp. 75–88.

Gendreau M. & Potvin J-Y (1998). Dynamic vehicle routing and dispatching, In: Crainic TG, Laporte G editors, *Fleet Management and Logistics*, Kluwer, Boston.

Gendreau, M.; Guertin, F.; Potvin, J.Y. & Taillard,E. (1999). Parallel tabu search for real-time vehicle routing and dispatching, *Transportation Science*, vol. 33, no. 4, pp. 381-390.

Gendreau, M.; Laporte, G. & Semet, F. (2001). A dynamic model and parallel tabu search heuristic for real-time ambulance relocation, In: *Parallel Computing*, vol. 27, no. 12, pp. 1641–1653.

Gendreau, M.; Laporte, G. & Potvin, J.Y. (2002). Metaheuristics for the capacitated VRP. In: *The vehicle routing problem*, P. & Vigo, D., (Ed.), Society for Industrial and Applied Mathematics, Philadelphia, PA,, pp 129–154.

Ghaziri, H. (1996). Supervision in the self-organizing feature map: Application to the vehicle routing problem, In: *Meta-Heuristics: Theory & Applications*, I. Osman and J. Kelly, (Ed.), Boston, pp. 651–660.

Ghiani, G.; Guerriero, F.; Laporte, G & Musmanno R. (2003). Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies, In: *European Journal of Operational Research*, 151(1):1-11.

Goncalves, G. ; Hsu, T. ; Dupas, R. & Housroum, H. (2007). Plateforme de simulation pour la gestion dynamique de tournées des véhicules, In : *Journal Européen des Systèmes Automatisés*, 41(5):515-539.

Kilby, P.; Prosser, P. & Shaw P. (1998). Dynamic VRPs: a study of scenarios, *Technical Report APES-06-1998*, University of Strathclyde, UK.

Kohonen, T. (2001). *Self-Organization Maps and associative memory*, 3rd ed., Springer, (Ed.), Berlin, 2001.

Kytöjokia, J.; Nuortiob, T.; Bräysya, O. & Gendreauc, M. (2007). An efficient variable neighbourhood search heuristic for very large scale vehicle routing problems, In: *Computers & Operations Research*, Elsevier Science Ltd, vol. 34, pp. 2743–2757.

Larsen, A. (2000). The Dynamic Vehicle Routing Problem, *PhD thesis*, Technical University of Denmark, Lyngby, Denmark.

Larsen A.; Madsen OBG & Solomon M. (2008). Recent Developments in Dynamic Vehicle Routing Systems, In: *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer, pp. 199-218.

Mester, D. & Bräysy, O. (2005). Active Guided Evolution Strategies for Large Scale Vehicle Routing Problems with Time Windows, In*: Computers & Operations Research*, Elsevier Science Ltd, vol. 32, no 6, pp.1593-1614.

Modares, A.; Somhom, S. & Enkawa, T. (1999). Self-organizing neural network approach for multiple travelling salesman and vehicle routing problems, In: *International Transactions in Operational Research*, vol. 6, pp. 591–606.

Montemanni, R.; Gambardella, L.M.; Rizzoli, A.E. & Donati A.V. (2005). Ant Colony System for a Dynamic Vehicle Routing Problem, In : *Journal of Combinatorial Optimization*, 2005;10.

Moscato, P. & Cotta, C. (2003). A Gentle Introduction to Memetic Algorithms, In: Glover F. & Kochenberger G., (Ed.), *Handbook of Metaheuristics*, Kluwer Academic Publishers, Boston MA, pp. 105-144.

Psaraftis H.N. (1995). Dynamic vehicle routing: Status and prospects, In: *Annals of Operationals Research*, 61:143-164.

Psaraftis H.N. (1998). Dynamic vehicle routing problems, In: Golden B, Assad AA editors, *Vehicle Routing: Methods and Studies*, Elsevier Science, Amsterdam, pp. 223–248.

Rochat, Y. & Taillard, E. D. (1995). Probabilistic diversification and intensification in local search for vehicle routing, In: *Journal of Heuristics*, vol. 1, pp. 147–167.

Toth, P. & Vigo, D. (2001). The vehicle routing problem, Toth, P. & Vigo, D., (Ed.), *Society for Industrial and Applied Mathematics*, Philadelphia, PA, p. 363.

Part 5

The Study of Meteorological, Geomorphological and Remotely Acquired Data

14

A Review of Self-Organizing Map Applications in Meteorology and Oceanography

Yonggang Liu and Robert H. Weisberg
College of Marine Science, University of South Florida
United States of America

1. Introduction

Coupled ocean-atmosphere science steadily advances with increasing information obtained from long-records of in situ observations, multiple-year archives of remotely sensed satellite images, and long time series of numerical model outputs. However, the percentage of data actually used tends to be low, in part because of a lack of efficient and effective analysis tools. For instance, it is estimated that less than 5% of all remotely sensed images are ever viewed by human eyes or actually used (Petrou, 2004). Also, accurately extracting key features and characteristic patterns of variability from a large data set is vital to correctly understanding the interested ocean and atmospheric processes (e.g., Liu & Weisberg, 2005). With the increasing quantity and type of data available in meteorological and oceanographic research there is a need for effective feature extraction methods.

The Self-Organizing Map (SOM), also known as Kohonen Map or Self-Organizing Feature Map, is an unsupervised neural network based on competitive learning (Kohonen, 1988, 2001; Vesanto & Alhoniemi, 2000). It projects high-dimensional input data onto a low dimensional (usually two-dimensional) space. Because it preserves the neighborhood relations of the input data, the SOM is a topology-preserving technique. The machine learning is accomplished by first choosing an output neuron that most closely matches the presented input pattern, then determining a neighborhood of excited neurons around the winner, and finally, updating all of the excited neurons. This process iterates and fine tunes, and it is called self-organizing. The outcome weight vectors of the SOM nodes are reshaped back to have characteristic data patterns. This learning procedure leads to a topologically ordered mapping of the input data. Similar patterns are mapped onto neighboring regions on the map, while dissimilar patterns are located further apart. An illustration of the work flow of an SOM application is given in Fig. 1.

The SOM is widely used as a data mining and visualization method for complex data sets. Thousands of SOM applications were found among various disciplines according to an early survey (Kaski et al., 1998). The rapidly increasing trend of SOM applications was reported in Oja et al. (2002). Nowadays, the SOM is often used as a statistical tool for multivariate analysis, because it is both a projection method that maps high dimensional data to low-dimensional space, and a clustering and classification method that order similar data patterns onto neighboring SOM units. SOM applications are becoming increasingly useful in geosciences (e.g., Liu and Weisberg, 2005), because it has been demonstrated to be an

effective feature extraction technique that has many advantages over conventional data analysis method (e.g., Liu et al. 2006a). The present paper serves as a survey of the SOM applications in meteorology and oceanography community. Recent advance in applications of the SOM in analyzing a variety of data sets in meteorology and oceanography (in situ long time series, remotely sensed satellite and radar data, and numerical model output) are reviewed. The advantages and weaknesses of the SOM are discussed with respect to conventional data analysis methods as used in the community. Suggestions are also given on how to tune the SOM parameters for accurate mapping of meteorological and oceanographic features.

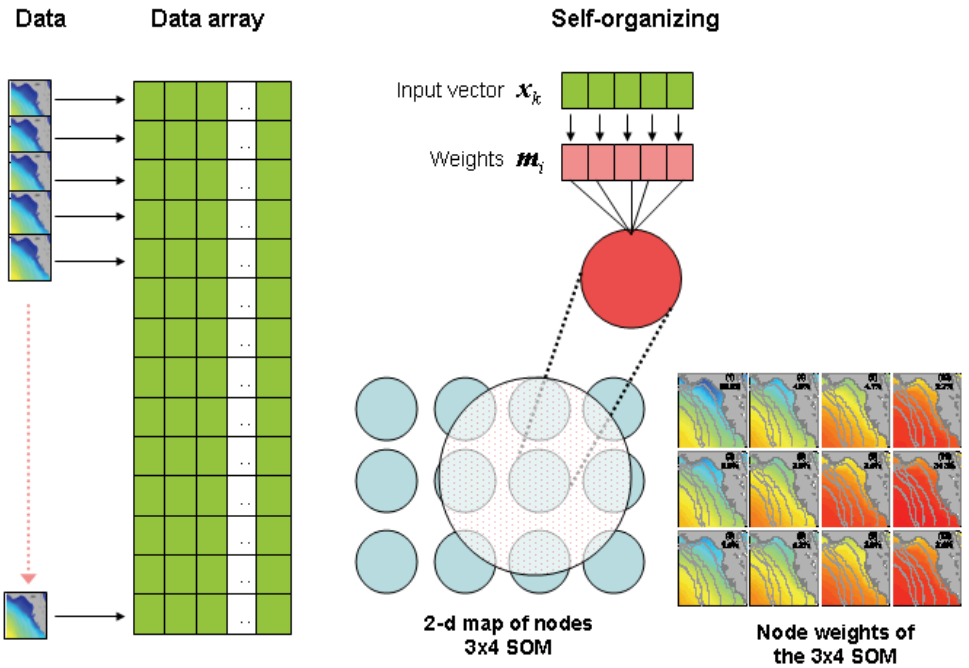


Fig. 1. Illustration of how an SOM works (adapted from Liu et al., 2006b). The data time series are rearranged in a 2D array such that the data at each time step are reshaped as a row vector. For each time step, the row vector is used to update the weight of the SOM via an unsupervised learning algorithm. This iterative process is called self-organizing. The outcome weight vectors of the SOM nodes are reshaped back into characteristic data patterns

2. Self-organizing map applications in meteorology

The SOM was introduced to meteorological and climatic sciences in late 1990s as a clustering and pattern recognition method (e.g., Hewitson & Crane, 1994, 2002; Cavazos, 1999, 2000; Malmgren & Winter, 1999; Ambroise et al., 2000). It is found to be a useful tool in meteorological applications of different spatial and temporal scales: synoptic climatology, extreme weather & rainfall pattern analysis, cloud classification, as well as climate change analysis (Table 1). Many types of meteorological data are analyzed using the SOM, for

example, observed and modeled sea level pressure, geopotential height at different pressure levels, air temperature, humidity, precipitation, evaporation, snow, sea ice, etc. Geographically, the SOM meteorological applications are found around the world: the Americas, Africa, Asia, Europe, Arctic and Antarctic (Table 1). The rest of this section is roughly organized by meteorological data type in SOM applications.

2.1 Sea level pressure and geopotential height data

The SOM is popular in synoptic climatology, especially in analyzing sea level pressure and geopotential height (Table 1). It is often used to summarize and describe the synoptic patterns of atmospheric circulation as indicated by sea level pressure and geopotential height at different levels, and to relate the characteristic circulation patterns with other meteorological variables. For example, Hewitson & Crane (2002) used SOM to describe synoptic atmospheric circulation changes with time as seen from sea level pressure and to relate the sea level pressure patterns with the precipitation time series. Cassano et al. (2006) used the SOM to produce a 55 yr synoptic climatology of daily sea level pressure patterns for the western Arctic, and to study circulation patterns associated with air temperature and high wind extremes. Schuenemann et al. (2009) applied the SOM to the 40-yr European Centre for Medium-Range Weather Forecasts Re-Analysis daily sea level pressure data to objectively identify synoptic sea level pressure patterns over the North Atlantic region. Schuenemann & Cassano (2010a, b) examined the changes of synoptic weather (sea level pressure) patterns from the 15 climate models, and related the SOM extracted circulation patterns with Greenland precipitation in the 20th and 21st centuries. Johnson & Feldstein (2010) presented an SOM analysis that illustrated coupled variability between the North Pacific sea level pressure field and outgoing longwave radiation in the tropical Indo-Pacific region so as to shed light on the relationship between the North Pacific continuum and tropical convection. Reusch et al. (2007) used the SOM to analyze the monthly mean sea level pressure for North Atlantic climate variability. A review of SOM classifications of atmospheric circulation patterns within synoptic climatology is provided in Huth et al. (2008), and an overview in remote sensing applications is seen in Filippi et al. (2010).

2.2 Air temperature, humidity, and wind data

Multiple variables can be simultaneously handled in the SOM algorithm. Thus, the SOM is often used to examine the patterns of co-variability among several meteorological variables. Cavazos (2000) used the SOM to explore the daily atmospheric variables (circulation and humidity) for climate anomalies of extreme precipitation events over the Balkan region. The SOM was used to discover meaningful intraseasonal evolution of North American monsoon from multiple daily atmospheric variables (850 hPa meridional winds, 700 hPa specific humidity, 500 hPa geopotential heights, and 850-500 hPa thickness), and to reveal interaction of the atmospheric variables during the monsoon evolution (Cavazos et al., 2002). The SOM was also used to classify the midtroposphere variables (700 hPa air temperature, geopotential height and specific humidity) for generalized atmospheric patterns, and to reconstruct the ice-core-based synoptic patterns of climate in Antarctic region (Reusch et al., 2005). SOM classification of the meteorological station data is seen in Raju & Kumar (2007), in which multiple variables (temperature, humidity, wind, sunshine hours and solar radiation, etc) are analyzed. Khedairia & Khadir (2008) also performed a classification analysis of meteorological data of Annaba region (North-East of Algeria) from

1995 to 1999 using the SOM and k-means clustering methods. Tambouratzis & Tambouratzis (2008) analyzed long-term (43 years) meteorological data from 128 weather stations in Greece.

Applications & data	Regions	References
Synoptic climatology or synoptic patterns of atmospheric circulation based on SOM analysis of sea level pressure, geopotential height at different pressure levels	Northeast America	Hewitson & Crane (2002), Crane & Hewitson (2003)
	Arctic and adjacent regions	Cassano et al. (2006, 2007, 2010), Higgins & Cassano (2009), Finnis et al. (2009a, b), Skific et al. (2009a, b)
	North Atlantic, Greenland	Reusch et al. (2007), Schuenemann et al. (2009), Schuenemann & Cassano (2010a, b)
	Europe, Middle East & Northern Africa	Michaelides et al. (2007)
	Northern hemisphere, North Pacific	Johnson et al. (2008, 2010), Johnson & Feldstein (2010).
	Australia	Hope et al. (2006), Verdon-Kidd & Kiem (2008), Alexander et al. (2010)
	Antarctic	Lynch et al. (2006), Uotila et a. (2007)
Extreme climate events, monsoon variability, synoptic variation based on various meteorological variables (air temperature, humidity, wind, etc.)	Balkan region	Cavazos (2000)
	Southeast Arizona	Cavazos et al. (2002)
	Europe	Luengo (2004)
	Antarctic	Reusch et al. (2005b)
	India	Raju & Kumar (2007)
	Northeast Algeria	Khedairia & Khadir (2008)
	Greece	Tambouratzis & Tambouratzis (2008)
	Taiwan Island	Chang et al. (2010), Lin & Wu (2010)
Evaporation, precipitation (rainfall, snow, sea ice), cloud classification based on in situ observations, model output, and satellite images	Eastern & central USA	Tian et al. (1999), Filippi et al. (2010)
	Atlantic	Ambroise et al. (2000)
	Europe	Walder & MacLaren (2000)
	Southwest USA	Hsu et al. (2002), Hong et al. (2004, 2005, 2006)
	South Africa	Tadross et al. (2005)
	China	Zhang et al. (2006)
	Japan	Nishiyama et al. (2007)
	Canada (Québec)	Pelletier et al. (2009)
	Taiwan Island	Lin & Chen (2006), Lin & Wu (2007, 2010), Hsu & Li (2010), Chang et al. (2010)
	Colorado River Basin	Fassnacht & Derry (2010)
	Puerto Rico	Malmgren & Winter (1999)
	Peru	Gutierrez et al. (2005)
	Antarctic	Reusch & Alley (2007)

Table 1. SOM applications in meteorology

2.3 Evaporation, precipitation and cloud data

Another category of SOM applications in meteorology include evaporation, precipitation (rainfall & snow) and cloud classification based on in situ observations, model output and satellite images. Many of these applications are also found in the field of hydrology. Malmgren & Winter (1999) used the SOM in climate zonation on the island of Puerto Rico in the Caribbean. They analyzed climate data, seasonal averages of precipitation, and maximum, mean, and minimum temperatures over the years 1960–1990, from 18 stations spread around the island, and identified four climate zones. Hsu et al. (2002) applied the SOM in a rainfall-runoff linear forecast model, called Self-Organizing Linear Output map (SOLO). Tadross et al. (2005) extracted characteristic rainfall patterns over South Africa and Zimbabwe from rainfall data products, and studied the rain-fed maize for the region. Gutierrez et al. (2005) applied the SOM to analyze atmospheric patterns over Peru and local precipitation observations at two nearby stations for the purpose of downscaling multi-model seasonal forecasts. Nishiyama et al. (2007) used the SOM to analyze a combined data set of precipitation and 850 hPa winds, and to identify the typical synoptic wind pattern that frequently causes heavy rainfall in Kyushu during the rainy season. Pelletier et al. (2009) applied the SOM in the characterization of 1-h rainfall temporal patterns in a Québec City case study. Lin & Chen (2006) and Lin & Wu (2007) used the SOM to analyze the rainfall data on Taiwan Island. Recently, Hsu & Li (2010) used the SOM and wavelet methods to explore spatio-temporal characteristics of the 22 years of precipitation data (1982–2003) for Taiwan Island. Chang et al. (2010) also proposed an SOM-based neural network to assess the variability of daily evaporation based on meteorological variables. Recently, the SOM was used to define regions of homogeneity in the Colorado River Basin using snow telemetry snow water equivalent data (Fassnacht & Derry, 2010). The SOM was also used to analyze a 24 year (1973-1996) sea ice data (monthly sea-ice edge positions) in Antartic (Reusch & Alley, 2007).

The SOM is often used as a feature extraction method in cloud classification of satellite imagery. In the pioneering work of the SOM-based cloud classification, Tian et al. (1999) showed the potential of such neural network system in extracting features from the multispectral Geostationary Operational Environmental Satellite (GOES)-8 satellite imagery. Ambroise et al. (2000) presented a probabilistic SOM-based method for segmenting multispectral satellite images, and applied this method in cloud classification of the Polarization and Directionality of the Earth's Reflectances (POLDER) data. Walder & MacLaren (2000) developed an SOM-based automatic cloud classificaton system and applied it to extract spectral and textural features from Advanced Very High Resolution Radiometer (AVHRR) images. Hong et al. (2004) presented a satellite-based rainfall estimation system, Precipitation Estimation from Remotely Sensed Information using Artificial Neural Networks (PERSIANN) cloud classification system, and used this SOM-based system to extract local and regional cloud features from infrared geostationary satellite imagery in estimating fine-scale rainfall distribution. Hong et al. (2005) developed a more accurate SOM-based neural network for cloud patch–based rainfall estimation, named as self-organizing nonlinear output (SONO) model. Hong et al. (2006) further introduced a satellite-based precipitation estimation system using watershed segmentation and growing hierarchical self-organizing map (GHSOM, Rauber et al. 2002), and found significant improvements of estimation accuracy in classifying the clouds into hierarchical sub-layers rather than a single layer.

3. Self-organizing map applications in oceanography

Early SOM applications in oceanography community were mainly limited to satellite and in situ biological/geochemical data analyses by remote sensing scientiests or biological/chemical oceanographers (e.g., Kropp & Klenke, 1997; Ainsworth, 1999; Ainsworth & Jones, 1999; Yacoub et al., 2001; Silulwane et al., 2001). Since the introducion and demonstration of the use of the SOM to the oceanography community by Richardson et al. (2003), SOM applications have been steadily increased in physical oceanography (e.g., Risien et al., 2004; Liu & Weisberg, 2005, 2007; Leloup et al., 2007, 2008; Iskandar et al., 2008), and other disciplinary of oceanography as well (e.g., Chazottes et al., 2006, 2007 ; Telszewski et al., 2009). The SOM is used in analyzing many kinds of oceanographic data, such as satellite ocean color, chlorophyll, sea surface temperature, sea surface height, in situ and modeled ocean currents, etc (Table 2). Geographically, SOM applications are seen in major world's oceans (Pacific, Atlantic, Indian Ocean, Antarctic, etc) and many coastal regions (e.g, Banguela upwelling region, West Florida Shelf, Washington-Oregon Shelf). The rest of this section is organized by oceanographic data type in SOM applications.

3.1 Satellite ocean color and chlorophyll

Satellite oceanography community needed effective feature extraction methods and used the SOM technique earlier because they have larger amount of data than other disciplinary of oceanography. Ainsworth (1999) and Ainsworth & Jones (1999) used the SOM to classify the Chlorophyll concentration data around the Pacific Ocean obtained from the Ocean Colour and Temperature Scanner on board of the Japanese Advanced Earth Observing Satellite (ADEOS), and demonstrated the use of the SOM in classifying ocean colors from multispectral satellite data. Yacoub et al. (2001) applied the SOM in satellite ocean color classifcation for the northwest African coast of the Atlantic Ocean. Niang et al. (2003) proposed an SOM-based automatic classification method to analyze ocean color reflectance measurements taken at the top of the atmosphere (TOA) by satellite-borne sensors, and identified aerosol types and cloud contaminated pixels from satellite ocean color reflectance spectra in the Cape Verde region of the Atlantic Ocean. Recently, Telszewski et al. (2009) applied the SOM to satellite chlorophyll-a concentration, reanalysis sea surface temperature, and mixed layer depth time series and estimated the partial pressure of carbon dioxide (pCO2) distribution in the North Atlantic.

3.2 In situ biological and geochemical data

Kropp & Klenke (1997) were among the earliest SOM users in oceanography. They applied the SOM to a data set of 170 sediment samples for biological and geochemical conditions of a tidal flat in the southern North Sea, and demonstrated the efficiency of the SOM technique in analyzing multivariate data sets of complex natural system (Kropp & Klenke, 1997). Silulwane et al. (2001) used the SOM to classify in situ vertical chlorophyll profiles from the Benguela upwelling system, and related the identified characteristic chlorophyll profiles to pertinent environmental variables, such as sea surface temperature, surface chlorophyll, mixed layer depth and euphotic depth. They pointed out that these relationships can be used semi-quantitatively to predict the subsurface chlorophyll field from known (water column depth) or easily measured variables from satellites, such as surface temperature or surface chlorophyll (Richardson et al., 2002). Lee et al. (2003) used the SOM to examine the plankton taxa in Antarctic area. Chazottes et al. (2006, 2007) applied the SOM to analyze the in situ absorption spectra of phytoplankton from ocean water, in conjunction with detailed

pigment concentrations. Solidoro et al. (2007) used the SOM to classify biogeochemical properties of 1292 water samples collected in a 3-year-long monitoring program in the northern Adriatic Sea, and identified a representative synthetic sample for each group. Bandelj et al. (2008) used the SOM to illustrate the spatial and temporal succession of multitrophic plankton assemblages in the Lagoon of Venice and relates them to biogeochemical properties. Astel et al. (2008) applied the SOM to evaluate the geochemical and environmental impact of 26th December 2004 tsunami disaster in Indian Ocean. Solidoro et al. (2009) applied the SOM to 9 biogeochemical parameters (temperature, salinity, dissolved oxygen, ammonia, nitrites, nitrates, phosphates, silicates, and chlorophyll a) of 7150 original water samples for water mass classification. Aymerich et al. (2009) presented an SOM-based technique for classifying fluorescence spectra, and found that if the data (emission spectra) were appropriately preprocessed, the SOM were able to properly identify between algal groups, such as diatoms and dinoflagellates, which could not be discriminated with previous methods.

3.3 Satellite sea surface temperature data

Remotely sensed sea surface temperature may be the most abundant type of satellite data in oceanography. It is an important variable in air-sea interaction, especially for heat budget. Along with the satellite chlorophyll data analysis, Ainsworth (1999) and Ainsworth & Jones (1999) used the SOM to classify the sea surface temperature around the Pacific Ocean obtained from the Ocean Colour and Temperature Scanner on board of ADEOS satellite. Richardson et al. (2003) gave an example SOM analysis of sea surface temperature in the southern Benguela region. Liu et al. (2006b) used a two-layer GHSOM to analyze the sea surface temperature on the West Florida Shelf in the eastern Gulf of Mexico, and summarized the seasonal evolution of the temperature patterns that were explained in terms of air-sea interactions on the shelf on seasonal time scale. Tozuka et al. (2008) investigated both satellite observed and coupled model outputs of sea surface temperature for tropical Indian Ocean climate variability using the SOM, and found that the SOM successfully captured the dipole sea surface temperature anomaly pattern associated with the Indian Ocean Dipole and basin-wide warming/cooling associated with El Ninõ-Southern Oscillation (ENSO). Morioka et al. (2010) used the SOM to examine the climate variability in the southern Indian Ocean by classifying the sea surface temperature anomaly poles. Iskandar (2010) applied the SOM to study the seasonal and interannual variations of sea surface temperature patterns in Banda Sea. Leloup et al. (2007) used the SOM to analyze the climate indices of equatorial Pacific and found the SOM to be useful both for seasonal ENSO predictability and for the detection of decadal changes in ENSO behavior. Leloup et al. (2008) used the SOM to assess the spatial characteristics of the twentieth century ENSO sea surface temperature variability along the equatorial Pacific simulated by 23 climate models.

3.4 Satellite sea surface height data

Sea surface height from satellite altimetry is another important type of oceanographic data that is related to ocean circulation dynamics and ocean heat content in the upper layer. In open ocean regions, sea surface height is often used to calculate surface geostrophic currents and hence to approximate surface currents. Hardman-Mountford et al. (2003) used the SOM to identify characteristic patterns of satellite derrived sea surface height (actually sea level gradient) data, and related to sardine recruitment in the Northern Benguela. Liu et al. (2008) applied the SOM to time series of altimetry (sea surface height anomaly) gridded data in the

South China Sea, extracted characteristic patterns of sea surface height variability, and calculated the associated surface geostrophic current anomalies. They found that the SOM successfully revealed the upper layer current variability in the South China Sea on seasonal and interannual time scales. Iskandar (2009) examined the satellite altimetry in the tropical Indian Ocean using the SOM, and found that the SOM was able to separate typical patterns associated with the ENSO and the Indian Ocean Dipole events.

3.5 Ocean current data from in situ observations and numerical models

Most of the SOM applications in physical oceanography were to extract characteristic circulation patterns from long time series of ocean current data. Liu & Weisberg (2005) and Liu et al. (2006a) used the SOM to extract the dominant patterns of ocean current variability from a mooed Acoustic Doppler Current Profiler (ADCP) array on the West Florida Shelf, and related the evolution of the characteristic coastal upwelling and downwelling current patterns with the local wind forcing. Liu & Weisberg (2007) examined velocity profiles from an across-shelf transect of ADCP moorings on the West Florida Shelf, and focused on the SOM extracted across-shelf structures of coastal upwelling/downwelling jet over the inner shelf. Cheng & Wilson (2006) also used the SOM to identify the characteristic vertical profiles of the currents in an estuary.

High frequency (HF) radar current data is an important type of data in coastal oceanography that has been developed quickly in recent years. The archived HF radar surface currents are usually hourly maps, i.e., the dimension of the data is high for multiple-year observations. Liu et al. (2007) applied the SOM to extract current pattern variability from a joint HF radar and ADCP dataset on the West Florida Shelf, and obtained dynamically distinctive spatial and temporal current structures on semidiurnal, diurnal and synoptic time scales. Mau et al. (2007) also used the SOM to characterize the Long Island Sound outflows from HF radar data.

Numerical ocean models also generate huge amount of "data" that need to be effectively analyzed. SOM has already found its application in describing numerical ocean model output. For example, Iskandar et al. (2008) applied the SOM to extract the characteristic vertical profiles of zonal currents in the equator of Indian Ocean from a numerical model output. Liu et al. (2009) used the SOM to summarize the synoptic variation of the Columbia River plume patterns (surface currents) from a numerical ocean circulation model. Recently, Jin et al. (2010) also used the SOM to examine modeled currents in Kerama Gap, and gained insights into the interaction of the Kuroshio in the East China Sea and the Ryukyu Current system east of the Ryukyu Islands. Additional opportunities abound for future use of SOM in analyzing numerical ocean model simulations.

3.6 Other oceanographic data

In addition to the above mentioned main data types, SOM applications were found in many other oceanographic data, such as wind stress, sea floor shape, tusnami and salinity. Richardson et al. (2003) and Risien et al. (2004) demostrated the use of SOM in characterizing coastal wind (wind stress) patterns and their variability. Chakraborty et al. (2003) implemented an SOM-based hybrid artificial neural network in sea-floor roughness classification of multibeam angular backscatter data in the central Indian Ocean basin. Liu et al. (2009) applied the SOM to analyze modeled surface salinity time series for characteristic patterns of Columbia River Plume, and associated these plume pattern evolution with local wind forcing and river flow temporal variation. Corchado et al. (2008)

and Mata et al. (2009) applied the SOM-based hybrid intelligent system to detect oil spill in the ocean. Borges et al. (2010) also applied the SOM in geographical classification of weathered crude oil samples. Some SOM applications in maritime environment (e.g., ship trajectory classification) were briefly reviewed in Lobo (2009). Recently, Ehsani & Quiel (2008) and Hentati et al. (2010) applied the SOM to geomorphology.

4. Advantages over other conventional methods

The empirical orthogonal function (EOF) or principal component analysis (PCA) method is often used to extract patterns of variability in meteorological and oceanographic data. Liu & Weisberg (2005, 2007) used both EOF and SOM to extract ocean current patterns from the same data set (a long time series of velocity from a moored ADCP array), and found that the SOM patterns were more accurate and intuitive than the leading mode EOF patterns. The asymmetric features (in current strength, coastal jet location and velocity vector veering with depth) between upwelling and downwelling current patterns extracted by the (nonlinear) SOM were not readily revealed by the (linear) EOF (Liu & Weisberg, 2005). Liu et al. (2006a) evaluated the feature extraction performance of the SOM by using artificial data representative of known patterns. The SOM was shown to extract the patterns of a linear progressive sine wave as the EOF did, even with noise added. However, in the experiment with multiple sets of more complex patterns, the SOM technique successfully chose all those patterns in contrast with the EOF method that failed to do that (Liu et al., 2006a). Reusch et al. (2005) also tested the SOM against the PCA method using synthetic datasets composed of positive and negative modes of four idealized North Atlantic sea level pressure fields, with and without noise components. They also found that the SOM was more robust than the PCA in extracting the predefined patterns of variability. Annas et al. (2007) and Astel et al. (2007) further confirmed the superior performance of the SOM over the PCA. These advantages, of course, must be tempered by the fact that PCA uses an empirical vector space that spans the data space, hence aspects of the data space may be quantitatively reconstructed from the vector space (Liu, 2006; Liu & Weisberg, 2005).

K-means is another popular artificial neural network widely used for clustering. After comparing the SOM and k-means methods, Bação et al. (2005) proposed the use of SOMs as possible substitutes for the k-means clustering algorithms. Lin & Chen (2006) tested the cluster accuracy of the SOM, the k-means method and Ward's method based on experimental data sets that the amount of cluster dispersion and the cluster membership are controlled and known. They found that the SOM determined the cluster membership more accurately than the K-means method and Ward's method. K-means somehow is a subset of SOM, meaning that SOM reduces to k-means for particular choice of parameters (e.g., Lobo, 2009), and therefore it is natural to assume that SOM is more flexible than k-means (Solidoro et al., 2007).

5. Self-organizing map parameter choices

Despite its wide applications as a tool for feature extraction and clustering, the Self-Organizing Map (SOM) remains a black box to most meteorologists and oceanographers. SOM new users may be perplexed by the choice of SOM parameters, because different parameter choices may result in different SOM patterns. This challenge may prevent some potential new users from pursuing further SOM applications. Liu et al. (2006a) evaluated the feature extraction performance of the SOM by using artificial time series data comprised of known patterns. Sensitivity studies were performed to ascertain the effects of the SOM

tunable parameters. A practical way to apply the SOM was proposed and demonstrated using several examples, including long time series of coastal ocean currents from the West Florida Shelf (Liu et al., 2006a).

Oceanographic data	Regions	References
Satellite ocean color, Chlorophyll	Pacific	Ainsworth (1999), Ainsworth & Jones (1999)
	Southeast Atlantic	Yacoub et al. (2001)
	Southwest Atlantic	Saraceno et al. (2006)
	North Atlantic	Niang et al. (2003), Telszewski et al. (2009)
In situ Chlorophyll, absorption spectra of phytoplankton, plankton taxa, ecological variables, microbiological and geochemical variables, pCO2	Southern North Sea	Kropp & Klenke (1997)
	Southeast Atlantic	Silulwane et al. (2001), Richardson et al. (2002)
	Europe	Barreto & Perez-Uribe (2007ab), Alvarez-Guerra et al. (2008), Aymerich et al. (2009), Skwarzec et al. (2009), Žibret & Šajn (2010)
	Lagoon of Venice	Bandelj et al. (2008)
	Northern Adriatic Sea	Solidoro et al. (2007, 2009)
	Antarctic	Lee et al. (2003)
	World oceans	Chazottes et al. (2006, 2007)
	Indian Ocean	Astel et al. (2008)
Satellite measured sea surface temperature, ENSO indices	Pacific	Ainsworth (1999), Ainsworth & Jones (1999)
	Tropical Pacific	Leloup et al. (2007, 2008)
	Southeast Atlantic	Richardson et al. (2003)
	West Florida Shelf	Liu et al. (2006b)
	North Atlantic	Telszewski et al. (2009)
	Indian Ocean	Tozuka et al. (2008), Morioka et al. (2010), Iskandar (2010)
Satellite measured sea surface height	Southeast Atlantic	Hardman-Mountford et al. (2003)
	South China Sea	Liu et al. (2008)
	Indian Ocean	Iskandar (2009)
Ocean currents from in situ observations and numerical models	West Florida Shelf	Liu & Weisberg (2005, 2007), Liu et al. (2006a, 2007)
	Columbia River plume	Liu et al. (2009)
	New York Harbor	Cheng & Wilson (2006)
	New York Bight	Mau et al. (2007)
	Indian Ocean	Iskandar et al. (2008)
	Kerama Gap	Jin et al. (2010)
Surface winds	Southeast Atlantic	Richardson et al. (2003), Risien et al. (2004)
Sea-floor roughness	South Atlantic	Chakraborty et al. (2003)
Salinity	Columbia River plume	Liu et al. (2009)
Oil spill	Galician coast, Europe	Corchado et al. (2008), Mata et al. (2009), Borges et al. (2010)
Maritime data	Europe	Lobo (2009)

Table 2. SOM applications in oceanography

6. Summary

In recent years, the SOM has gained its popularity in meteorology and oceanography community as a powerful pattern recognition and feature extraction method. The SOM analysis has been applied to a variety of data sets in meteorology and oceanography, such as in situ long time series, remotely sensed satellite and radar data, and numerical model output. With the steadily increasing quantity and type of data, the SOM users are expected to increase within the meteorology and oceanography community. Note that there are still many types of meteorological and oceanographic data not analyzed using the SOM, especially output from various numerical models. There are vast opportunities for meteorologists, oceanographers and climate scientists, especially modelers, to have fruitful applications of the SOM, a promising applied mathematical tool for feature extraction and pattern recognition from large and complex data sets.

The SOM has many advantages over conventional feature extraction methods in the community, such as the EOF, k-means methods. It is proposed as a complement to these established methods. One obstacle of SOM application, especially to new users, would be the choice of many tunable parameters, which may prevent potential users from pursuing further SOM applications. Suggestions were given in Liu et al. (2006a) on how to tune the SOM for accurate mapping of meteorological and oceanographic features.

7. Acknowledgements

Support was by the Office of Naval Research, Grants N00014-05-1-0483 and N00014-10-0786 and by NOAA Grant NA06NOS4780246. The first two are for observing and modeling thee West Florida Shelf circulation. The last is for applications to harmful algae under the ECOHAB program. This is CPR contribution #10.

8. References

Ainsworth, E. J. (1999), Visualization of ocean colour and temperature from multispectral imagery captured by the Japanese ADEOS satellite, *Journal of Visualization*, Vol. 2, 195– 204, ISSN 1343-8875

Ainsworth, E. J. & Jones, I. S. F. (1999), Radiance spectra classification from the ocean color and temperature scanner on ADEOS, *IEEE Transaction on Geosciences Remote Sensing,* 37, 1645–1656, ISSN 0196-2892

Alexander, L. V.; Uotila, P., Nicholls, N. &, Lynch, A. (2010). A new daily pressure dataset for Australia and its application to the assessment of changes in synoptic patterns during the last century. *Journal of Climate,* Vol. 23, 1111-1126, ISSN 0894-8755

Alvarez-Guerra, M.; González-Piñuela, C., Andrés, A., Galán, B. & Viguri, J. R. (2008). Assessment of Self-Organizing Map artificial neural networks for the classification of sediment quality, *Environment International,* Vol. 34, No. 6, 782-790, ISSN 0160-4120

Ambroise, C.; Seze, G. Badran F. & Thiria S. (2000), Hierarchical clustering of self-organizing maps for cloud classification, *Neurocomputing,* Vol. 30, 47–52, ISSN 0925-2312

Annas, S.; Kanai T. & Koyama, S. (2007). Principal component analysis and self-organizing map for visualizing and classifying fire risks in forest regions, *Agricultural Information Research*, Vol. 16, No. 2, 44–51, *ISSN*, : 1881-5219

Astel, A.; Tsakouski, S., Barbieri, P. & Simeonov, V. (2007). Comparison of self-organizing maps classification approach with cluster and principal components analysis for large environmental data sets, *Water Research,* Vol. 41, 4566–4578, ISSN 0043-1354

Astel, A.; Boszke, L., Niedzielski, P. & Kozak, L. (2008). Application of the self-organizing mapping in exploration of the environmental impact of a tsunami disaster. *Journal of Environmental Science and Health, Part A: Toxic/Hazardous Substances and Environmental Engineering*, Vol. 43, No. 9, 1016–1026, ISSN 1532-4117

Awad, M. (2010). An unsupervised artificial neural network method for satellite image segmentation, *International Arab Journal of Information Technology,* Vol. 7, No. 2, 199 - 205, ISSN 1683-3198

Aymerich, I. F.; Piera, J., Soria-Frisch, A. & Cros L. (2009). A rapid technique for classifying phytoplankton fluorescence spectra based on self-organizing maps, *Applied Spectroscopy,* Vol. 63 No. 6, 716-726, ISSN 0003-7028

Bação, F.; Lobo, V. & Painho, M. (2005), Self-organizing maps as substitutes for k-means clustering, *Lecture Notes in Computer Science,* V. S. Sunderam, G. v. Albada, P. Sloot, and J. J. Dongarra (Eds.), Vol. 3516, PP. 476-483, ISSN 0302-9743, Berlin Heidelberg, Springer-Verlag

Bandelj, V.; Socal, G., Park, Y.-S., Lek, S., Coppola, J., Camatti, E., Capuzzo, E., Milani L. & Solidoro C. (2008). Analysis of multitrophic plankton assemblages in the Lagoon of Venice, *Marine Ecology Progress Series,* Vol. 368, 23–40, doi: 10.3354/meps07565, ISSN 1616-1599

Barreto, M. A. & Perez-Uribe, A. (2007a). Improving the correlation hunting in a large quantity of SOM component planes, Classification of Agro-ecological variables related with productivity in the sugar cane culture, J. Marques de Sa et al. (Eds.), ICANN 2007, Part II, *Lecture Notes in Computer Science,* Vol. 4669, pp. 379–388, ISSN 0302-9743, Springer-Verlag Berlin Heidelberg

Barreto, M. A. & Perez-Uribe, A. (2007b). Classification of similar productivity zones in the sugar cane culture using clustering of SOM component planes based on the SOM distance matrix, *Proceedings of the 6th International Workshop on Self-Organizing* Map (WSOM 2007), The Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-000-022473-7

Borges, C.; Gómez-Carracedo, M.P., Andrade, J.M., Duarte, M.F., Biscaya, J.L. & Aires-de-Sousa, J. (2010). Geographical classification of weathered crude oil samples with unsupervised self-organizing maps and a consensus criterion, *Chemometrics and Intelligent Laboratory Systems,* Vol. 101, No. 1, 43-55, ISSN 0169-7439

Cassano, E. N.; Lynch, A. H., Cassano, J. J. & Koslow, M. R. (2006). Classification of synoptic patterns in the western Arctic associated with extreme events at Barrow, Alaska, USA, *Climate Research,* Vol. 30, 83-97, ISSN 1616-1572

Cassano, J. J.; Uotila, P., Lynch, A. H. & Cassano, E. N. (2007). Predicted changes in synoptic forcing of net precipitation in large Arctic river basins during the 21st century, *Journal of Geophysical Research*, Vol. 112, G04S49, doi:10.1029/2006JG000332, ISSN 0148-0227

Cassano, N.E. & Cassano, J. J. (2010). Synoptic forcing of precipitation in the Mackenzie and Yukon River basins, *International Journal of Climatology,* Vol. 30, 658–674, ISSN 0899-8418

Cavazos, T. (2000). Using self-organizing maps to investigate extreme climate events: An application to wintertime precipitation in the Balkans, *Journal of Climate,* Vol. 13, 1718–1732, ISSN 0894-8755

Cavazos, T.; Comrie, A. C. & Liverman, D. M. (2002). Intraseasonal variability associated with wet monsoons in southeast Arizona, *Journal of Climate,* Vol. 15, 2477–2490, ISSN 0894-8755

Chakraborty, B.; Kodagali, V. & Baracho, J. (2003). Sea-floor classification using multibeam echo-sounding angular backscatter data: a real-time approach employing hybrid neural network architecture, *IEEE Journal of Oceanic Engineering,* Vol. 28, NO. 1, 121–128, ISSN 0364-9059

Chang, F.-J.; Chang, L.-C. Kao, H.-S. & Wu, G.-R. (2010). Assessing the effort of meteorological variables for evaporation estimation by self-organizing map neural network, *Journal of Hydrology,* Vol. 384, No. 1-2, 118-129, ISSN 0022-1694

Chazottes, A.; Bricaud, A., Crépon, M. & Thiria, S. (2006). Statistical analysis of a database of absorption spectra of phytoplankton and pigment concentrations using self-organizing maps, *Applied Optics,* Vol. 45, No. 31, 8102-8115, ISSN 0003-6935

Chazottes, A.; Crépon, M., Bricaud, A., Ras, J. & Thiria, S. (2007). Statistical analysis of absorption spectra of phytoplankton and of pigment concentrations observed during three POMME cruises using a neural network clustering method, *Applied Optics,* Vol. 46, No. 18, 3790-3799, ISSN 0003-6935

Cheng, P. & Wilson, R. E. (2006). Temporal variability of vertical nontidal circulation pattern in a partially mixed estuary: Comparison of self-organizing map and empirical orthogonal functions, *Journal of Geophysical Research*, Vol. 111, No. C12021, doi:10.1029/2005JC003241, ISSN 0148-0227

Corchado, E.; Baruque, B., Mata, A. & Corchado, J. M. (2008). A WeVoS-CBR approach to oil spill problem, *Lecture Notes in Computer Science,* Vol. 5271, 378-384, DOI: 10.1007/978-3-540-87656-4_47, ISSN *0302-9743*, Springer-Verlag Berlin Heidelberg

Crane, R. G. & Hewitson, B.C. (2003). Clustering and upscaling of station precipitation records to regional patterns using self-organizing maps (SOMs). *Climate Research,* Vol. 25, 95-107, ISSN 1616-1572

Ehsani, A.H. & Quiel, F. (2008). Geomorphometric feature analysis using morphometric parametrization and artificial neural networks, *Geomorphology*, Vol. 99, No. 1-4, 1–12, ISSN 0169-555X

Faro, A.; Giordano, D. & Maiorana, F. (2008). Input noise robustness and sensitivity analysis to improve large datasets clustering by using the GRID, J.-F. Boulicaut, M.R. Berthold, and T. Horváth (Eds.), *DS 2008, Lecture Notes in Artificial Intelligence,* Vol. *5255,* pp. 234–245, ISSN 0302-9743, Springer-Verlag Berlin Heidelberg

Fassnacht, S. R. & Derry, J. E. (2010), Defining similar regions of snow in the Colorado River Basin using self-organizing maps, *Water Resources Research*, Vol. 46, W04507, doi:10.1029/2009WR007835, ISSN 0043-1397

Filippi, A.; Dobreva, I., Klein, A.G. & Jensen, J.R. (2010). Self-Organizing Map-based Applications in Remote Sensing, In: *Self-Organizing Maps,* Matsopoulos, G.K. (Ed.), 231-248, ISBN 978-953-307-074-2

Finnis, J.; Cassano, J. J., Holland, M.M. & Serreze, M.C. (2009a). Synoptically forced hydroclimatology of major Arctic watersheds in general circulation models, Part 1: the Mackenzie River basin. *International Journal of Climatology,* Vol. 29, 1226-1243, ISSN 0899-8418

Finnis, J.; Cassano, J. J., Holland, M.M., Serreze, M.C. & Uotila P. (2009b). Synoptically forced hydroclimatology of major Arctic watersheds in general circulation models, Part 2: Eurasian watersheds. *International Journal of Climatology,* Vol. 29, 1244-1261, ISSN 0899-8418

Gutierrez, J.M.; Cano, R., Cofino, A.S. & Sordo, C. (2005). Analysis and downscaling multi-model seasonal forecasts in Peru using self-organizing maps, *Tellus A*, Vol. 57, No. 3, 435-447, ISSN 0280-6495

Hardman-Mountford, N. J.; Richardson, A. J., Boyer, D. C. , Kreiner, A. & Boyer, H. J. (2003), Relating sardine recruitment in the northern Benguela to satellite-derived sea surface height using a neural network pattern recognition approach, *Progress in Oceanography,* Vol. 59, 241– 255, ISSN 0079-6611

Hentati, A.; Kawamura, A., Amaguchi, H. & Iseri, Y. (2010). Evaluation of sedimentation vulnerability at small hillside reservoirs in the semi-arid region of Tunisia using the Self-Organizing Map, *Geomorphology*, Vol. 122, No. 1-2, 56-64, ISSN 0169-555X

Hewitson, B. C. & Crane, R. G. (1994). Neural Nets: Applications in Geography, ISBN 978-0792327462, Springer, New York

Hewitson, B. C. & Crane, R. G. (2002). Self-organizing maps: applications to synoptic climatology. *Climate Research,* Vol. 22, No. 1, 13-26, ISSN 0936-577X

Higgins, M. E. & Cassano, J. J. (2009). Impacts of reduced sea ice on winter Arctic atmospheric circulation, precipitation, and temperature, *Journal of Geophysical Research*, Vol. 114, D16107, doi:10.1029/2009JD011884, ISSN 0148-0227

Hong, Y.; Chiang, Y. M.; Liu, Y.; Hsu, K. L. & Sorooshian, S. (2006). Satellite-based precipitation estimation using watershed segmentation and growing hierarchical self-organizing map. *International Journal of Remote Sensing,* Vol. 27, No. 23-24, 5165-5184, ISSN 0143-1161

Hong, Y.; Hsu, K., Sorooshian, S. & Gao, X. (2004), Precipitation estimation from remotely sensed imagery using an artificial neural network cloud classification system, *Journal of Applied Meteorology,* Vol. 43, 1834–1853, ISSN 1520-0450

Hong, Y.; Hsu, K., Sorooshian, S. & Gao, X. (2005), Self-organizing nonlinear output (SONO): A neural network suitable for cloud patch-based rainfall estimation at small scales, *Water Resources Research,* Vol. 41, No. W03008, doi:10.1029/2004WR003142, ISSN 0043-1397

Hope, P. K.; Drosdowsky, W. & Nicholls, N. (2006). Shifts in the synoptic systems influencing southwest Western Australia, *Climate Dynamics*, Vol. 26, 751-764, ISSN 0930-7575

Hsu, K.-C. & Li, S.-T. (2010). Clustering spatial–temporal precipitation data using wavelet transform and self-organizing map neural network, *Advances in Water Resources*, Vol. 33, No. 2, 190-200, ISSN 0309-1708

Hsu, K.-C. & Li, S.-T. (2010). Clustering spatial-temporal precipitation data using wavelet transform and self-organizing map neural network, *Advances in Water Resources,* Vol. 33, No. 2, 190-200, ISSN 0309-1708

Hsu, K.-L.; Gupta, H. V., Gao, X., Sorooshian, S. & Imam, B. (2002). Self organizing linear output map (SOLO): An artificial neural network suitable for hydrologic modeling and analysis, *Water Resources Research,* Vol. 38, No. 12, 1302, doi:10.1029/2001WR000795, ISSN 0043-1397

Huth, R.; Beck, C., Philipp, A., Demuzere, M., Ustrnul, Z., Cahynová, M., Kyselý, J. & Tveito, O. E. (2008). Classifications of atmospheric circulation patterns, Recent advances and applications, *Annals of the New York Academy of Sciences*, Vol. 1146, No. 1, 105 - 152, ISSN 0077-8923

Iseri, Y.; Matsuura, T., Iizuka, S., Nishiyama, K. & Jinno, K. (2009). Comparison of pattern extraction capability between self-organizing maps and principal component analysis, *Memoirs of the Faculty of Engineering, Kyushu University,* Vol. 69, No. 2, 37–47.

Iskandar, I. (2009). Variability of satellite-observed sea surface height in the tropical Indian Ocean: comparison of EOF and SOM Analysis, *Makara Seri Sains,* Vol. 13, No. 2, 173-179, ISSN 1693-6671

Iskandar, I. (2010). Seasonal and interannual patterns of sea surface temperature in Banda Sea as revealed by self-organizing map, *Continental Shelf Research*, Vol. 30, 1136–1148, ISSN 0278-4343

Iskandar, I., T. Tozuka, Y. Masumoto, and T. Yamagata (2008), Impact of Indian Ocean dipole on intraseasonal zonal currents at 90°E on the equator as revealed by self-organizing map, *Geophysical Research Letters*, 35, L14S03, doi:10.1029/2008GL033468, ISSN 0094-8276

Jin, B.; Wang, G., Liu, Y. & Zhang, R. (2010). Interaction between the East China Sea Kuroshio and the Ryukyu Current as revealed by the self-organizing map. *Journal of Geophysical Research,* (in press)

Johnson, N. C. & Feldstein, S.B. (2010). The Continuum of North Pacific Sea Level Pressure Patterns: Intraseasonal, Interannual, and Interdecadal Variability. *Journal of Climate, Vol.* 23, 851-867, ISSN 0894-8755

Johnson, N. C.; S. B. Feldstein & B. Tremblay (2008). The continuum of Northern Hemisphere teleconnection patterns and a description of the NAO shift with the use of self-organizing maps, *Journal of Climate*, Vol. 21, 6354–6371, ISSN 0894-8755

Khedairia, S. & Khadir, M. T. (2008). Self-organizing map and k-means for meteorological day type identification for the region of Annaba –Algeria, In: *7th Computer Information Systems and Industrial Management Applications,* pp.91-96, 2008, ISBN: 978-0-7695-3184-7, Ostrava

Kropp, J. & Klenke, T. (1997). Phenomenological pattern recognition in the dynamical structures of tidal sediments from the German Wadden Sea, Ecological Modelling, Vol. 103, No. 2-3, 151-170, ISSN 0304-3800

Kohonen, T. (1988). *Self-Organization and Associative Memory,* Springer-Verlag, ISBN 0-387-18314-0, New York, Berlin, Heidelberg

Kohonen, T. (2001). *Self-Organizing Maps.* Springer-Verlag, ISBN 3-540-67921-9, New York, Berlin, Heidelberg

Lee, W.; Kang, S.-H., Montagna, P.A. & Kwak, I.-S. (2003). Temporal dynamics and patterning of meiofauna community by self-organizing artificial neural networks, *Ocean and Polar Research,* Vol. 25, No. 3, 237-247, ISSN 1598-141X

Leloup, J. A.; Lachkar, Z., Boulanger, J. P. & Thiria, S. (2007). Detecting decadal changes in ENSO using neural networks, *Clim*ate *Dynamics*, Vol. 28, 147-162, doi:10.1007/s00382-006-0173-1, ISSN 0930-7575

Leloup, J. A.; Lengaigne, M. & Boulanger, J. P. (2008). Twentieth century ENSO characteristics in the IPCC database, *Climate Dynamics*, Vol. *30*, 277-291, doi:10.1007/s00382-007-0284-3, ISSN 0930-7575

Lin, G.-F. & Chen, L.-H. (2006). Identification of homogeneous regions for regional frequency analysis using the self-organizing map, *Journal of Hydrology,* Vol. 324, 1-9, ISSN 0022-1694

Lin, G.-F. & Wu, M.C. (2007). A SOM-based approach to estimating design hyetographs of ungauged sites. *Journal of Hydrology*, Vol. 339, No. 3-4, 216-226, ISSN 0022-1694

Lin, G.-F. & Wu, M.-C. (2010). A hybrid neural network model for typhoon-rainfall forecasting, *Journal of Hydrology,* Vol. 375, No. 3-4, 450-458, ISSN 0022-1694

Liu, Y. (2006). *Patterns and Dynamics of Ocean Circulation Variability on the West Florida Shelf.* Ph.D. Dissertation, College of Marine Science, University of South Florida.

Liu, Y. & Weisberg, R. (2005). Patterns of ocean current variability on the West Florida Shelf using the self-organizing map, *Journal of Geophysical Research,* Vol. 110, C06003, doi:10.1029/2004JC002786, ISSN 0148-0227

Liu, Y. & Weisberg, R. H. (2007). Ocean currents and sea surface heights estimated across the West Florida Shelf. *Journal of Physical Oceanography,* Vol. 37, 1697-1713, ISSN 0022-3670

Liu, Y.; Weisberg, R. H. & Mooers, C. N. K. (2006a). Performance evaluation of the self-organizing map for feature extraction, *Journal of Geophysical Research,* Vol. 111, C05018, doi:10.1029/2005JC003117, ISSN 0148-0227

Liu, Y.; Weisberg, R. H. & He, R. (2006b). Sea surface temperature patterns on the West Florida Shelf using the growing hierarchical self-organizing maps. *Journal of Atmospheric and Oceanic Technology,* Vol. No. 2, 23, 325-328, ISSN 0739-0572

Liu, Y.; Weisberg, R. H. & Shay, L. K. (2007). Current patterns on the West Florida Shelf from joint self-organizing map analyses of HF radar and ADCP data. *Journal of Atmospheric and Oceanic Technology,* Vol. 24, No. 4, 702-712, ISSN 0739-0572

Liu, Y.; Weisberg, R. H. & Yuan, Y. (2008). Patterns of upper layer circulation variability in the South China Sea from satellite altimetry using the Self-Organizing Map. *Acta Oceanologica Sinica,* Vol. 27(Supp.), 129-144, ISSN 1869-1099

Liu, Y.; MacCready, P. & Hickey, B. M. (2009). Columbia River plume patterns in summer 2004 as revealed by a hindcast coastal ocean circulation model, *Geophysical Research Letters,* Vol. 36, L02601, doi:10.1029/2008GL036447, ISSN 0094-8276

Lobo, V. J.A.S. (2009). Application of Self-Organizing Maps to the Maritime Environment, In: *Information Fusion and Geographic Information Systems, Lecture Notes in Geoinformation and Cartography,* V.V. Popovich et al, (Eds.), pp.19-36, ISSN 1863-2246, Springer-Verlag, Berlin, Heidelberg

Luengo, F.; Cofiño, A.S. & Gutiérrez, J.M. (2004). GRID Oriented Implementation of Self-organizing Maps for Data Mining in Meteorology, *Lecture Notes in Computer Science,* Vol. 2970, 163-170, DOI: 10.1007/978-3-540-24689-3_21, ISSN 0302-9743, Springer-Verlag Berlin Heidelberg

Lynch, A. H.; Uotila, P. & Cassano, J. J. (2006). Changes in synoptic weather patterns in the polar regions in the 20th and 21st centuries, Part 2: Antarctic, *International Journal of Climatology,* doi:10.1002/joc.1305, ISSN 0899-8418

Malmgren, B. A. & Winter, A. (1999). Climate zonation in Puerto Rico based on principal components analysis and an artificial neural network, *Journal of Climate,* Vol. 12, 977–985, ISSN 0894-8755

Marques, N.C. & Chen N. (2003). Border detection on remote sensing satellite data using self-organizing maps, 11th Portuguese Conference on Artificial Intelligence, Beja, Portugal; Springer, Berlin

Mata, A.; Corchado, E., Baruque, B. (2009). Solving the Oil Spill Problem Using a Combination of CBR and a Summarization of SOM Ensembles, *Advances in Soft Computing,* Vol. 50, 658-662, DOI: 10.1007/978-3-540-85863-8_78, ISSN 1860-0794

Mau, J.-C.; Wang, D.-P., Ullman, D. S. & Codiga, D. L. (2007). Comparison of observed (HF radar, ADCP) and model barotropic tidal currents in the New York Bight and Block Island Sound, *Estuarine, Coastal, and Shelf Science,* Vol. 72, 129–137, ISSN 0272-7714

Michaelides, S.C.; Liassidou, F.& Schizas, C.N. (2007). Synoptic classification and establishment of analogues with artificial neural networks, *Pure and Applied Geophysics,* Vol. 164, No.6-7, 1347-1364, ISSN 1420-9136

Morioka, Y.; Tozuka, T. & Yamagata, T. (2010). Climate variability in the southern Indian Ocean as revealed by self-organizing maps, *Climate Dynamics,* DOI 10.1007/s00382-010-0843-x, ISSN 0930-7575

Niang, A; Gross, L., Thiria, L., Badran, F. & Moulin, C. (2003). Automatic neural classification of ocean colour reflectance spectra at the top of the atmosphere with introduction of expert knowledge, *Remote Sensing of Environment*, Vol. 86, No. 2, 257-271, ISSN 0034-4257, DOI: 10.1016/S0034-4257(03)00113-5

Nishiyama, K.; Endo, S., Jinno, K., Uvo, C.B., Olsson, J. & Berndtsson, R. (2007). Identification of typical synoptic patterns causing heavy rainfall in the rainy season in Japan by a Self-Organizing Map, *Atmospheric Research,* Vol. 83, No. 2-4, 185-200, ISSN 0169-8095

Pelletier, G.; Anctil, F. & Filion, M. (2009). Characterization of 1-h rainfall temporal patterns using a Kohonen neural network: a Québec City case study, *Canadian Journal of Civil Engineering*, Vol. 36, No. 6, 980-990, ISSN 0315-1468

Raju, K. S. & Kumar, D. N. (2007). Classification of Indian meteorological stations using cluster and fuzzy cluster analysis, and Kohonen artificial neural networks, *Nordic Hydrology,* Vol. 38 No. 3, 303–314, ISSN 0029-1277

Rauber, A.; Merkl, D. & Dittenbach, M. (2002). The growing hierarchical self-organizing map: Exploratory analysis of high-dimensional data. *IEEE Transactions on Neural Networks,* Vol. 13, 1331–1341, ISSN 1045-9227

Reusch, D. B.; Alley, R. B. & Hewitson, B. C. (2005b). Towards ice-core-based synoptic reconstructions of west antarctic climate with artificial neural networks. *International Journal of Climatology,* Vol. 25, 581-610, ISSN 0899-8418

Reusch, D. B. & Alley, R.B. (2007). Antarctic sea ice: a self-organizing map-based perspective, *Annals of Glaciology,* Vol. 46, 391-396, ISSN 0260-3055

Reusch, D. B.; Alley, R. B. & Hewitson, B. C. (2005a). Relative performance of self-organizing maps and principal component analysis in pattern extraction from synthetic climatological data, *Polar Geography*, Vol. *29*, 188-212, ISSN 1939-0513

Reusch, D.B.; Alley, R. & Hewitson, B. (2007). North Atlantic climate variability from a self-organizing map perspective, *Journal of Geophysical Research*, Vol. 112, D02104, doi:10.1029/2006JD007460, ISSN 0148-0227

Richardson, A. J.; Risien, C. & Shillington, F. A. (2003). Using self-organizing maps to identify patterns in satellite imagery. *Progress in Oceanography,* Vol. 59, No. 2-3, 223-239, ISSN 0079-6611

Richardson, A. J.; Pfaff, M. C., Field, J. G., Silulwane, N. F. & Shillington, F. A. (2002). Identifying characteristic chlorophyll a profiles in the coastal domain using an artificial neural network, *Journal of Plankton Research,* Vol. 24, 1289– 1303, ISSN 0142-7873

Risien, C. M.; Reason, C. J. C., Shillington, F. A. & Chelton, D. B. (2004). Variability in satellite winds over the Benguela upwelling system during 1999-2000, *Journal of Geophysical Research*, Vol. *109 No.* C3, C03010, doi:10.1029/2003JC001880, ISSN 0148-0227

Saraceno, M.; Provost, C. & Lebbah M. (2006). Biophysical regions identification using an artificial neuronal network: A case study in the South Western Atlantic, *Advances in Space Research,* Vol. 37, 793–805, ISSN 0273-1177

Schuenemann, K. C. & Cassano, J. J. (2009). Changes in synoptic weather patterns and Greenland precipitation in the 20th and 21st centuries: 1. Evaluation of late 20th century simulations from IPCC models, *Journal of Geophysical Research*, Vol. 114, D20113, doi:10.1029/2009JD011705, ISSN 0148-0227

Schuenemann, K. C. & Cassano, J. J. (2010). Changes in synoptic weather patterns and Greenland precipitation in the 20th and 21st centuries: 2. Analysis of 21st century atmospheric changes using self-organizing maps, *Journal of Geophysical Research*, Vol. 115, D05108, doi:10.1029/2009JD011706, ISSN 0148-0227

Schuenemann, K. C.; Cassano, J. J. & Finnis, J. (2009). Forcing of precipitation over Greenland: Synoptic Climatology for 1961– 99, *Journal of Hydrometeorology,* Vol. 10, 60–78, doi:10.1175/2008JHM1014.1, ISSN 1525-7541

Sheridan, S. C. & Lee, C. C. (2010). Synoptic climatology and the general circulation model, *Progress in Physical Geography,* Vol. 34, No. 1, 101-109, ISSN 1477-0296

Silulwane, N. F.; Richardson, A. J., Shillington, F. A. & Mitchell-Innes, B. A. (2001), Identification and classification of vertical chlorophyll patterns in the Benguela upwelling system and Angola-Benguela Front using an artificial neural network, in: A Decade of Namibian Fisheries Science, edited by A. I. L. Payne, S. C. Pillar, and R. J. M. Crawford, *South Africa Journal of Marine Science,* Vol. 23, 37– 51, ISSN 0257-7615

Skific, N.; Francis, J. A. & Cassano, J. J. (2009a). Attribution of seasonal and regional changes in Arctic moisture convergence. *Journal of Climate,* Vol. 22, 5115–5134, DOI: 10.1175/2009JCLI2829.1, ISSN 0894-8755

Skific, N.; Francis, J. A. & Cassano, J. J. (2009b). Attribution of projected changes in atmospheric moisture transport in the Arctic: A self-organizing map perspective. *Journal of Climate,* Vol. 22, 4135–4153, DOI: 10.1175/2009JCLI2645.1, ISSN 0894-8755

Skwarzec, B.; Kabat, K. & Astel, A. (2009). Seasonal and spatial variability of 210Po, 238U and 239+240Pu levels in the river catchment area assessed by application of neural-network based classification, *Journal of Environmental Radioactivity,* Vol. 100, No. 2, 167-175, ISSN 0265-931X

Solidoro, C.; Bastianini, M., Bandelj, V., Codermatz, R., Cossarini, G., Melaku Canu, D., Ravagnan, E., Salon S. & Trevisani S. (2009). Current state, scales of variability & trends of biogeochemical properties in the northern Adriatic Sea, *Journal of Geophysical Research*, Vol. 114, C07S91, doi:10.1029/2008JC004838, ISSN 0148-0227

Solidoro, C.; Bandelj, V., Barbieri, P., Cossarini, G. & Fonda Umani, S. (2007). Understanding dynamic of biogeochemical properties in the northern Adriatic Sea by using self-organizing maps and k-means clustering, *Journal of Geophysical Research*, Vol. 112, C07S90, doi:10.1029/2006JC003553, ISSN 0148-0227

Tadross, M. A.; Hewitson, B. C. & Usman, M. T. (2005). The Interannual variability of the onset of the maize growing season over South Africa and Zimbabwe. *Journal of Climate, Vol*. 18, 3356-3372, ISSN 0894-8755

Tambouratzis, T. & Tambouratzis, G. (2008). Meteorological data analysis using Self-Organizing Maps, *International Journal of Intelligent Systems,* VOL. 23, 735–759. DOI 10.1002/int.20294, ISSN 1740-8873

Telszewski, M.; Chazottes, A., Schuster, U., Watson, A. J., Moulin, C., Bakker, D. C. E., Gonzalez-Davila, M., Johannessen, T., Kortzinger, A., Luger, H., Olsen, A., Omar, A., Padin, X. A., Rios, A. F., Steinhoff, T., Santana-Casiano, M., Wallace, D. W. R. & Wanninkhof, R. (2009). Estimating the monthly pCO2 distribution in the North Atlantic using a self-organizing neural network, *Biogeosciences,* Vol. 6, 1405–1421, ISSN 1726-4170

Tian, B.; Shaikh, M.A., Azimi-Sadjadi, M.R., Vonder Haar T. H. & Reinke, D.L. (1999). A study of cloud classification with neural networks using spectral and textural Features, *IEEE Transactions on Neural Networks*, Vol. 10, No. 1, 138-151, ISSN 1045-9227

Tozuka, T.; Luo, J.-J., Masson S. & Yamagata, T. (2008). Tropical Indian Ocean variability revealed by self-organizing maps, *Climate Dynamics,* Vol. 31, No. 2-3, 333-343, DOI 10.1007/s00382-007-0356-4, ISSN 0930-7575

Uotila, P.; Lynch, A. H., Cassano J. J. & Cullather, R.I. (2007). Changes in Antarctic net precipitation in the 21st century based on Intergovernmental Panel on Climate Change (IPCC) model scenarios, *Journal of Geophysical Research*, Vol. 112, D10107, doi:10.1029/2006JD007482, ISSN 0148-0227

Verdon-Kidd, D. & Kiem, A. S. (2008). On the relationship between large-scale climate modes and regional synoptic patterns that drive Victorian rainfall, *Hydrology Earth System Science Discussions,* Vol. 5, 2791–2815, ISSN 1812-2108

Vesanto, J. & Alhoniemi, E. (2000). Clustering of the self-organizing map. *IEEE Transactions on Neural Networks,* Vol. 11, 586–600, ISSN 1045-9227

Walder, P. & MacLaren, I. (2000). Neural network based methods for cloud classification on AVHRR images, *International Journal of Remote Sensing*, Vol. 21, No. 8, 1693-1708, ISSN 0143-1161

Yacoub, M.; Badran, F. & Thiria, S. (2001). A topological hierarchical clustering: Application to ocean color classification, In *Artificial Neural Networks – ICANN 2001, Lecture Notes in Computer Science*, Vol. 2130, 492 -499, ISSN 0302-9743

Zhang, R.; Wang, Y., Liu, W., Zhu, W. & Wang, J. (2006). Cloud classification based on self-organizing feature map and probabilistic neural network, *Proceedings of the 6th World Congress on Intelligent Control and Automation*, June 21 - 23, 2006, Dalian, China, 41-45, ISSN 0272-1708 (in Chinese)

Žibret, G. & Šajn, R. (2010). Hunting for geochemical associations of elements: factor analysis and self-organising maps, *Mathematical Geosciences,* Vol. 42, No. 6, 681-703, DOI: 10.1007/s11004-010-9288-3, ISSN 1874-8961

15

Using Self Organising Maps in Applied Geomorphology

Ferentinou Maria[1], Karymbalis Efthimios[1],
Charou Eleni[2] and Sakellariou Michael[3]
[1]Harokopio University of Athens,
[2]National Center of Scientific Research 'Demokritos',
[3]National Technical University of Athens
Greece

1. Introduction

Geomorphology is the science that studies landscape evolution, thus stands in the centre of the Earth's surface sciences, where, geology, seismology, hydrology, geochemistry, geomorphology, atmospheric dynamics, biology, human dynamics, interact and develop a dynamic system (Murray, 2009). Usually the relationships between the various factors portraying geo-systems are non linear. Neural networks which make use of non–linear transformation functions can be employed to interpret such systems. Applied geomorphology, for example, adaptive environmental management and natural hazard assessment on a changing globe requires, expanding our understanding of earth surface complex system dynamics. The inherent power of self organizing maps to conserve the complexity of the systems they model and self–organize their internal structure was employed, in order to improve knowledge in the field of landscape development, through characterization of drainage basins landforms and classification of recent depositional landforms such as alluvial fans. The quantitative description and analysis of the geometric characteristics of the landscape is defined as geomorphometry. This field deals also, with the recognition and classification of landforms.

Landforms, according to Bishop & Shroder, (2004) carry two geomorphic meanings. In relation to the present formative processes, a landform acts as a boundary condition that can be dynamically changed by evolving processes. On the other hand formative events of the past are inferred from the recent appearance of the landform and the material it consists of. Therefore the task of geomorphometry is twofold: (1) Quantification of landforms to derive information about past forming processes, and (2) determination of parameters expressing recent evolutionary processes. Basically, geomorphometry aims at extracting surface parameters, and characteristics (drainage network channels, watersheds, planation surfaces, valleys side slopes e.t.c), using a set of numerical measures derived usually from digital elevation models (DEMs), as global digital elevation data, now permit the analysis of even more extensive areas and regions. These measures include slope steepness, profile and plan curvature, cross- sectional curvature as well as minimum and maximum curvature, (Wood, 1996a; Pike, 2000; Fischer et al., 2004). Numerical characterizations are used to quantify

generic landform elements (also called morphometric features), such as point–based features (peaks, pits and passes), line-based features (stream channels, ridges, and crests), and area based features (planar) according to Evans (1972) and Wood, (1996b).

In the past, manual methods have been widely used to classify landforms from DEM, (Hammond, 1964). Hammond's (1964) typology, first automated by Dikau et al., (1991), was modified by Brabyn, (1997) and reprogrammed by Morgan & Lesh, (2005). Bishop & Shroder, (2004) presented a landform classification of Switzerland using Hammond's method. Most recently, Prima et al., (2006) mapped seven terrain types in northeast Honshu, Japan, taking into account four morphometric parameters. Automated terrain analyses based on DEMs are used in geomorphological research and mainly focus on morphometric parameters (Giles & Franklin, 1998; Miliaresis, 2001; Bue & Stepinski, 2006). Landforms as physical constituents of landscape may be extracted from DEMs using various approaches including combination of morphometric parameters subdivided by thresholds (Dikau, 1989; Iwahashi & Pike, 2007), fuzzy logic and unsupervised classification (Irvin et al., 1997; Burrough et al., 2000; Adediran et al., 2004), supervised classification (Brown et al., 1998; Prima et al., 2006), probabilistic clustering algorithms (Stepinski & Collier, 2004), multivariate descriptive statistics (Evans, 1972; Dikau, 1989; Dehn et al.,2001) discriminant analysis (Giles, 1998), and neural networks (Ehsani & Quiel, 2007).

The Kohonen self organizing maps (SOM) (Kohonen, 1995) has been applied as a clustering and projection algorithm of high dimensional data, as well as an alternative tool to classical multivariate statistical techniques. Chang et al., (1998, 2000, 2002) associated well log data with lithofacies, using Kohonen self organizing maps, in order to easily understand the relationships between clusters. The SOM was employed to evaluate water quality (Lee & Scholtz, 2006), to cluster volcanic ash arising from different fragmentation mechanisms (Ersoya et al., 2007), to categorize different sites according to similar sediment quality (Alvarez–Guerra et al., 2008), to assess sediment quality and finally define mortality index on different sampling sites (Tsakovski et al., 2009). SOM was also used for supervised assessment of erosion risk (Barthkowiak & Evelpidou, 2006). Tselentis et al., (2007) used P-wave velocity and Poisson ratio as an input to Kohonen SOM and identified the prominent subsurface lithologies in the region of Rion–Antirion in Greece. Esposito et al., (2008) applied SOM in order to classify the waveforms of the very long period seismic events associated with the explosive activity at the Stromboli volcano. Achurra et al., (2009) applied SOM in order to reveal different geochemical features of Mn-nodules, that could serve as indicators of different paleoceanographic environments. Carniel et al., (2009) describe SOM capability on the identification of the fundamental horizontal vertical spectral ratio frequency of a given site, in order to characterize a mineral deposit. Ferentinou & Sakellariou (2005, 2007) applied SOM in order to rate slope stability controlling variables in natural slopes. Ferentinou et al., (2010) applied SOM to classify marine sediments.

As evidenced by the above list of references, modeling utilizing SOM has recently been applied to a wide variety of geoenvironmental fields, though in the 90s, this approach was mostly used for engineering problems but also for data analysis in system recognition, image analysis, process monitoring, and fault diagnosis. It is also evident that this method has a significant potential.

Alluvial fans are prominent depositional landforms created where steep high power channels enter a zone of reduced stream power and serve as a transitional environment between a degrading upland area and adjacent lowland (Harvey, 1997). Their morphology

resembles a cone segment with concave slopes that typically range from less than 25 degrees at the apex to less than 1 degree at the toe (Figure 1a).

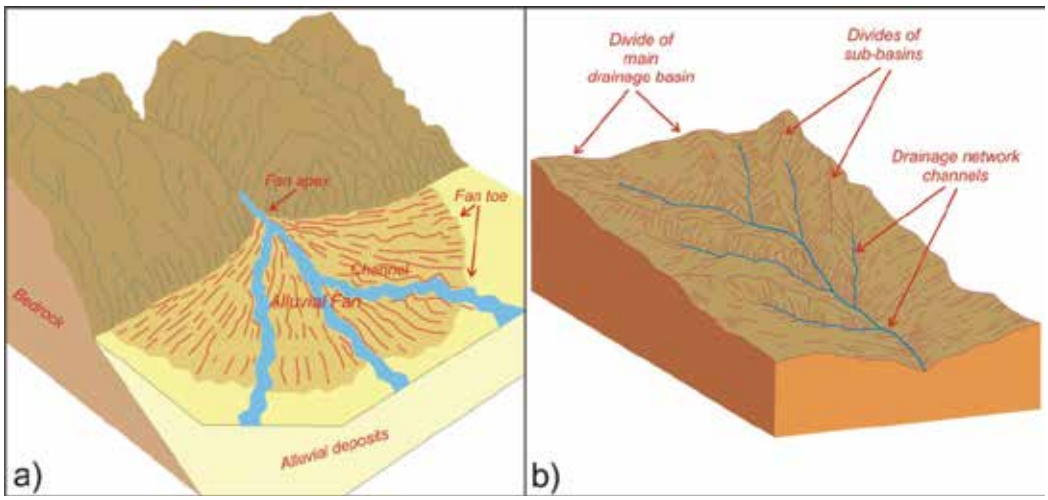


Fig. 1. (a) Schematic representation of a typical alluvial fan, and (b) representation of a typical drainage basin

Alluvial fan characterization is concerned with the determination of the role of the fluvial sediment supply for the evolution of fan deltas. The analysis of the main controlling factors on past and present fan processes is also of major concern in order to distinguish between the two dominant sedimentary processes on alluvial fan formation and evolution: debris flows and stream flows. Crosta & Frattini, (2004), among others, have worked in two dimensional planimetric area used discriminant analysis methods, while Giles, (2010), has applied morphometric parameters in order to characterize fan deltas as a three dimensional sedimentary body. There are studies which have explored on a probabilistic basis the relationships, between fan morphology, and drainage basin geology (Melton, 1965; Kostaschuck et al., 1986; Sorisso-Valvo & Sylvester, 1993; Sorisso-Valvo, 1998). Chang & Chao (2006), used back propagation neural networks for occurrence prediction of debris flows.

In this paper the investigation focuses on two different physiographic features, which are recent depositional landforms (alluvial fans) in a microrelief scale, and older landforms of drainage basin areas in a mesorelief scale (Figure 1b). In both cases landform characterization, is manipulated through the technology of self organising maps (SOMs). Unsupervised and supervised learning artificial neural networks were developed, to map spatial continuum among linebased and surface terrain elements. SOM was also applied as a clustering tool for alluvial fan classification according to dominant formation processes.

2. Method used

2.1 Self organising maps

Kohonen's self-organizing maps (SOM) (Kohonen, 1995), is one of the most popular unsupervised neural networks for clustering and vector quantization. It is also a powerful

visualization tool that can project complex relationships in a high dimensional input space onto a low dimensional (usually 2D grid). It is based on neurobiological establishments that the brain uses for spatial mapping to model complex data structures internally: different sensory inputs (motor, visual, auditory, etc.) are mapped onto corresponding areas of the cerebral cortex in an ordered form, known as topographic map. The principal goal of a SOM is to transform an incoming signal pattern of arbitrary dimension n into a low dimensional discrete map. The SOM network architecture consists of nodes or neurons arranged on 1-D or usually 2-D lattices (Fig. 2). Higher dimensional maps are also possible, but not so common.

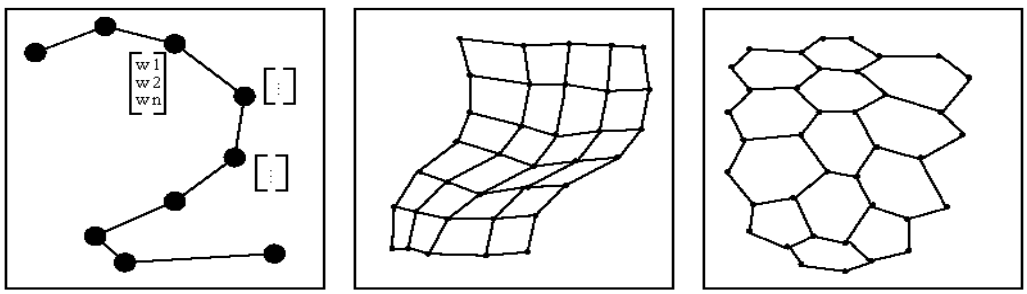


Fig. 2. Examples of 1-D, 2-D Orthogonal and 2-D Hexagonal Lattices

Each neuron has a d dimensional weight vector (prototype or codebook vector) where d is equal to the dimension of the input vectors. The neurons are connected to adjacent neurons by a neighborhood relation, which dictates the topology, or structure, of the map.

The SOM is trained iteratively. In each training step a sample vector x from the input data set is chosen randomly and the distance between x and all the weight vectors of the SOM, is calculated by using an Euclidean distance measure. The neuron with the weight vector which is closest to the input vector x is called the Best Matching Unit (BMU). The distance between x and weight vectors is computed using the equation below:

$$\|x - m_c\| = \min\{\|x_i - m_i\|\} \tag{1}$$

where ||.|| is the distance measure, typically Euclidean distance. After finding the BMU, the weight vectors of the SOM are updated so that the BMU is moved closer to the input vector in the input space. The topological neighbors of the BMU are treated similarly. The update rule for the weight vector of i is

$$x_i(t+1) = m_i(t) + a(t)h_{ci}(t)\left[x(t) - m_i(t)\right] \tag{2}$$

where x(t) is an input vector which is randomly drawn from the input data set, a(t) function is the learning rate and t denotes time. A Gaussian function $h_{ci}(t)$ is the neighborhood kernel around the winner unit m_c, and a decreasing function of the distance between the i_{th} and c_{th} nodes on the map grid. This regression is usually reiterated over the available samples.

All the connection weights are initialized with small random values. A sequence of input patterns (vectors) is randomly presented to the network (neuronal map) and is compared to weights (vectors) "stored" at its node. Where inputs match closest to the node weights, that

area of the map is selectively optimized, and its weights are updated so as to reproduce the input probability distribution as closely as possible. The weights self-organize in the sense that neighboring neurons respond to neighboring inputs (topology which preserves mapping of the input space to the neurons of the map) and tend toward asymptotic values that quantize the input space in an optimal way. Using the Euclidean distance metric, the SOM algorithm performs a Voronoi tessellation of the input space (Kohonen, 1995) and the asymptotic weight vectors can then be considered as a catalogue of prototypes, with each such prototype representing all data from its corresponding Voronoi cell.

2.2 SOM visualization and analysis

The goal of visualization is to present large amounts of information in order to give a qualitative idea of the properties of the data. One of the problems of visualization of multidimensional information is that the number of properties that need to be visualized is higher than the number of usable visual dimensions.

SOM Toolbox (Vesanto, 1999; Vesanto & Alboniemi, 2000), a free function library package for MATLAB, offers a solution to use a number of visualizations linked together so that one can immediately identify the same object from the different visualizations (Buza et al., 1991). When several visualizations are linked in the same manner, scanning through them is very efficient because they are interpreted in a similar way. There is a variety of methods to visualize the SOM. An initial idea of the number of clusters in the SOM as well as their spatial relationships is usually acquired through visual inspection of the map. The most widely used methods for visualizing the cluster structure of the SOM are distance matrix techniques, especially the unified distance matrix (U-matrix). The U-matrix visualizes distances between prototype vectors and neighboring map units and thus shows the cluster structure of the map. Samples within the same unit will be the most similar according to the variables considered, while samples very different from each other are expected to be distant in the map. The visualization of the component planes help to explain the results of the training. Each component plane shows the values of one variable in each map unit. Simple inspection of the component layers provides an insight to the distribution of the values of the variables. Comparing component planes one can reveal correlations between variables.

Another visualization method offered by SOM is displaying the number of hits in each map unit. Training of the SOM, positions interpolating map units between clusters and thus obscures cluster borders. The Voronoi sets of such map units have very few samples (“hits”) or may even be empty. This information is utilized in clustering the SOM by using zero-hit units to indicate cluster borders.

The most informative visualizations of all offered by SOM are simple scatter plots and histograms of all variables. Original data points (dots) are plot in the upper triangle, though map prototype values (net) are plot on the lower triangle. Histograms of main parameters are plot on the diagonal. These visualizations reveal quite a lot of information, distributions of single and pairs of variables both in the data (upper triangle) and in the trained map (lower triangle). They visualize the parameters in pairs in order to enhance their correlations. A scatter diagram can extend this notion to the multiple pairs of variables.

3. Study area

The case study area is located on the northwestern part of the tectonically active Gulf of Corinth which is an asymmetric graben in central Greece trending NW-SE across the Hellenic mountain range, approximately perpendicular to the structure of Hellenides (Brooks & Ferentinos, 1984; Armijo et al., 1996). The western part of the gulf, where the study area is located, is presently the most active with geodetic extension rates reaching up to 14-16 mm/yr (Briole et al., 2000). The main depositional landforms along this part of the gulf's coastline are coastal alluvial fans (also named fan deltas) which have developed in front of the mouths of fourteen mountainous streams and torrents. Alluvial fan development within the study area is the result of the combination of suitable conditions for fan delta formation during the Late Holocene. Their evolution and geomorphological configuration is affected by the tectonic regime of the area (expressed mainly by submergence during the Quaternary), weathering and erosional surface processes throughout the corresponding drainage basins, mass movement (especially debris flows), and the stabilization of the eustatic sea-level rise about 6,000 years ago (Lambeck, 1996).

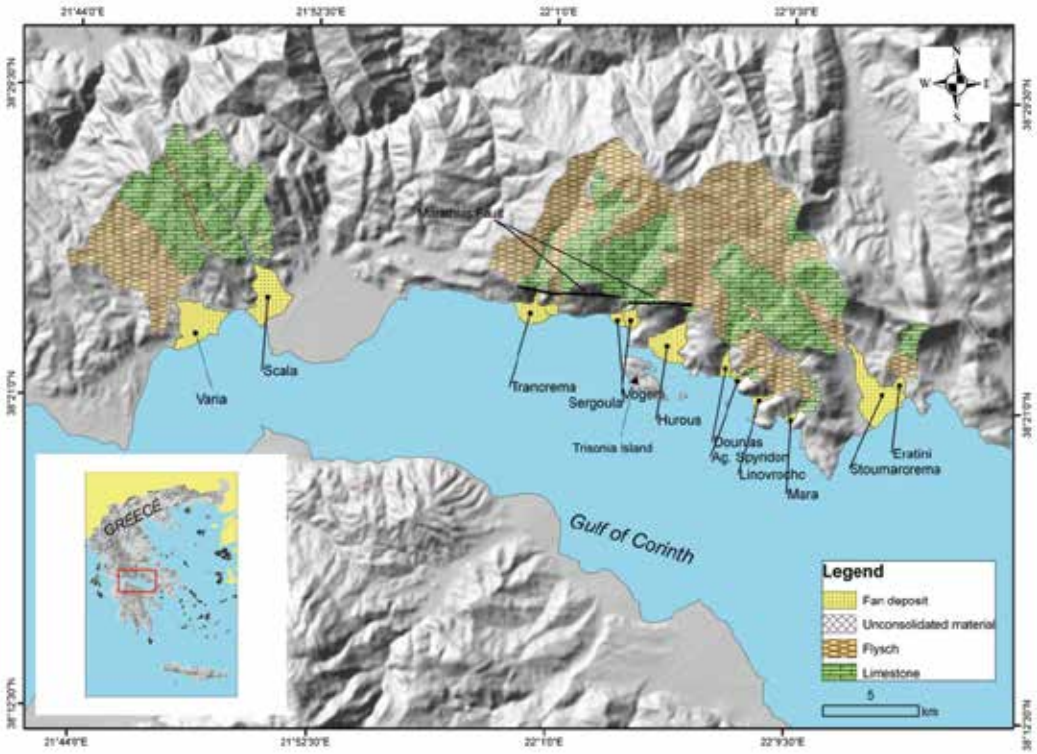


Fig. 3. Simplified lithological map of the study area

Apart from the classification of microscale landforms, such as the above mentioned coastal alluvial fans, this study also focuses on mesoscale landforms characterization. This attempt concerns the hydrological basin areas of the streams of (from west to east) Varia, Skala, Tranorema, Marathias, Sergoula, Vogeri, Hurous, Douvias, Gorgorema, Ag. Spiridon, Linovrocho, Mara, Stournarorema and Eratini, focusing on the catchments of Varia and

Skala streams. Landforms distribution within the studied drainage basins are mainly controlled by the bedrock lithology. Therefore, it is important to outline the geology of the area. The basic structural pattern of the broader area of the drainage basins was established during the Alpine folding. The drainage basins are dominated by geological formations of the geotectonic zones of Parnassos–Ghiona, Olonos-Pindos and Ionian and the Transitional zone between those of Parnassos–Ghiona and Olonos-Pindos. The easternmost basins (Eratini and part of Stournarorema) are made up of Tithonian to Senonian limestones of the Parnassos–Ghiona zone and the Transitional Sedimentary Series (limestones of Upper Triassic to Paleocene age and sandstones and shales of the Paleocene–Eocene flysch). The majority of the catchments consist of the Olonos–Pindos zone formations which are represented by platy limestones of Jurassic-Senonian age and Upper Cretaceous - Eocene flysch lithological sequences (mainly sandstones and shales). Part of the westernmost Varia drainage system drains flysch formations (mainly marls, sandstones and conglomerates) of the Ionian zone. A simplified lithological map of the catchments is presented in Fig3. Tectonically the area is affected by an older NW-SE trending fault system, contemporaneous to the Alpine folding and a younger one having an almost E-W direction with the active normal fault of Marathias (Gallousi & Koukouvelas, 2007) and normal faults located in the broader area of Trizonia Island being the most significant.

4. Application of SOM in landform characterization - Input variables and data preparation

This research is based on quantitative and qualitative data depicting the morphology and morphometry of fans and their drainage basins. These data derived from field-work, SRTM DEM data and topographic and geological maps at various scales. The correlation between geomorphological features (expressed by morphometric parameters) of the drainage basins and features of their fan deltas was detected, in order to determine the role of the fluvial sediment supply for the evolution of the fan deltas.

A simplified lithological map of the area was constructed from the geological maps of Greece at the scale of 1:50,000 obtained from the Institute of Geology and Mineral Exploration of Greece (I.G.M.E.). The lithological units cropping out in the basins area were grouped in three categories including limestones, flysch formations (sandstones, shales and conglomerates) and unconsolidated sediments. The area cover occupied from each one of the three main lithological types in the area of each basin was also estimated.

The identification and delineation of the fans was based upon field observations, aerial photo interpretation and geological maps of the surficial geology of the area at the scale of 1:50,000 (Paraschoudis, 1977; Loftus & Tsoflias, 1971). Detailed topographic diagrams at the scale of 1:5.000, were used for the calculation of the morphometric parameters of the fan deltas. All topographic maps were obtained from the Hellenic Military Geographical Service (H.M.G.S). The elevation of the fan apex was measured by altimeter or GPS for most of the studied fans. All measurements and calculations of the morphometric parameters were performed using Geographical Information System (GIS) functions. The morphometric variables obtained for each fan and its corresponding drainage basin are described in Table 1.

Table 2 presents the values of the (fifteen) morhometric parameters measured and estimated for the coastal alluvial fans and their drainage basins.

Drainage basin morphometric parameters			
	Morphometric Parameter	Symbol	Explanation
1	Drainage basin area	(A_b)	The total planimetric area of the basin above the fan apex, measured in km^2.
2	Basin crest	(C_b)	The maximum elevation of the drainage basin given in m.
3	Perimeter of the drainage basin	(P_b)	The length of the basin border measured in km.
4	Total length of the channels within the drainage basin	(L_c)	Measured in km.
5	Total length of 20 m contour lines within the drainage basin	(ΣL_c)	Measured in km.
6	Basin relief	(R_b)	Corresponds to the vertical difference between the basin crest and the elevation of fan apex, given in m.
7	Melton' s ruggedness number	(M)	An index of basin ruggedness (Melton, 1965, Church and Mark, 1980) calculated by the following formula: $M=R_bA_b^{-0.5}$
8	Drainage basin slope	(S_b)	Obtained using the following equation : $S_b=e\Sigma L_c/A_b$ e is the equidistance (20m for the maps that were used in this study).
9	Drainage basin circularity	(Cir_b)	It is given by the equation: $Cir_b=4\pi A_b/P_b^2$ and expresses the shape of the basin.
10	Drainage basin density	(D_b)	The ratio of the total length of the channels to the total area of the basin.
Fan delta morphometric parameters			
11	Fan area	(A_f)	The total planimetric area of each fan, measured in km^2.
12	Fan length	(L_f)	The distance between the toe (coastline for most of the fans) and apex of the fan, measured in m.
13	Fan apex	(Ap_f)	The elevation of the apex of the fan in m.
14	Fan slope	(S_f)	The mean gradient measured along the axial part of the fan.
15	Fan concavity	(C_f)	An index of concavity along the fan axis defined as the ratio of a to b, where a is the elevation difference between the fan axis profile and the midpoint of the straight line joining the fan apex and toe, and b is the elevation difference between the fan toe and midpoint.

Table 1. Definition of drainage and fan delta morphometric parameters

	Stream/fan name	A_b	C_b	P_b	L_c	ΣL_c	R_b	M	S_b	Cir_b	D_b	A_f	L_f	Ap_f	S_f	C_f
1	Varia	27.5	1420	26.5	85.9	592.2	1376	0.26	0.43	0.49	3.13	4.2	2.6	44	0.017	1.10
2	Skala	28.2	1469	25.6	80.6	785.1	1375	0.26	0.56	0.54	2.86	4.2	2.9	94	0.033	1.29
3	Tranorema	30.3	1540	26.4	112.4	798.7	1452	0.26	0.53	0.55	3.70	1.6	2.1	88	0.042	1.05
4	Marathias	2.3	880	6.8	6.6	52.8	788	0.52	0.46	0.63	2.87	0.4	0.6	92	0.157	1.28
5	Sergoula	18.4	1510	19.7	59.7	569.8	1456	0.34	0.62	0.60	3.24	0.5	1.2	54	0.046	1.16
6	Vogeni	2.4	1035	7.9	5.6	63.7	817	0.53	0.53	0.49	2.34	0.7	1.3	218	0.167	1.38
7	Hurous	6.8	1270	11.6	23.2	158.6	1054	0.41	0.47	0.63	3.43	2.7	2.8	216	0.077	1.63
8	Douvias	6.8	1361	10.6	23.6	190.3	1269	0.49	0.56	0.77	3.46	0.6	1.6	92	0.059	1.42
9	Gorgorema	2.5	1060	7.3	6.2	67.7	1012	0.64	0.55	0.59	2.52	0.1	0.6	48	0.082	1.18
10	Ag. Spiridon	1.0	585	4.4	3.5	32.2	515	0.50	0.62	0.69	3.39	0.1	0.7	70	0.095	1.33
11	Linovrocho	3.6	1020	8.6	11.3	86.4	926	0.49	0.47	0.62	3.09	0.3	1.2	94	0.080	1.04
12	Mara	2.1	711	6.8	7.8	51.4	651	0.45	0.50	0.57	3.76	0.2	0.8	60	0.076	1.14
13	Stournaro-rema	47.1	1360	31.5	142.1	1236.0	1268	0.18	0.53	0.60	3.02	4.7	4.5	92	0.021	1.56
14	Eratini	3.4	1004	8.8	8.6	77.7	974	0.53	0.46	0.55	2.55	0.3	0.7	30	0.044	1.30

Table 2. Values of the measured morphometric parameters for the 14 alluvial fans and their drainage basins

Two more qualitative parameters were studied, the existence or not of a well developed channel in fan area (R), and the geological formation that prevails in the basin area (GEO). Channel occurrence or absence was coded in a binary condition, whereas geological formation prevalence was coded according to relative erodibility.

Nr	Stream/fan name	GEO	R	Nr	Stream/fan name	GEO	R
1	Varia	flysch	1	8	Douvias	limestone	1
2	Skala	limestone	1	9	Gorgorema	flysch	1
3	Tranorema	flysch	0	10	Ag. Spiridon	flysch	0
4	Marathias	limestone	1	11	Linovrocho	flysch	1
5	Sergoula	limestone	0	12	Mara	flysch	1
6	Vogeni	limestone	0	13	Stournarorema	flysch	1
7	Hurous	flysch	1	14	Eratini	limestone	0

Table 3. Values of the studied categorical parameters for the 14 alluvial fans and their drainage basins

Satellite derived DEMs were also used for digital representation of the surface elevation. The source were global elevation data sets from the Shuttle Radar Topography Mission (SRTM)/SIR-C band data, (with 1 arc second and 3 arc seconds) released from (NASA). In this study, two DEMs were re-projected to Universal Transverse Mercator (UTM) grid, Datum WGS84, with 250m and 90m spacing. In the proposed semi-automatic method, it is necessary to implement algorithms, which identify landforms from quantitative, numerical attributes of topography. Morphometric analysis of the study area was performed using the DEM and the first and second derivatives (slope, aspect, curvature, plan and profile curvature), applying Zevenbergen & Thorne (1987) method. Morphometric feature analysis and extraction of morphometric parameters are implemented in the open source SAGA GIS software, version 2.0 (SAGA development team 2004). Routines were applied in order to perform terrain analysis and produce terrain forms using Peuker & Douglas (1975), method. This method considers the slope gradients to all lower and higher neighbors for the cell being processed. For example, if all the surrounding neighbor cells have higher elevations than the cell being processed, the cell is a pit and vice versa is a peak. If half of the surrounding cells are lower in elevation and half are higher in elevation, then the cell being processed is on a hill-slope. The cell being processed is identified as a ridge cell if only one of the neighboring cells is higher, and, conversely, a channel when only one neighbor cell is lower. When slope gradients are considered, a hill-slope cell can be further characterized between a convex or concave hill-slope position. At locations with positive values for slope, channels have negative cross sectional curvature whereas ridges have positive cross sectional curvatures. The differentiation to plan hill-slopes is performed by using a threshold.

Symbol	Description	Nr of data samples in 250m DEM spacing of the whole data set	Nr of data samples in 90m DEM spacing of the subset of Varia and Scala basin
-9	Pit	26	113
-7	Channel	825	6,322
-2	Concave break form valleys	683	5,284
0	Flat	99	1,060
1	Pass	4	371
2	Convex break form ridges	713	5,441
7	Ridge	805	6,289
9	Peak	17	138

Table 4. Terrain form classification according to Peuker & Douglas method

Sampling procedure for the data set describing the drainage basins and alluvial fan regions, was performed. A sampling function was applied to the derivatives grids in order to prepare a matrix of sample vectors. The produced ASCII file was exported to MATLAB in order to use SOM artificial neural networks. The main geomorphological elements according to Peuker and Douglas (1975) method, are channels, ridges, convex breaks and concave breaks and are presented in Table 4. Pits, peaks and passes are not so often in the study area. The morphometric parameters derived were used as input to SOM. Data preparation in general is a diverse and difficult issue. It aims to, select variables and data

sets to be used for building the model, clean erroneous or uninteresting values from the data. It also aims to transform the data into a format which the modelling tool can best utilize and finally normalize the values in order to accomplish a unique scale and avoid problems of parameter prevalence according to their high values.
The quality of the SOM obtained with each normalization method is evaluated using two measures as criteria: the quantization error (QE) and the topographic error (TE). QE is the average distance between each data set data vector and its best mapping unit, and thus, measures map resolution (Kohonen, 1995). TE is used as a measure of topology preservation. The map size is also important in the SOM model. If the map is too small, it might not explain some important differences, but if the map is too large (i.e. the number of map units is larger than the number of samples), the SOM can be over fitted (Lee & Scholz, 2006). Under the condition that the number of neurons could be close to the number of the samples, the map size was selected, for each application.

5. Results

5.1 Microscale landform characterization (coastal alluvial fan classification)

The application of the SOM algorithm in the current data set, and the result of the clustering are presented through the multiple visualization in Fig.4. The examined variables are the morphometric parameters of the alluvial fans and their corresponding drainage basins, analytically presented in Tables 1 and 2. The lowest values of QE and TE were obtained using logistic function which scales all possible values between [0,1]. Batch training took place in two phases. The initial phase is a robust one and then a second one is fine-tuning with a smaller neighborhood radius and smaller (learning rate). During rough initial neighborhood radius and learning rate were large. Gradually the learning rate decreased and was set to 0.1, and radius was set to 0.5.
Visualization in Fig. 4 consists of 19 hexagonal grids (the U-matrix upper left, along with the 17 component layers and a label map on the lower right). The first map on the upper left gives a general picture of the cluster tendency of the data set. Warm colors represent the boundaries of the clusters, though cold colors represent clusters themselves. In this matrix four clusters are recognized. In Fig.5a and Fig.5c the same vislualization is presented through hit numbers in Fig. 5a and the post–it labels in Fig. 5c. The hit numbers in the polygons represent the record number, of the data set that belong to the same neighborhood (cluster). Through the visual inspection of both Fig.5a and Fig.5c, one corresponds the hit numbers to the particular record, which is the alluvial fan name. Four clusters were generated. The records that belong to the same cluster are mapped closer and have the same color. For example, Marathias and Vogeni belong to the same cluster represented with blue. The common characteristics of these two fans are visualized through Fig. 4. Using similarity coloring and position, one can scan through all the parameters and reveal that these two records mapped in the upper corner of each parameter map have always the same values represented by similar color.
Except from general clustering tendency, scanning through parameter layers one can reveal correlation schemes, always following similarity colouring and position. Each parameter map is accompanied with a legend bar that represents the range values of the particular parameter. Drainage basin area (A_b) is correlated with fan area (A_{pf}) and fan length (L_f).

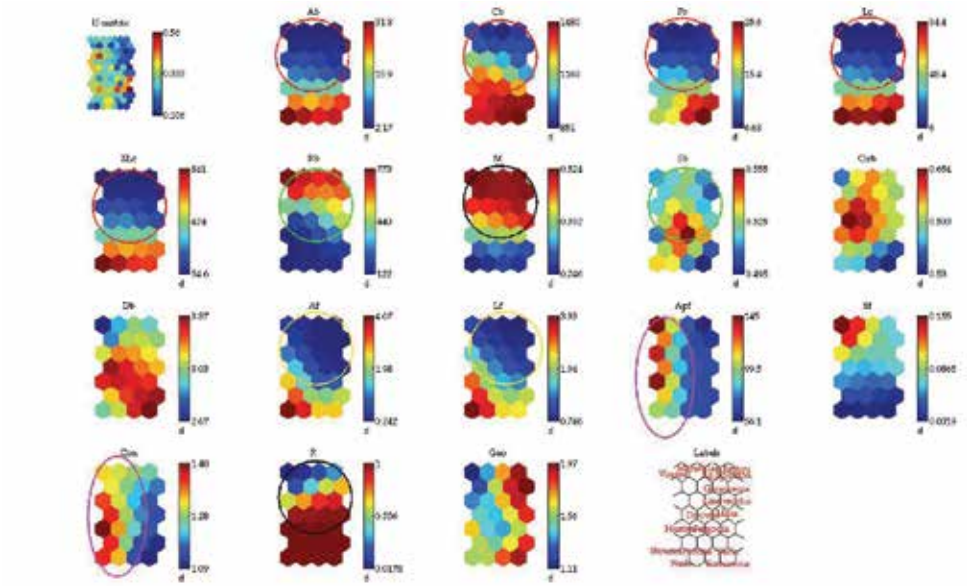


Fig. 4. SOM visualization through U-matrix (top left), and 17 component planes, one for each parameter examined. The figures are linked by position: in each figure, the hexagon in a certain position corresponds to the same map unit

Total length of channels (L_c) within basin area (A_b), and total length of contours (ΣL_c) within the drainage basin are also correlated (see red and yellow circles in Fig. 4). Basin crest (C_b), and basin relief (R_b) are inversely correlated (see green circle in Fig. 4). Melton's' ruggedness number is inversely correlated to fan slope (S_f), but correlated to channel development in fan area (see black circles in Fig. 4). The geological formation prevailing to basin area seems to be inversely correlated to concavity, (i.e. limestone basins have produced less concave fans compared to the flysch ones). Concavity (C_f) is also correlated to fan area (A_{pf}).

Analysis of each cluster is then carried out to extract rules that best describe each cluster by comparing with component layers. The rules to model and predict the generation of alluvial fans, are extracted by mapping the four clusters presented in Fig.5 with the input morphometric parameters (component planes) in Fig.4. Prior to rules extraction each input variable is divided in three categories, that is low and high and medium. The threshold value, which separates each category, is determined from the component planes legend bar in Fig.4.

In the following description, the response of the given data to the map (adding hits number) for each cluster was calculated as a cluster index value (CIV). The higher the cluster index value the stronger the cluster and therefore the most important in the data set and the most representative for the study area.

Cluster 1: Varia, Skala, Sergoula, Stournarorema, Tranorema. The cluster index was calculated (5). Varia and Tranorema form a subgroup. Stournarorema and Scala form a second subgroup. This group includes fans formed by streams with well developed drainage networks and large basins with high values of basin relief. The produced fans are extensively and relatively gently sloping (with a mean slope of 0.03). Varia, Skala, Sergoula and Stournarorema fans have a triangular shape and resemble small deltas while Tranorema has a more semicircular morphology. These fans are intersected by well developed and clearly defined distributary channels consisting of coarse grained material (pebbles, cobles

and few boulders). These are generally aggrading fans with an active prograding area near the river mouth. The fans of this group are characterized as fluvial dominated.

Cluster 2: Marathias, Vogeni. The cluster index is (2). This second group involves fans formed by torrents with small drainage basins. They have developed laterally overlaying or confining fans of the cluster 1. Their shape is conical, they do not present well developed channels and are also characterized from high fan gradients (mean fan slope reaches 0.4). Flysch formations prevail in their basin area. According to these features, they seem to be debris flow dominated. Their formation and evolution is inferred to be highly governed from the two serious landslides of Marathias and Sergoula, occurred in the study area.

Cluster 3: Gorgorema, Mara, Linovrocho, Ag. Spyridon, Eratini. The cluster index is (5). This group includes alluvial fans formed by streams of well developed drainage networks with large basins dominated by the presence of flysch formations. The fans are elongated and have well developed and clearly defined distributary channels, relatively incised in the most proximal part of the fan, near the apex, which become indefinite at the lower part near the coastline. The slope of their surface (mean gradient of 0.08) is higher than the slope of the cluster 1 fans and lower than those of cluster 2. According to these findings they are characterized as fluvial dominated with debris flow influences.

Cluster 4: Hurus and Douvias. The cluster index value is (2). The drainage basins of these two streams have similar features. These two fans are elongated and have well developed distributary channels, low slope values and high concavity. Their main characteristic is the large fan area if compared with the catchment area. The anomalously large Hurus torrent alluvial fan in relation to its drainage basin area is interpreted to be the result of abnormally high sediment accumulation at the mouth of this torrent. This exceptional accumulation rate is attributed to reduce of marine processes effectiveness due to the presence of Trisonia Island in front of the torrent mouth. This island protects the area of the fan resulting in deposition of the fluvio-torrential material. They are characterised as fluvial dominated fans.

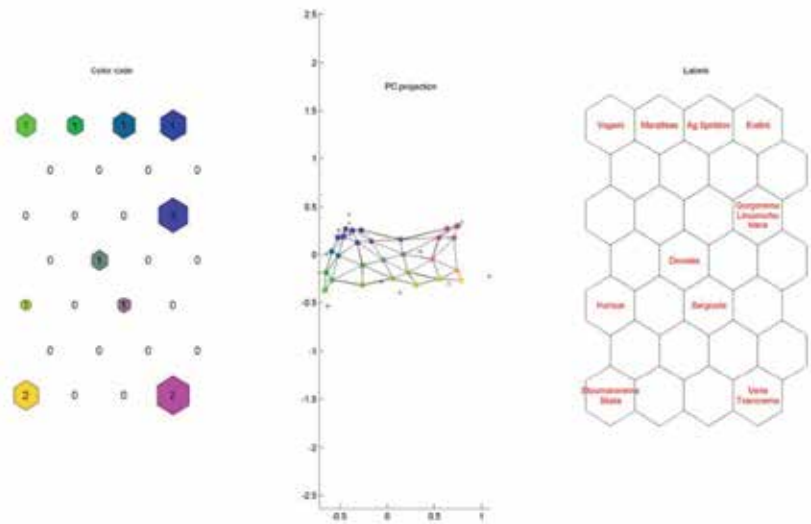


Fig. 5. Different visualizations of the clusters obtained from the classification of the morphological variables through SOM. (a) Colour code using k-means; (b) Principal component projection; (c) Label map with the names of the alluvial fans, using k-means. . The four clusters are indicated through the coloured circles

In Table 5 the rules governing each class are described.

Explanation	Symbol	Group1	Group2	Group3	Group4
Cluster Index Value	CIV	5	2	5	2
		fluvial dominated Varia, Skala, Sergoula, Stournarorema, Tranorema	debris flow Marathias, Vogeni	fluvial dominated with debris flow influences Ag. Spyridon, Mara, Gorgorema, Linovrocho, Eratini	fluvial dominated Dounias, Hurous
Drainage basin area	(A_b)	> 15.8 High	< 15.8 Low	< 15.8 Low	Medium
Basin crest	(C_b)	> 1160 High	< 1160 Low	< 1160 Low	> 1160 High
Perimeter of the drainage basin	(P_b)	>15.4 High	< 15.4 Low	< 15.4 Low	< 15.4 Low
Total length of the channels within the drainage basin	(L_c)	> 48.6 High	< 48.6 Low	< 48.6 Low	< 48.6 Low
Total length of 20 m contour lines within the drainage basin	(ΣL_c)	> 421 High	< 421 Low	< 421 Low	< 421 Low
Basin relief	(R_b)	< 437 Low	> 437 High	> 437 High	< 437 Low
Melton' s ruggedness number	(M)	< 0.4 Low	> 0.4 High	> 0.4 High	Medium
Drainage basin slope	(S_b)	Medium to high	<0.08 Low	<0.08 Low	>0.08 High
Drainage basin circularity	(Cirb)	<0.60 Low	<0.60 Low	Medium	>0.60 High
Drainage basin density	(D_b)	> 3.05 High	< 3.05 Low	Medium	> 3.05 High
Fan area	(A_f)	>1.97 High	<1.97 Low	<1.97 Low	Medium
Fan length	(L_f)	<1.93 High	>1.93 Low	>1.93 Low	>1.93 Medium
Fan apex	(Ap_f)	not clear	> 100 High	< 100 Low	> 100 High
Fan slope	(S_f)	< 0.03 Low	> 0.03 High	Medium	< 0.03 Low
Fan concavity	(C_f)	not clear	>1.28 High	Medium	>1.28 High

Well developped channels	R	Yes	No	Yes	Yes
Prevailing geological formation in basin area	Geo	Limestone	Flysch	Flysch	Limestone

Table 5. Clusters originating from SOM classification

5.2 Mesoscale landform characterization using unsupervised SOM

The systematic classification of landforms, their components, and associations, as well as their regional structure is one prerequisite for understanding geomorphic systems on different spatial and temporal scales (Dikau & Schmidt, 1999). The aim is to locate any correlation schemes between first and second derivatives describing the basin areas and alluvial fan regions, and examine clustering tendency of the data to certain line or surface morphometric features, (i.e. channels, ridges, planar surfaces). The data set comprised 3222 records, from a 250m spacing DEM, covering the whole study area (i.e. fourteen drainage basins and corresponding alluvial fans).

In order to assess the optimum SOM, 11 SOMs were developed. Learning of SOM was performed with random initial weighs of the map units. The initial radius was set to 3 and the final radius to 1. The initial learning rate was set to 0.5 and the final to 0.05. Experimenting towards SOM optimization the size of the map progressively augmented from 70 to 300, with a decreasing (QE) from 0.37 to 0.25. The optimum architecture was built through trial and error procedure. The SOM which gave the best map had QE 0.111 after 1000 epochs (Fig. 6). The optimum architecture was used in 10 more trials with random initial weights, so as to test the influence, on (QE). According to the findings of this study, there was no influence, which is probably attributed to the long time of training. That is, initial random weight values are being trained and Euclidian distances between input data vectors and best matching units decrease and reach the minimum value and become stable.

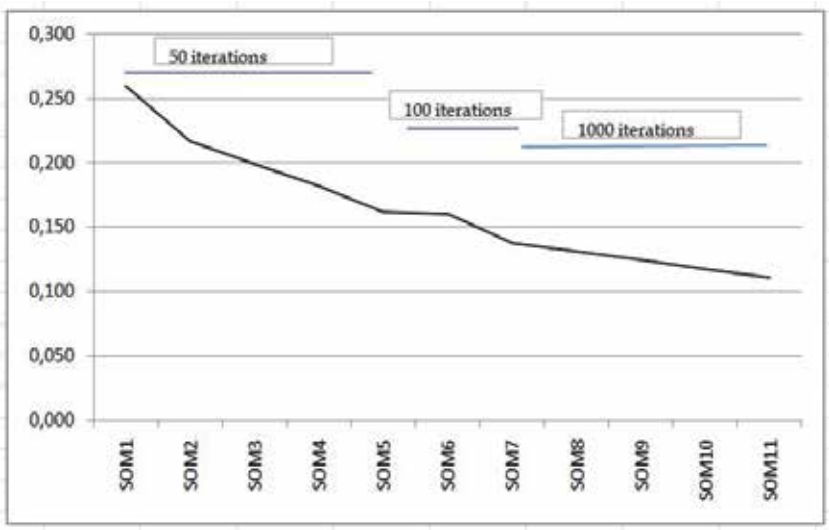
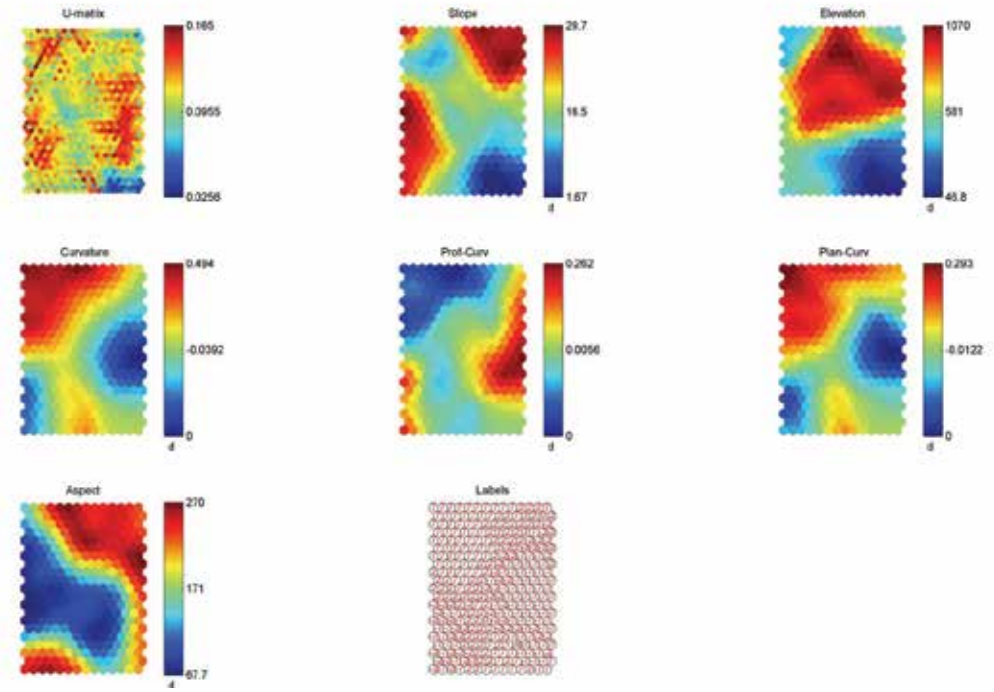
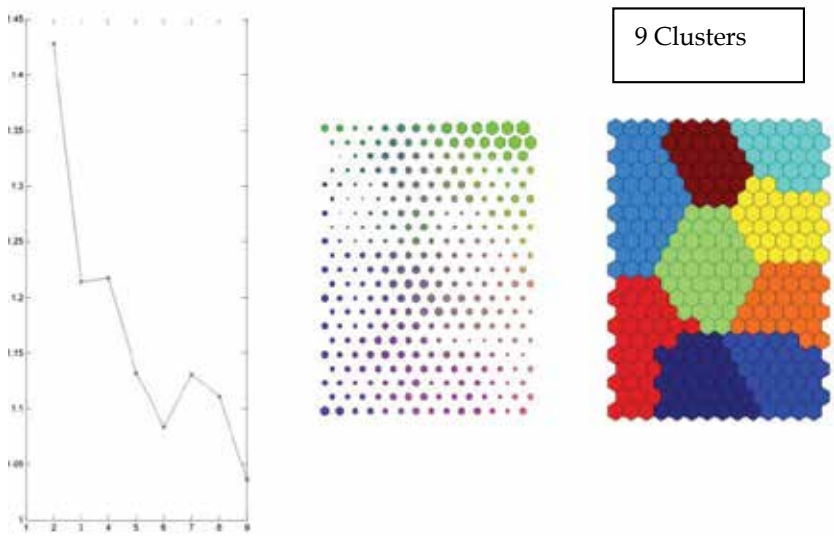


Fig. 6. Effect of number of epochs on average quantization error



a



b

Fig. 7. (a) SOM visualization through U-matrix (top left), and 6 component planes, one for each parameter examined (b) from left to right, through, Davis - Bouldin validity index versus cluster number, colour coding, and clustering using k-means (upper left (1) counting clockwise, (9) in the centre

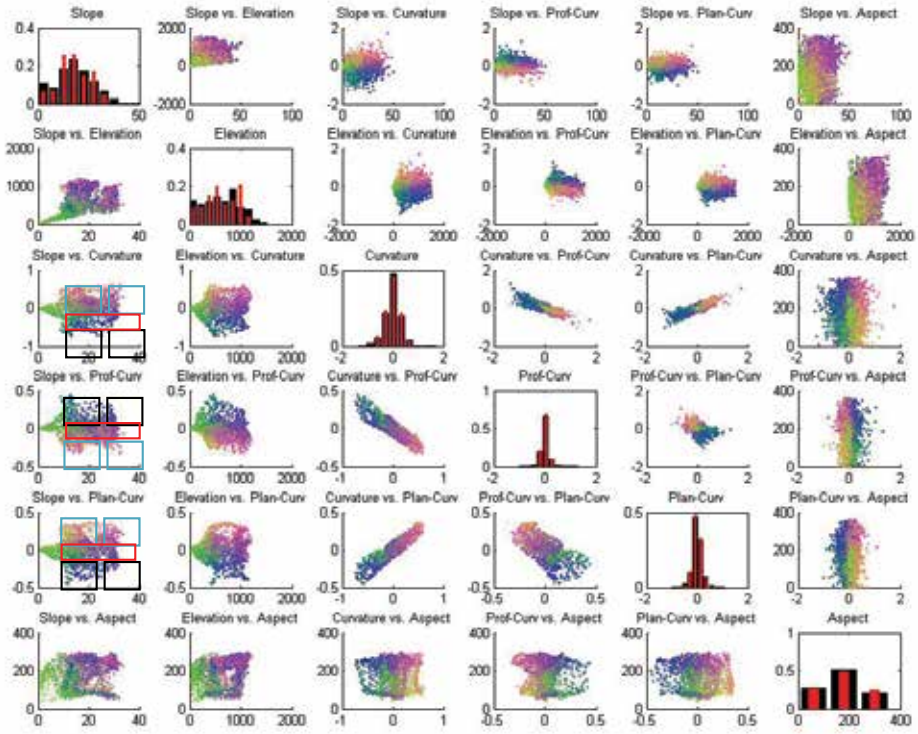


Fig. 8. SOM visualization through scatter diagrams of studied morphometric parameters

The next step is the analysis, interpretation and labeling of the map units as morphometric features. Correlation between slope and elevation, curvature and plan curvature is displayed through U-matrix (Fig. 7) and scatter diagrams (Fig. 8). Profile curvature is inversely correlated to plan curvature. No clear correlation on aspect and the other derivatives is portrayed.

U-matrix shows no clear separation between clusters, but using k-means algorithm and Davis – Bouldin (1979) index (Fig. 7b), it seems that 9 existing clusters correspond to different terrain forms. From the component planes, it can be seen that the features differentiating the clusters are the following presented in Table 7. In this table, the categorized map units and the corresponding morphometric features are summarized. For example ridges in the study area are represented with clusters 1,2,7 but with different slope and elevation conditions. This feature corresponds to both steeper and slopes representing an approximately flat area. Cluster 9 corresponds to flat area, possibly planation areas, in higher elevation almost flat terrain. Cluster 3 and 8 correspond to channels, with different slope conditions.

The black boxes plotted in Fig.8 refer to convex ridges, and the cyan boxes to concave channels. In order to hunt correlations between parameters, one should scan through the scatter diagrams in the lower triangle (resulting after training) where both data and map units are plot. According to SOM training, channels (negative concavities) are recognized and constitute two subgroups from low to steep slopes. Convex ridges are also recognized separated in classes from moderate to steep slopes. Planar surfaces are also recognized and differentiated according to slope angle. It is evident in Fig.8, that planar surfaces of gentle to steep slopes exist, in the study area.

Class	Morphometric element	Slope (^{0})	Elevation (m)	Curvature	Profile curvature	Plan curvature	Aspect
Cluster 1	Ridge	Medium (16)	Medium to High 580 to 1070	+	0	+	E to SE
Cluster 2	Ridge	Medium to High > (16)	High > 750	+	0	+	W
Cluster 3	Channel	High > (23)	Medium - High > 560	-	+	-	W
Cluster 4	Planar	Medium to high	High > 750	0	+	0	E to N
Cluster 5	Planar	Low to Medium	Low < 560	0	+	0	E to NE
Cluster 6	Chanel	Very Low	Low	-	+	-	E
Cluster 7	Ridge	Low	Low	+	-	+	S to SW
Cluster 8	Chanel	High	Low	-	+	-	W
Cluster 9	Planar	Low	High	0	0	0	NE to E

Table 7. Clusters originating from SOM

5.3 Mesoscale landform characterization using supervised SOM

SOM algorithm was proposed, as an alternative procedure for terrain analysis to Peuker and Douglass method. SOM training was performed with a subset of the DEM, referring to Varia and Scala drainage basins (see Fig.3). Six morphometric parameters were used, as input and a two-dimensional output of 3,000 neurons. Sampling procedure for the data set describing the drainage basins was performed. A sampling function was applied to the derivatives grids in order to prepare a matrix of sample vectors. The sampling was performed to the DEM and DEM derivatives, at 90m spacing. Problems handling memory had to be faced, this is why a small subset of the training DEM was used. The produced ASCII file was exported to MATLAB in order to use SOM unsupervised neural networks. The data set is presented in Table 4. The data dimensions was 25,024 x 6.

At the beginning of the learning procedure, neurons in the SOM were distributed randomly. The BMUs (final classes) with minimum average (QE) 0.135 were extracted. The number of map units was finally set to 3,000. Turning the SOM into a supervised classifier the final error was 30%. In table 8 the results of the applied normalizations are displayed. The error of supervised clustering is also presented. "HistD" normalization gave the best results, after 1,000 iterations.

Normalization method	QE	TE	error
histC	0.182	0.040	36.8
Var	0.40	0.045	35.4
Log	0.198	0.036	28.4
Logistic	0.34	0.054	33.61
Range	0.180	0.050	41.52
histD	0.210	0.033	27.92

Table 8. Normalization methods, and calculated QE and TE, for supervised clustering

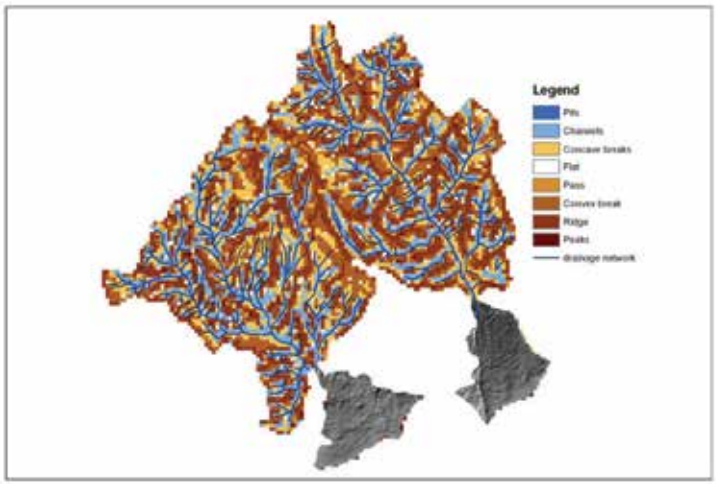


Fig. 9. Outcome of Peuker and Douglas classification

The results of the supervised clustering are presented in Fig. 10. The results illustrate a very clear distinction between the disparate morphometric features. Line-based and planar features were mainly recognized. A rather good network of ridges and channels with different slope classes is revealed. Compared to the outcome of classic morphometric analysis in Fig. 9 the outcome we get through SOM seems more compact, with a very good representation of crest lines. According to Peuker and Douglas method about 41% of the area are concave and convex breaks, 27 % are channels and 28% are ridges. As expected point- based features such as peaks, passes and pits cover only 4% of the study area. This is probably attributed to the fact that point based features are comparatively rare.

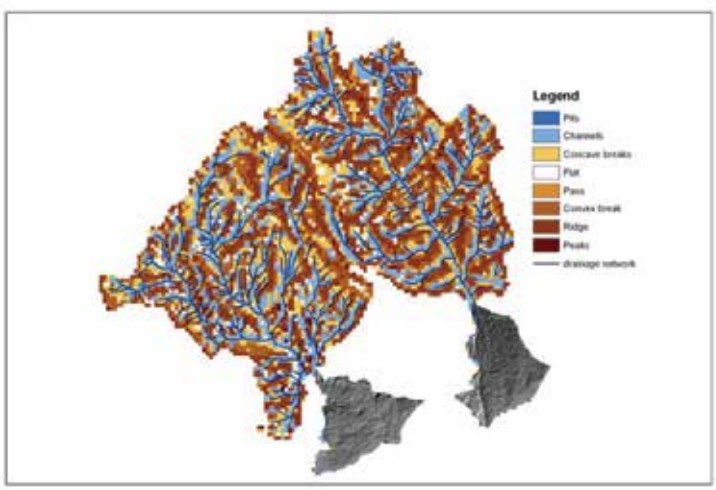


Fig. 10. Outcome of SOM clustering

6. Discussion

Given that geomorphological mapping is the basis for terrain assessment, a geomorphological map was constructed, to validate the results of the SOM drainage basin landscape mesoscale classification, for the catchments of Varia and Skala streams which are the two westernmost among the studied basins (Fig.11). Geomorphological mapping was performed using a 1:50,000 base topographic map through fieldwork, and aerial photo interpretation taking also into account previous geological maps.

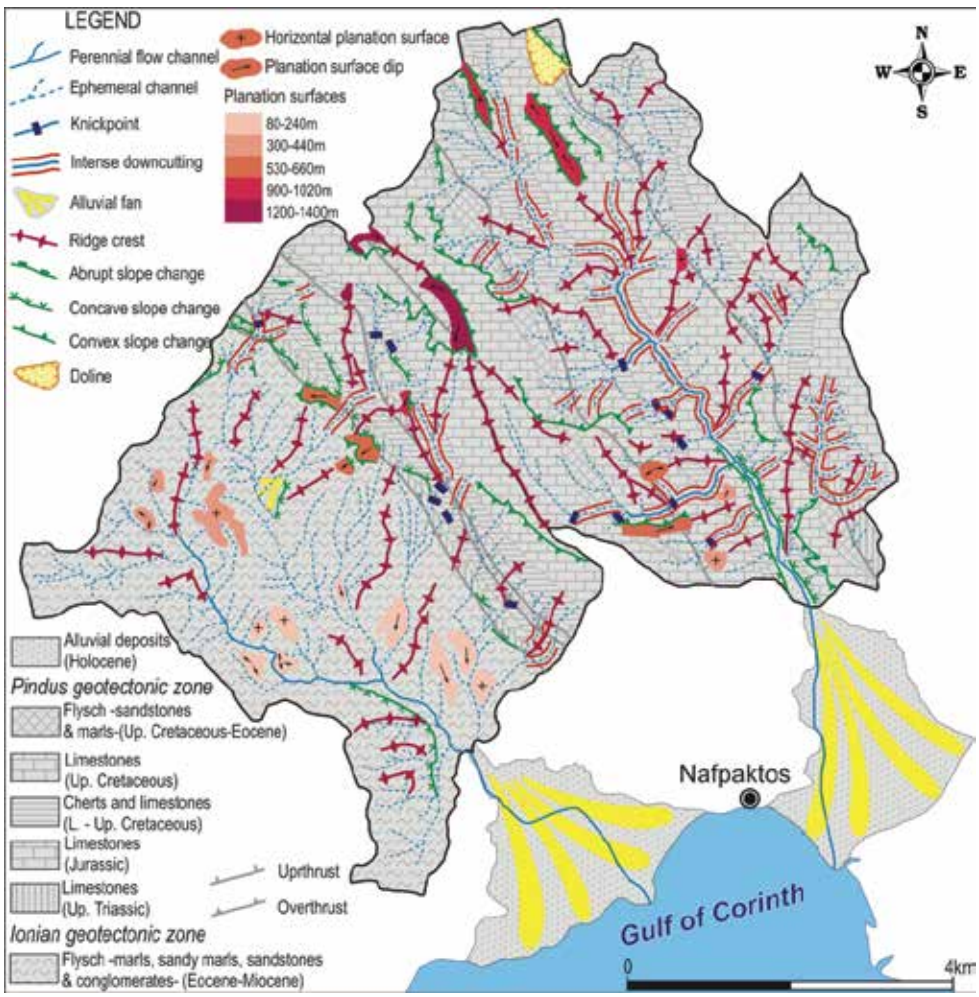


Fig. 11. Geomorphological map of the Varia and Skala streams and drainage basin areas, bedrock lithology is derived by the geological maps of (IGME) and field observations

The purpose of the mapping, which was its comparison with the SOM derived classification map, was the main criterion for the selection of the scale of the map. The scale is critical for effective information delivery. The final map provides information on the distribution of geological formations while landforms identifying landscape features created by surface processes were recorded combining field inspection with maps and aerial photo

interpretation. These landforms which include erosional planation surfaces ranging in elevation from 500 m to 1,000m, stream channels and valleys of various shape, knickpoints, abrupt slope breaks, gentler slope changes, ridges and crests, alluvial fans and cones, intense channel downcutting, provide information on earth surface form processes.

Comparative observation of the geomorphological map, Peuker and Douglas classification, and the SOM clustering reveals information on the accuracy of the landscape characterization approach through SOM. Both methods identified stream channels of the drainage networks with very accurately. The more well developed high order channels like those of the main streams of the networks were better detected and recognized using SOM. Additionally, SOM identified correctly ridges and drainage divides providing an ideal method for drawing drainage basins borders. On the other hand landforms like erosional planation surfaces or knickpoints (discrete negative steps in the longitudinal profile of a river), are not identifiable on the SOM clustering.

In terms of evaluation results, Peuker and Douglas method and SOM, were compared, with an oblique view, overlaying contour lines (Figure 12). SOM is much closer to the geomorphological mapping, approach, and has much more potential for identification of non-point morphometric features than Peuker and Douglas method. The overall pattern of channels, ridges and planes is similar in both methods, but the SOM results are more concrete and seem to resemble to the classification of the geomorphological mapping, which recognizes unique landforms. Furthermore, the SOM capability of identifying crest lines on mountain ranges is also important. Last, the SOM method does not rely on curvature and slope tolerance values. In this method, the slope parameter, elevation and aspect, are important in characterizing classes, rather than just being a threshold to separate horizontal surfaces from sloping surfaces. Using the whole potential of the slope parameter in extracting features that are more informative is one of the advantages of the SOM.

Concerning the accuracy of the alluvial fan classification utilizing SOM it is obvious that this approach provides one of the best methods to characterize alluvial fans considering the correlation between alluvial fans and geomorphometric characteristics and quantitative morphometric indices of their corresponding drainage basins.

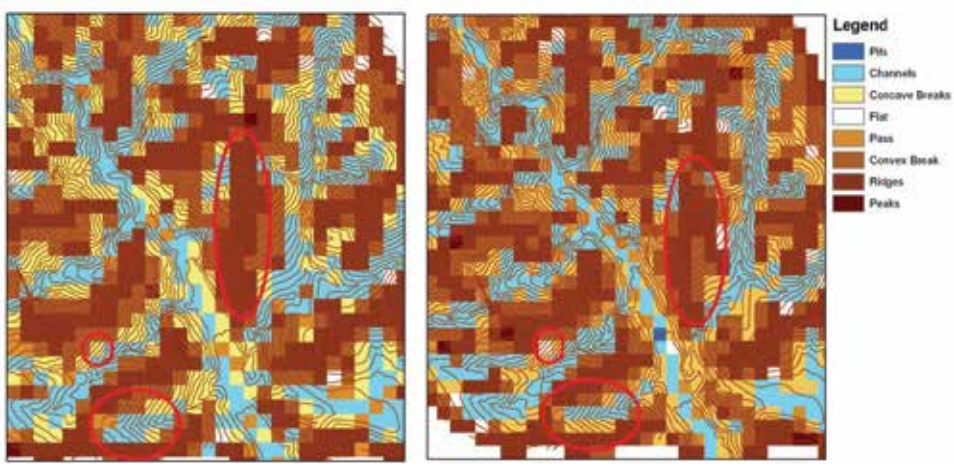


Fig. 12. Classification results (a) Terrain analysis according to SOM clustering, (b) Terrain analysis according to Peuker and Douglas

The results of the SOM characterization of the studied coastal alluvial fans according to the primary processes which are responsible for their formation and evolution, in four classes are in good agreement with a classification of the same fans performed in previous study based on entirely qualitative geomorphological observations and investigation of the relationship between pairs of selected fan – basin morphometric variables (Karymbalis, 2007). Especially the grouping of Marathias and Vogeri fans (cluster 2) and their characterization as debris flow dominated is validated by the existence of two landslides triggered by earthquakes activity along the Marathias normal fault scarp (Gallousi & Koukouvelas, 2007). Field observation showed that these two fans have formed by coarse grained material supplied by landslides. One of the advantages of the method is that provides the opportunity to correlate quantitative variables with qualitative information in order to achieve better results in alluvial fan classification.

7. Conclusion

The purpose of this study is to investigate the effectiveness of Self Organising Map (SOM), as a clustering tool in the field of applied geomorphology for mapping meso and microrelief scale morphometric elements. Unsupervised artificial neural networks, which use knowledge discovery methods, were developed in order to detect the trend of the data to clustering in microrelief scale according to alluvial fan formation and evolution process and in mesorelief scale according to linebased and planar morphometric features. SOM was also used as a semiautomatic tool in terrain analysis, with an accuracy result of 70%. Comparison of the geomorphological map and the SOM mesoscale landform classification results for two drainage basins in central Greece showed that the applied methodology is a promissing method for the mapping of drainage network channels. Furthermore, this approach resulted in successful identification of ridges and crests providing a good way to draw drainage divides and border hydrological basins. Compared to classic terrain analysis method, SOM presented a more concrete and accurate result in line base and planar elements.

The systematic and objective method which was applied in the field of alluvial fan classification, compared to statistical methods (Karymbalis et al., 2010) and geomorphological observations, was reasonable and accurate. This method could be applied as a generic tool of alluvial fan classifier to larger data sets, in order to assess and interpret dominant formation processes, through the study of many morphometric features describing alluvial fans and corresponding drainage basins.

In both mesoscale and microscale the SOM proved an efficient scalable tool for the analysis of geomorphometric features as meaningful landform elements, and uses the full potential of morphometric characteristics, leading to better understanding complex geomorphological systems.

8. References

Achurra, L.E., Lacassie, J.P., Le Roux, J.P., Marquardt, C., Belmar, M., Ruiz-del-Solar, J. & Ishman, S.E. (2009). Manganese nodules in the Miocene Bahia Inglesa Formation, north-central Chile: Petrography, geochemistry, genesis and palaeoceanographic significance. *Sedimentary Geology*, 217, 128-139.

Adediran, A.O., Parcharidis, I., Poscolieri, M., & Pavlopoulos, K. (2004). Computer-assisted discrimination of morphological units on north-central Crete (Greece) by applying multivariate statistics to local relief gradients. *Geomorphology*, 58, 357–370

Alvarez-Guerra, M., González-Piñuela, C., Andrés, A., Galán, B. & Viguri, J.R. (2008) Assessment of Self-Organizing Map artificial neural networks for the classification of sediment quality. *Environment International*, 34, 782-790.

Armijo, R., Meyer, B., King, G., Rico, A. & D. Papanastasiou (1996). Quaternary evolution of the Corinth Rift and its implications for the Late Cenozoic evolution of the Aegean. *Geophys. J. Int.*, 126, 11-53.

Barthkowiak, A. & Evelpidou, N. (2006) Visualizing some multi-class erosion data using kernel methods. In: *Proceedings in Computational Statistics* , A. Rizzi, M. Vichi, (Ed.), 805–812, Physica Verlag, Springer Company.

Bishop, M.P. & Shroder Jr, J.F. (2004). *Geographic Information Science and Mountain Geomorphology*, Springer – Praxis, 3-540-42640-X, Berlin, Heidelberg, New York.

Brabyn, L.K. (1997). Classification of macro landforms using GIS. *ITC J*,1, 26–40.

Briole, P., Rico, A., Lyon-Caen, J., Ruegg, J., Papazissi, K., Mitsakaki, A., Balodimou, G., VeisS, G., Hatzfeld, D. & A. Deschamps (2000). Active deformation of the Corinth Rift, Greece: Results from repeated GPS surveys between 1990 and 1995. *J. Geophys. Res.*, 105(B11), 25606-25626.

Brooks, M. & Ferentinos G. (1984). Tectonics and sedimentation in the Gulf of Corinth and the Zakynthos and Kefallinia Channels, western Greece. *Tectonophysics*, 101, 25-54.

Brown, D.G., Lusch, D.P. & Duda, K.A. (1998). Supervised classification of types of glaciated landscapes using digital elevation data. *Geomorphology*, 21, 233-250.

Bue, B.D. & Stepinski, T.F. (2006). Automated classification of landforms on Mars. *Computers & Geosciences* 32, 604–614.

Burrough, P.A., Van Gaans, P.F.M. & MacMillan, R.A. (2000). High-resolution landform classification using fuzzy k-means. *Fuzzy Sets and Systems*, 113, 37–52.

Buza, A., McDonald, J.A., Michalak, J. & Stuetzle, W. (1991) Interactive data visualization using focusing and linking. In: *Proceedings of IEEE conference of visualization*, 156–63

Carniel, R., Barbui, L.& Malisan, P. (2009). Improvement of HVSR technique by self-organizing map (SOM) analysis. *Soil Dynamics and Earthquake Engineering*, 29, 1097-1101.

Chang T.C. & Chao R. J.(2006). Application of back – propagation networks in debris flow prediction. *Engineering Geology*, 85,270-280.

Chang, H.-C., Chen, H.-C. & Kopaska-Merkel, D.C. (1998) Identification of lithofacies using ART neural networks and group decision making. In: *Proceedings of Artificial Neural Networks in Engineering Conference*, (Eds), St. Louis, Missouri, USA, 855–860.

Chang, H.-C., Kopaska-Merkel, D.C., Chen, H-C., Durrans & Rocky, S. (2000) Lithofacies identification using multiple adaptive resonance theory neural networks and group decision expert system. *Computers & Geosciences*, 26, 591–601.

Chang, H-C., Kopaska-Merkel, D.C. & Chen H-C. (2002) Identification of lithofacies using Kohonen self-organising maps. *Computers and Geosciences*, 28, 223-229.

Crosta, G. & P. Frattini (2004). Controls on modern alluvial fan processes in the central Alps, Northern Italy. *Earth Surface Processes and Landforms*, 29, 267-293.

Davies, D. L. & Bouldin, D. W. (1979) A Cluster Separation Measure. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. PAMI-1 (2): 224-227.

Dehn, M., Gärtner, H. & Dikau, R. (2001). Principles of semantic modeling of landform structures. *Computers & Geosciences*, 27, 1005-1010

Dikau, R. (1989). The application of a digital relief model to landform analysis in geomorphology. In: *Three Dimensional Applications in Geographical Information Systems,* Raper, J. (Ed.), 51–77, Taylor & Francis, London.

Dikau, R., Brabb, E.E. & Mark, R.K. (1991). Landform Classification of New Mexico by Computer. *Open-File Rep. (U. S. Geol. Surv.)* 91–634, 15 pp.

Dikau,. R& Schmidt,. J (1999). Georeliefklassifikation. In. Schneider-Sliwa,R. Schaub D, &. Gerold D (Eds.), *Angewandte Landschaftsökologie*. Grundlagen und Methoden Springer. Berlin: 217–244.

Ehsani, A.H., & Quiel, F. (2008). Geomorphometric feature analysis using morphometric parameterization and artificial neural networks. *Geomorphology*, 99(1-4):1-12.

Ersoya, O., Aydara, E., Gourgaudb, A., Artunerc, H. & Bayhan, H. (2007) Clustering of Volcanic Ash Arising from Different Fragmentation Mechanisms Using Kohonen Self-Organizing Maps. *Computers and Geosciences*, 33, 821-828.

Esposito, A.M., Giudicepietro, F., D'Auria, L., Scarpetta, S., Martini, M.G., Coltelli, M. & Marinaro M. (2008) Unsupervised Neural Analysis of Very-Long-Period Events at Stromboli Volcano Using the Self-Organizing Maps. *Bull. of the Seismological Society of America*, 98, 2449–2459

Evans, I.S. (1972). General geomorphology, derivatives of altitude and descriptive statistics. In: *Spatial Analysis in Geomorphology*, Chorley, R.J. (Ed.), 17–90, Methuen & Co. Ltd, London

Ferentinou, M. & Sakellariou, M. (2005) Assessing landslide hazard on medium and large scales, using SOM. In: *Landslide Risk Management*, Hungr O., Fell R., Couture R. & Eberhardt E., (Ed.), 639-648, Taylor & Francis.

Ferentinou, M. & Sakellariou, M. (2007) Computational intelligence tools for the prediction of slope performance. *Computers and Geotechnics*, 34, 362-384.

Ferentinou, M., Hasiotis, T. & Sakellariou, M. (2010) Clustering of geotechnical properties of marine sediments through self – organizing maps: An example from the Zakynthos Canyon–Valley system, Greece. In: *Submarine Mass Movements and their consequences IV*, Mosher, D., Shipp, C., Moscardelli, L., Chaytor, J., Baxter C., Lee, H. & Urgeles, R., (Ed.), Advances in Natural and Technological Hazards Research, v. 28, 43-54, Springer, The Netherlands.

Fisher, P., Wood, J. & Cheng, T. (2004). Where is Helvellyn? Fuzziness of multiscale landscape morphometry. *Transactions of the Institute of British Geographers*, 29, 106–128.

Gallousi, C. & Koukouvellas I., (2007). Quantifying geomorphic evolution of earthquake-triggered landslides and their relation to active normal faults. An example from the Gulf of Corinth, Greece. *Tectonophysics*, 440, 85-104.

Giles, P.T. (1998). Geomorphological signatures: classification of aggregated slope unit objects from digital elevation and remote sensing data. *Earth Surface Processes and Landforms*, 23, 581-594

Giles, P.T. & Franklin, S.E. (1998). An automated approach to the classification of the slope units using digital data. *Geomorphology*, 21, 251–264.

Giles, P.T., Investigating the use of alluvial fan volume to represent fan size in morphometric studies, *Geomorphology* (2010), 121, 317-328.

Hammond, E.H. (1964). Analysis of properties in land form geography:an application to broad-scale landform mapping. *Ann. Assoc. Am. Geogr*, 54, 11–19.

Harvey, A.M. (1997). The role of alluvial fans in arid zone fluvial systems. *In: Arid Zone Geomorphology: Process, Form and Change in Drylands,* D.S.G. Thomas (Ed.), John Wiley and Sons Limited, Chichester, 231-259.

Irvin, B.J., Ventura, S.J. & Slater, B.K. (1997). Fuzzy and isodata classification of landform elements from digital terrain data in Pleasant Valley, *Wisconsin Geoderma*, 77, 137-154.

Iwahashi, J. & Pike, R.J. (2007). Automated classifications of topography from DEMs by an unsupervised nested-means algorithm and a three-part geometric signature. *Geomorphology*, 86, 409-440

Loftus, D.L. & P. Tsoflias (1971). Geological map of Greece, *Nafpaktos sheet*, scale 1:50000. Greek Institute of Geological and Mining Reseach.

Karymbalis, E. (2007). Fan deltas geomorphology in the northern coast of Gulf of Corinth, Greece: *Proceedings of the Eighth International Conference on the Mediterranean Coastal Environment MEDCOAST 2007* , E. Ozhan (ed.), Alexandria, Egypt, 1321-1332.

Karymbalis, E., Gaki- Papanastasiou, K., Ferentinou, M., (2010) Coastal Fan Deltas Classification Along the NW Coast of Gulf Of Corinth, Greece Coupling Morphometric Analysis and Artificial Neural Networks, *Hellenic Journal of Geosciences*, vol. 45, (in press).

Kostaschuck, R.A., MacDonald G.M. & Putnam P.E. (1986) Depositional processes and alluvial fan-drainage basin morphometric relationships near Banff, Alberta, Canada. In: *Earth Surf. Proc. 11*, pp. 471–484.

Kohonen, T. (1995) *Self - Organising Maps*. Springer, Berlin.

Lambeck, K., 1996. Sea-level change and shore-line evolution in Aegean Greece since Upper Palaeolithic Time. *Antiquity*, 70, 588-611.

Lee, B.-H. & Scholz, M. (2006) Application of the self-organizing map (SOM) to assess the heavy metal removal performance in experimental constructed wetlands. *Water Research*, 40, 3367-3374.

Melton, M.A. (1965). The geomorphic and palaeoclimatic significance of alluvial deposits in Southern Arizona. *Journal of Geology*, 73, 1-38.

Miliaresis, G.C. (2001). Geomorphometric mapping of Zagros Ranges at regional scale. *Computers & Geosciences*, 27, 775–786.

Morgan, J.M. & Lesh, A.M. (2005). Developing landform maps using ESRI's ModelBuilder. *ESRI User Conference Proceedings*, Redlands, CA.

Murray, A.B., Lazarus, E., Ashton, A., Baas, A., Coco, G., Coulthard, T., Fonstad, M., Haff, P., McNamara, D., Paola, C., Pelletier, J. & Reinhardt, L. (2009). Geomorphology, complexity and the emerging science of the earth's surface. *Geomorphology*, 103, 496-505.

Paraschoudis, B. (1977). Geological map of Greece, Amygdalea sheet, scale 1:50000. Greek Institute of Geological and Mining Reseach.

Peuker T. & Douglas D. (1975). Detection of surface – specific points by local parallel processing of discrete terrain. Computer *Graphics and Image Processing*, 4, 375-387.

Pike, R.J. (2000). Geomorphology: diversity in quantitative surface analysis. *Progress in Physical Geography*, 24, 1–20.

Prima, O.D.A., Echigo, A., Yokoyama, R., & Yoshida, T. (2006). Supervised landform classification of Northeast Honshu from DEM-derived thematic maps. Geomorphology, 78, 373-386.

Sorriso-Valvo, M. & Sylvester, A.G., (1993). The relationship between geology and landforms along a coastal mountain front, northern Calabria, Italy. *Earth Surface processes and Landforms*, 18, 257-273.

Sorriso-Valvo, M., Antronico L., Le Pera E.. (1998). Controls on modern fan morphology in Calabria, Southern Italy. *Geomorphology*, 24, 169-187.

Stepinski, T.F., & Collier, M.L. (2004). Extraction of Martian valley networks from digital topography. *Journal of Geophysical Research*, 109

Tsakovski, S., Kudlakb, B., Simeonova, V., Wolskab, L. & Namiesnikb, J. (2009). Ecotoxicity and chemical sediment data classification by the use of self-organising maps. *Analytica Chimica Acta*, 631, 142–152.

Tselentis, G-A., Serpetsidaki, A., Martakis, N., Sokos, E., Paraskevopoulos, P. & Kapotas, S. (2007) Local high-resolution passive seismic tomography and Kohonen neural networks - Application at the Rio-Antirio Strait, central Greece. *Geophysics*, 72, B93-B106.

Vesanto, J. & Alboniemi, E. (2000) Clustering of the Self – Organising Maps. *IEEE Transactions on Neural Networks*, 11, 586-600.

Vesanto, J. (2000) Using SOM in Data Mining, *Licentiate's thesis*. Helsinki University of Technology.

Wood, J. (1996a). The Geomorphological Characterization of Digital Elevation Models. *Ph.D. Thesis*, Department of Geography, University of Leicester, UK.

Wood, J. (1996b). Scale-based characterisation of digital elevation models. In: *Innovations in GIS 3: Selected Papers from the Third National Conference on GIS Research UK (GISUK)*.Parker, D. (Ed.), 163–175, Taylor & Francis, London.

Zevenbergen, L.W., & Thorne, C.R. (1987). Quantitative analysis of land surface topography, *Earth Surface Processes and Landforms* 12 , 47–56.

16

Land-Cover Classification Using Self-Organizing Maps Clustered with Spectral and Spatial Information

M. L. Gonçalves[1], J. A. F. Costa[2] and M. L. A. Netto[3]
[1]*Department of Computer Science,*
Pontifical Catholic University of Minas Gerais, Poços de Caldas, MG
[2]*Department of Electrical Engineering*
Federal University of Rio Grande do Norte, Natal, RN
[3]*School of Electrical and Computer Engineering*
State University of Campinas, Campinas, SP
Brazil

1. Introduction

Digital classification methods of remotely sensed images have acquired a growing importance in the automatic recognition of the land cover patterns. The enormous quantity of images that are being generated from an increasing number of highly sophisticated sensor systems require the development of innovative classification methodologies, which allow an automatic and efficient detection of the great volume of data available in the images and at the same time makes the mapping process of terrestrial surfaces less subjective and with greater potential for reuse in subsequent situations.

Particularly, unsupervised classification methods have traditionally been considered as an important approach for the interpretation of remotely sensed images. This approach of classification plays an especially significant role when very little a priori information about image data is available, and for that reason continues to be a popular choice for analysts without ample field knowledge or for those wanting to avoid introduced bias in classification analysis (Duda & Canty, 2002; Kelly et al., 2004).

Unsupervised classification is frequently performed through clustering methods. These methods examine the unknown pixels in an image and incorporate them into a set of classes defined through the natural clusters of the gray levels of the pixels. Cluster analysis provides a practical method for organizing a large set of data so that the retrieval of information may be made more efficiently. However, although there is a large quantity of different clustering methods in the pattern recognition area (Xu & Wunsch II, 2005), only a limited quantity of them can be used in remote sensing applications. As pointed out in Tran et. al. (2005), several problems are encountered when clustering remotely sensed images, but above all the image size and the feature dimension problems are those that often make a method inappropriate due to computation time and computer memory.

The most common clustering method applied to remotely sensed data is partitional. The majority of software or computational systems meant for the digital processing of remotely

sensed images specifically contain two partitional clustering algorithms to perform unsupervised classification: the K-means algorithm, and a variation of it, the ISODATA (Ball & Hall, 1967). Despite being widely used, these partitional clustering methods have various limitations. The objective functions that they use begin with the assumption that the number of classes, K, is known a priori. In the hypothesis that an inadequate K' value has been chosen, the method will impose, through the use of optimization techniques, K' clusters to the data. The user must also manually specify various parameters in order to control the clustering process, among them: the initial centroids of each cluster, the maximum number of iterations, thresholds to perform the division, fusion, or exclusion of clusters. K-means and ISODATA are very sensitive to these parameters, which can generate different partitions when various simulations are done for the same data set. Facing this, the optimal value of these parameters is frequently encountered through trial and error. These needs certainly increase the level of interaction between the user and the computational algorithm, consequently increasing the degree of subjectivity of the categorization process of the image. Other no less important limitations of partitional algorithms, such as K-means and ISODATA, are: the high computational cost when the data to be analyzed is very large (at each iteration, all of the pixels in the image are compared with all of the clusters centroids) and the existence of suppositions about the cluster forms. Generally only one prototype (centroid) is used to represent a cluster, thus these methods become adequate only for the analysis of clusters that have hyper-spherical formats (Shah et al., 2004).

Another possible, though uncommon, way of performing unsupervised classification of remotely sensed images is through hierarchical clustering methods. Unlike partitional methods, hierarchical methods do not require the user to specify the number of clusters and other additional parameters beforehand, and therefore are not affected by initialization and local minima. Another significant advantage of these methods is that they make it possible to visualize the result of the classification by means of a dendrogram which illustrates in a hierarchical form the degree of similarity between the clusters that are formed by fusions (or divisions) at each successive stage of the analysis. However, hierarchical methods have some characteristics that prevent their application in the classification of remotely sensed images: (a) in general they require memory space in the order of $O(N^2)$, in which N is the number of records in the data set; (b) dendrograms are impractical for large data sets, and therefore the results can be difficult to interpret; (c) in order to determine the cut-off of the dendrogram (ideal number of clusters) some decision criteria must be applied (Duda & Canty, 2002; Kelly et al., 2004).

Consequently, hierarchical clustering modules are rare in remote sensing software packages and for this reason are normally utilized in an uncoupled way. In the literature, there is not much research that makes use of hierarchic methods in unsupervised classification of remotely sensed images, and in general common versions of these methods are applied in practice when the main objective is basically to compare performance with other unsupervised methods (Duda & Canty, 2002; Wilson, et al., 2002). However, a relatively recent progression of approaches attempting to make the use of hierarchical clustering methods viable in remote sensing applications can be seen. For instance, instead of applying agglomerative hierarchical methods directly on a pixel by pixel basis, some works have employed these methods only after the application of some other clustering method on the original image data. In Tran et al. (2003) the K-means algorithm was used to classify the image data before the application of a hierarchical method, and the ISODATA was used for the same reason in Marçal & Castro (2005) and in Marçal & Borges (2005). Although this

strategy is effective at preventing the (expensive) application of hierarchic methods to large data sets, a possible disadvantage of it is that the results of the classification methods presented in these works can be influenced by the limitations of the K-means and ISODATA algorithms, which were mentioned earlier here.

Evidently, the literature contains other proposals for unsupervised methods to classify remote sensing data that do not make use only of conventional clustering approaches (Huang, 2002; Shah et al., 2004; Tso & Olsen, 2005). In a special way, a number of researchers have proposed several sophisticated approaches to classify remote sensing images based on computational intelligence methods like neural networks, fuzzy systems, and evolutionary algorithms (Bandyopadhyay, 2005; Ji, 2003; Liu et al., 2005).

Particularly, some works has investigated the use of the Kohonen neural network (more commonly known as SOM - Self-Organizing Map) for the task of clustering and classification of remotely sensed images. Ji (2000) showed that the SOM algorithm is capable of achieving higher classification accuracy as compared to that of the maximum-likelihood method and concluded that the neural network is a useful viable alternative for the classification of remotely sensed images. Vilmann et al. (2003) studied the effect of the application of several extensions of the SOM in the cluster analysis of multispectral and hyperspectral images. The authors showed excellent data clustering and classification, and concluded that some considerations about the consequence of the dimensionality reduction produced by SOM must still be investigated. In Hadjimitsis (2003), an unsupervised classification method using SOM was proposed for mapping land-cover changes. Preliminary results show that utilizing SOM for image classification is efficient and opens up significant area for further research.

Although SOM has been considered as an effective tool for clustering in remote sensing images, it is worth noting that the detection of cluster boundaries through the output space of the network is not a trivial task. Moreover, as described in Ji (2000) the process of segmentation of the trained neural map can affect representation of the classes. Consequently, the majority of the classification methods based on SOM carry out the cluster detection with the help of an image analyst that manually segments the neural map using a priori knowledge of the data and some alternative tool to make cluster visualization easier in SOM (Merkl & Rauber, 1997). In this way, the analyst interacts with the cluster analysis process by visualizing and interpreting the trained map. However, in unsupervised classification approaches that have as basic objective to discover the inherent structure of the data, the detection of clusters through SOM neurons must be done without user interference.

Faced with this scenario, this work presents a methodology for unsupervised classification of remotely sensed images that employs a two-level clustering approach, where the image data are first clustered using the SOM, and then the SOM is automatically segmented without any user intervention. This strategy considerably reduces the computational complexity of the data analysis and provides significant advantages over conventional unsupervised classification techniques, such as K-means and ISODATA. In this purpose, the prototype vectors of the SOM, which represent the original patterns of the image, are clustered through an agglomerative hierarchical clustering method generating a simplified dendrogram for the image data. The SOM neighborhood relation is used to constrain the possible merges in the construction of the dendrogram. In addition, spatial information measures and a cluster validation index calculated in a modified manner using the vector quantization property of the SOM are incorporated into merging criterion of neighboring clusters. The experimental results show examples of the application of the proposed classification methodology and compare its performance with the K-means algorithm.

2. The SOM

The SOM is certainly one of the main models of neural networks at present and is used in countless applications. Unlike other neural network approaches, the SOM is a type of neural net based on competitive and unsupervised learning (Kohonen, 1997). The network essentially consists of two layers: an input layer I and an output layer U with neurons generally organized in a 2-dimensional topological array. The input to the net corresponds to a p-dimensional vector, $\boldsymbol{x}$, generally in the space $\Re^p$. All of the p components of the input vector feed each of the neurons on the map. Each neuron i can be represented by a synaptic weight vector $\boldsymbol{w_i} = [w_{i1}, w_{i2},..., w_{ip}]^T$, also in the p-dimensional space. For each input pattern $\boldsymbol{x}$ a winner neuron, c, is chosen, using the criterion of greatest similarity:

$$\left\| x - w_c \right\| = \min_i \left\{ \left\| x - w_i \right\| \right\} \tag{1}$$

where $\|.\|$ represents the Euclidian distance. The winner neuron weights, together with the weights of the neighboring neurons, are adjusted according to the following equation:

$$w_i(t+1) = w_i(t) + h_{ci}(t)\left[x(t) - w_i(t) \right] \tag{2}$$

where t indicates the iteration of the training process, $\boldsymbol{x}(t)$ is the input pattern and $h_{ci}(t)$ is the nucleus of neighborhood around the winner neuron c.

Once the SOM training algorithm has converged the computed feature map displays important statistical characteristics of the input space, which can be summarized as follows:

i. Vector quantization: the basic objective of SOM is to store a large set of input vectors by finding a smaller set of prototypes that provides a good approximation to the input space.
ii. Topological ordering: the features map computed by SOM is ordered topologically. Similar input vectors are mapped close to each other, while dissimilar ones are mapped far apart.
iii. Density Matching: the SOM reflects the probability distribution of data in the input space. Regions in the input space in which the input patterns are taken with a high probability of occurrence are mapped onto larger domains of the output space, and thus have better resolution than regions in the output space from which input patterns are taken with a low probability of occurrence.

3. The proposed method

The methodology proposed in this work attempts to exploit the characteristics and properties of the SOM to perform the unsupervised classification of remotely sensed images. The key point of the proposed method is to perform the cluster analysis of the image through a set of SOM prototypes instead of working directly with the original patterns of the scene. For this, the classification task is carried out using a two-level SOM-based clustering approach.

The two clustering levels, which consist basically of the training and segmentation of the SOM, are illustrated in Fig. 1 and described in greater detail in the next subsections.

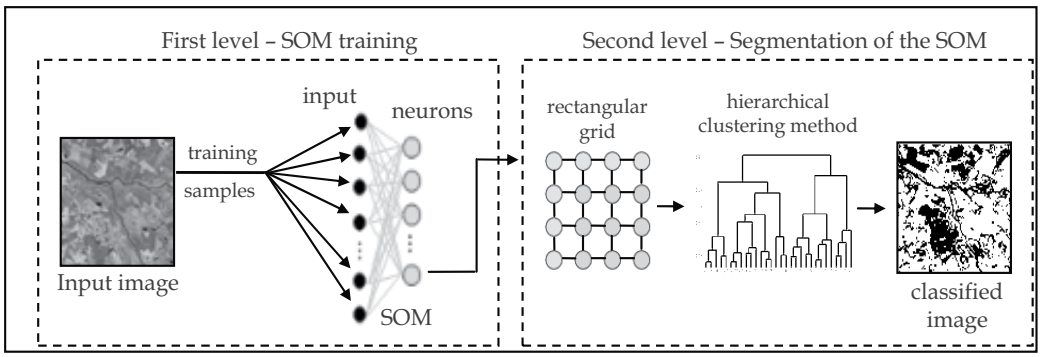


Fig. 1. Illustration of the proposed classification methodology

3.1 First level clustering – SOM training

At the first clustering level the SOM is used to map the original patterns of the image to a reduced set of prototypes arranged in a two-dimensional rectangular grid. The objective is to quantize and to represent the image patterns in a space of smaller dimension, seeking to preserve the probability distribution and topology of the input space. In this processing level, two important factors should be considered: the sampling process of the input image and the determination of the SOM training parameters.

3.1.1 Image sampling process

Unlike pixel by pixel approaches that only use the spectral information of individual points to find homogenous regions; the proposed method performs the sampling of the image through pixel windows. The idea is to incorporate in the classification process information about the neighborhood (context) of the pixels, considering that isolated pixels are not able to represent the majority of cover land patterns, especially in the case of images that have higher spatial resolutions. Even though the strategy of using information from neighboring pixels has a greater computational cost that the pixel by pixel approach, in the literature various works have shown that this effort is justified by a proportional increase in classification accuracy (Magnussen et al., 2004).

In this purpose, the sample windows are collected uniformly across the entire region of the image, without overlappings and at regular intervals. All of the samples are square and have the same size. The determination of the size of samples is not a process that can be easily automated. Depending on the characteristics of the image to be classified, windows that are too small might not capture the particular pattern of the land cover classes, while larger windows might include pixels from more than one class. Nevertheless, beginning with a size no smaller than 3×3, the user has within the proposed methodology a certain degree of flexibility to define the dimensions of the sample windows, and this is not therefore a critical point in the process. The proposed method (in the second level clustering) processes in a special way the SOM prototypes that represent sample windows which incorporate more than one class (heterogeneous prototypes).

3.1.2 SOM parameters

When the SOM is used some parameters must be specified to define the structure of the map and to control its training process. Based on the existing literature, on experimental tests,

and some peculiarities of the application of SOM on remotely sensed images, the proposed classification methodology utilizes the following parameters and criteria to train the SOM:

a. Linear initialization of weights: Although the random initialization is the most common form to initialize SOM weights, Kohonen defends the idea of linear initialization, since this permits the learning algorithm to directly enter the convergence phase, as well as avoiding undesirable skews in the array (Kohonen, 1997).
b. Batch training mode: The batch training mode makes the result of the SOM mapping insensitive to the presentation order of the input patterns. This can be done by adding the average of the contributions of all of the input patterns for each neuron. This average displacement has been leading to better results than the conventional SOM algorithm in which the neurons are updated every time a new input pattern is provided to the net.
c. Gaussian neighborhood function: Instead of utilizing the "bubble" type neighborhood proposed in the original SOM algorithm, the Gaussian type neighborhood is employed in the proposed method because it better emphasizes the clusters in terms of distances between neurons (Kohonen, 1997).
d. Rectangular array: The rectangular shape was chosen to organize the array of the neurons of the net, not just because it is the most commonly used, but also because it is the best at generating an image of the SOM features map, which will be described ahead.
e. Stopping criteria: To determine the number of training epochs the disorder measure "AUD" (Average Unit Disorder) proposed by Azcarraga (2000) was used in this approach. Essentially, AUD evaluates the degree of organization of the SOM during the training process through the differences of synaptic weights and geometric distance between neurons in the rectangular grid. The SOM is considered sufficiently trained when the AUD measure does not change in value by more than a certain threshold.
f. Size of the map: The dimension of the map is one of the free parameters of SOM that particularly depends on the input image and the objectives of the classification. If the objective is to detect all of the patterns in the image, including those with low probability of occurrence, large-sized maps must be employed in the analysis; in the opposite case, if the interest is concentrated only on the predominant patterns in the scene, a smaller-sized SOM can be utilized. Nevertheless, the performance of the proposed classification methodology is not significantly affected if sufficiently large sizes for the SOM are utilized. While maps with larger dimensions than are necessary have a larger quantity of inactive neurons, this event is not prejudicial within the proposed methodology. Just as the method deals with the heterogeneous prototypes in an adequate manner, the same occurs with inactive prototypes.

3.2 Second level clustering – segmentation of the SOM

Unlike in other applications in which the SOM is used as a visual aid tool to indicate cluster tendencies, in this work the detection of clusters through trained SOM is executed without user intervention. Thus, after completion of training performed by the SOM, the second level of the proposed methodology basically consists of segmenting the SOM output map using an additional clustering method.

Several approaches have been proposed for clustering the neurons of a trained SOM (Costa & Netto, 1999; Costa & Netto, 2001; Gonçalves et al., 2005; Gonçalves et al., 2006). Approaches that make use of conventional cluster methods, such as K-means and

hierarchical clustering, are those most commonly applied (Sezgin et al., 2004; Vesanto & Alhoniemi, 2000; Wang, 2002). However, it is worth noting that methods based on K-means algorithm are only feasible for hyper-spherical-shaped clusters and approaches based on classical hierarchical clustering only use inter-cluster distance information to merge the nearest neighboring clusters. Seeking to overcome the limitations of these approaches, Wu & Chow (2004) proposed then a hierarchical algorithm for clustering the SOM that uses more information about the data in each cluster in addition to inter-cluster distances. To determine which pair of clusters to be merged the proposed algorithm locally uses the CDbw clustering validity index (Composing Density Between and Within Clusters), presented in Halkidi & Vazirgiannis (2002), which allows to incorporate the inter-cluster and intra-cluster density into merging criteria in addition to distance information. Wu and Chow showed that the algorithm proposed by them clusters data better than the classical clustering algorithms on the SOM.

For this reason, the classification methodology proposed in this work employs a hierarchical method for clustering the SOM that also uses the CDbw clustering validity index to decide which pair of clusters to be merged. However, instead of using the original data to perform the calculation of the CDbw index (as is made in Wu & Chow (2004)), the algorithm applied here executes the computation of the CDbw index using the own prototype vectors of the trained SOM (which represent the image original data). Furthermore, the adopted merging criterion also incorporates two spatial information measures and still uses neighborhood relation between the neurons in the SOM grid to constrain the possible merges.

3.2.1 Prototypes filtering

Seeking to guarantee even greater efficiency in the segmentation process of the trained SOM, before the hierarchical clustering algorithm is applied, the proposed method filters two types of prototypes that generally appear in the mapping of image patterns through SOM. These prototypes are called inactive and heterogeneous.

The inactive prototypes correspond to the neurons that have null activity in the SOM competitive learning process, i.e., they are not associated with any input patterns. These prototypes are simply discarded of the analysis.

Heterogeneous prototypes are those that have a high degree of spectral-textural heterogeneity and are normally associated with input patterns that incorporate more than one land cover class. Most of the time, these patterns correspond to transition regions between land cover classes present in the image and are captured in consequence of the sampling through pixel windows. The heterogeneous prototypes are not considered by the hierarchical clustering method applied to segment the SOM. The objective of excluding these prototypes is to prevent them (and consequently the input patterns associated with them) from being erroneously attributed to one of the classes that are part of them. Heterogeneous prototypes can be seen as noisy or divergent patterns, and if they are not filtered the hierarchical method can incorporate them in the clusters that will be produced or retain them in separate clusters. Because of this, the input patterns associated with these heterogeneous prototypes are classified only at the end of the analysis, considering the neighboring pixels that have already been labelled.

In the proposed classification methodology the spectral-textural heterogeneity degree of each SOM prototype is computed from Haralick's co-occurrence matrix (Haralick et al., 1973). Since the weight vectors of the SOM prototypes have the same dimensions as the input patterns (that in this case are pixel windows), it makes it possible to generate an image

of each prototype of the net and to calculate the co-occurrence probability of all pairwise combinations of grey levels in each one of them. The energy (sometimes called uniformity) was the measure chosen to calculate the spectral-textural heterogeneity of each prototype from co-occurrence matrix. This measure, described through the equation (3), gets values next to 1 when the area of interest presents uniform texture (similar grey levels), and values that tend to zero when the area is not uniform.

$$ENE = \sum_i \sum_j P(i,j)_{d,\theta} \tag{3}$$

where $P(i,j)_{d,\theta}$ is the co-occurrence probability of two grey levels i and j, separate to a distance d in the direction θ. The prototypes whose *ENE*'s satisfy the relationship given below are considered heterogeneous and are consequently filtered:

$$ENE < \mu_{ENE} - \frac{1}{2}\sigma_{ENE}. \tag{4}$$

Here μ_{ENE} and σ_{ENE} are, respectively the average and the standard deviation of the *ENE*'s of all of the SOM prototypes.

The processing time consumed by this filtering step is compensated not just by the reduction of the number of SOM prototypes that will be analyzed by the hierarchical clustering method, but mainly by a possible increase in the precision of the image classification. In addition, inactive and heterogeneous prototypes can act as "interpolation units" or "borders" in the SOM grid, facilitating the separation of the clusters.

3.2.2 Hierarchical clustering method

If an agglomerative hierarchical clustering method is applied to the SOM neurons it must respect the topological relationships of the neural net output space. Unlike traditional hierarchical clustering methods, which consist of comparing all of the pairs of objects to decide on a fusion, the approach utilized in this work verifies the possibility of fusions only between adjacent neuron pairs in the SOM grid. As shown in Fig. 2, given a $m \times n$ neurons grid, the neuron in position (3,2), for example, only has possibility of fusion with neighboring neurons in positions (2,1), (2,2), (2,3), (3,1), (3,3), (4,1), (4,2) and (4,3). This approach significantly diminishes the processing cost of the hierarchical clustering method applied.

Other important characteristic of the hierarchical method utilized for clustering the SOM is in its merging criterion. It utilizes more information about the pair of clusters than traditional methods. As mentioned earlier, the CDbw clustering validity index which allows to incorporate inter-cluster and intra-cluster density information is used in conjunction with two spatial information measures to determine which cluster pair to be merged.

Although the CDbw index has a reasonable computational complexity in comparison with others clustering validity indexes (Halkidi & Vazirgiannis, 2002), the local use of it to decide about the cluster fusion added to the large data volume that normally arises from remote sensing images may require a considerable processing time for clustering the SOM. For this reason, seeking to reduce the processing volume, instead of calculating the CDbw index directly on image original data the method proposed here performs the calculation of it using the own SOM prototype vectors (which represent the image original data).

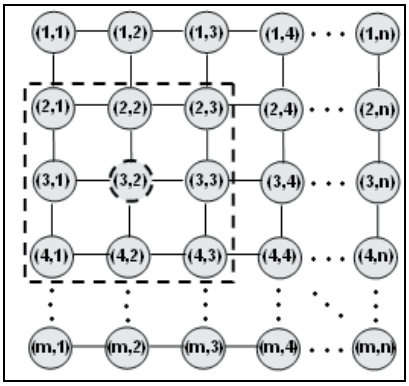


Fig. 2. Adjacent neurons in the SOM grid

Even though the SOM performs a good approximation of the input space, it is true that the strategy of applying a validation index on the network prototypes instead of applying them directly to the original pattern can cause some numerical difference in the results. To diminish possible approximation errors, the proposed method modifies the way that CDbw index is calculated. The index is calculated by utilizing not just the weight vectors of the SOM prototypes, but also the activity level of each one of them. The activity level of one prototype is equal to the number of input patterns that are associated to it by the SOM mapping process.

An example of the modifications applied to the calculation of CDbw index can be described through the equation (5). This formula represents the calculation of the standard deviation vector (σ) of a data cluster *C*:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \mu\right)^2} \tag{5}$$

where x_i is the data belonging to the cluster *C*, n is the number of data points in *C*, and μ represents the sample mean of the same cluster.

Applying the strategy of modifying the index calculation, the computation of the standard deviation vector of a data cluster *C*, equivalent to equation (5), becomes formulated in the following way:

$$\sigma^{som} = \sqrt{\frac{1}{n}\sum_{i=1}^{m} h(w_i).\left(w_i - \mu^{som}\right)^2} \tag{6}$$

in which wi is the SOM prototype, h(wi) is the activity level of prototype wi, m is the number of SOM prototypes that represent data from cluster C, n is the number of data points in C, and μsom represents the sample mean of the cluster C, which is computed as follows:

$$\mu^{som} = \frac{1}{n}\sum_{i=1}^{m} h(w_i).w_i \tag{7}$$

Whereas the calculations of σ and μ use all of the points belonging to *C*, σ^{som} and μ^{som} only uses the SOM prototypes. Therefore it can be observed that σ^{som} is an approximate

calculation of σ, but with a lower processing cost, since the number of data points n is generally much smaller than number of SOM protoypes m. The inclusion of the activity levels $h(.)$ of the prototypes in the calculation of σ^{som} helps to reduce the effect of quantization error caused by SOM mapping.

The strategy of using SOM prototypes and its activity levels instead of the original data is applied to all the steps of the CDbw index algorithm. This modified version of the CDbw index is therefore locally applied to each neighboring cluster pair at all of the levels of the dendrogram produced by hierarchical method used for clustering the SOM. The pair of clusters with the lower value of the CDbw is considered that one with the strongest tendency to be clustered. The values of the CDbw index for all clusters pairs (i,j) (designated as $CDbw_{ij}$) are normalized within the interval [0,1].

The spatial information incorporated in the merging criterion is calculated through two indices, denominated *spatial boundary index* and *spatial compactness index*. These indices, projected by Marçal & Castro (2005), are computed here from classified image using the SOM prototype clusters in each level of the dendrogram generated by hierarchic method.

The spatial boundary index (B_{ij}) calculates the boundary length between all class pairs (i,j) considering eight neighbors for each pixel (four adjacent and four oblique). Its formula is given as follows:

$$B_{ij} = 1 - \frac{1}{2}\left(\frac{b_{ij}}{\sum_{k=1(k\neq i)}^{N} b_{ik}} + \frac{b_{ij}}{\sum_{k=1(k\neq j)}^{N} b_{kj}} \right) \tag{8}$$

where b_{ij} is the number of boundary counts for the class pair (i,j) and N is the number of classes (SOM prototypes) in the dendrogram level that is being analyzed. This index provides values within the interval [0,1], with lower values when the boundary between the pair of classes is considerable. According to idea behind this index, two classes with an extensive common boundary must be more inclined to merge than classes with very little or no common boundaries (Marçal & Castro, 2005).

The spatial compactness index (C_{ij}), defined through the equation (9), is based on the number of self-boundary counts (b_{ii}) for each class. This index penalizes the merger of compact classes. The index C_{ij} also results values between 0 and 1, with lower values when the class is composed of isolated pixels.

$$C_{ij} = \frac{1}{2}\left(\frac{b_{ii}}{b_{ii} + 6\sum_{k=1(k\neq i)}^{N} b_{ik}} + \frac{b_{jj}}{b_{jj} + 6\sum_{k=1(k\neq j)}^{N} b_{jk}} \right) \tag{9}$$

The merging criterion adopted here for clustering the SOM establishes that the pair of prototype classes (i,j) with the lowest value resultant of the average computed between $CDbw_{ij}$, B_{ij} and C_{ij} must be selected for merger.

At the end of this processing level there is a dendrogram that shows in a hierarchical way the similarity levels between the SOM prototypes. Remembering that the SOM prototypes

represent all of the input patterns; the obtained dendrogram reflects the relationships that exist between the original image data, and it can be considered therefore as a simplified dendrogram of the image.

3.2.3 Image classification

At each level of the dendrogram produced by the hierarchical method there is a different cluster configuration for the SOM prototypes that can be utilized to represent the classes by which the original image will be classified. Thus, as with any hierarchical clustering method, some decision criterion needs to be applied in order to verify which level of the dendrogram represents the ideal cluster configuration (or the ideal number of classes).

According to Vesanto & Alhoniemi (2000), the most typical solution is to cut the dendrogram where there is a large distance between two merged clusters. However, this approach ignores the fact that the within-cluster distance may be different for different clusters. Therefore, the decision criterion used here consists of applying once more the CDbw index (in its modified version) as a function of the number of clusters at all levels of the dendrogram, and choosing the level at which the index presents its optimum value among the values obtained.

Having defined the dendrogram level that presents the best cluster configuration for the SOM prototypes, and consequently for the image patterns, the SOM is then labeled. According to the proposed methodology, each discovered land cover class will be therefore represented by a group of SOM prototypes and not just one single prototype as occurs in the majority of the partitional clustering methods, such as *K*-means algorithm.

In order to classify the image, pixel windows are collected from the original image with equal dimensions from the training sample and are compared to all of the SOM prototypes (including also the heterogeneous prototypes). This comparison is performed through the distances calculated between the pixel windows and each of the prototypes. The central pixel of the pixel window receives the label of the prototype that has the least distance from it. The image is then entirely run through until all of the pixels have been classified.

In the end, the results of the classification process performed are improved. The pixels in the image that receive the label of the heterogeneous prototypes class are reclassified. Each one of these pixels is compared to its neighboring pixels in the image that have not been labeled by the heterogeneous prototype class, and it receives a new label that will be the same as the neighbor that has the smallest distance from it in terms of spectral features.

4. Experimental results

This section shows examples of the application of the proposed classification methodology on two test images. The results are compared with those obtained by applying the K-means algorithm to the same images. It is worth emphasizing that the comparisons performed here refer to the results obtained by two algorithms with significantly different principles and characteristics and therefore this work do not intend to decide which method is better, but to analyze the applicability and the benefits of the proposed classification methodology.

4.1 Experiment 1

This experiment was performed on a segment taken from a CBERS image (provided by National Institute for Space Research, INPE, Brazil). It has an IFOV of 20 m and is composed of three bands in the visible spectrum. This imagery was acquired on 4 August 2005 and

shows part of the city of Rio de Janeiro, Brazil. The scene (392×394 pixels) presents four large land cover classes: urban area, vegetation, sandy soil and water. Fig. 3 shows a color composite of the segment of the original image (denominated here as test image 1).
Application of the proposed methodology was initially performed with a sampling process of the scene. Sample windows of size 7×7 were collected uniformly across the entire region of the image, without overlapping and at regular 10 pixel intervals, resulting in a total of 1521 samples obtained without user intervention.

Fig. 3. Color composite of the test image 1

A SOM composed of 64 neurons arranged in a 8×8 rectangular grid was trained with all of the samples previously collected. As mentioned earlier, the weight vectors of each neuron have the same dimensions as the input patterns (that in this case are pixel windows of size 7×7), it makes it possible to generate an image of each SOM prototype. Fig. 4a shows the images of each SOM prototype arranged in the rectangular grid after the training. By means of them it is possible to visualize the properties of the mapping produced by the neural net. The four land cover classes present in the original image appear in the form of clusters on the SOM grid image. In the upper right corner the prototypes associated with the water pattern are found, in the upper left corner the prototypes associated with the vegetation pattern are present, in the lower right those that correspond to urban area are found, and in the lower left corner there are prototypes associated with sandy soil class. If we observe the original image, we can verify that the sandy soil class occupies the least area in the scene. Because of this, since the SOM reflects the probability distribution of the input data, the prototypes associated with sandy soil class (in nearly white tones on the SOM grid image) are present in a lesser number than the other prototype classes. SOM's topological ordering property can also be seen. The prototypes of the water and sandy soil classes are isolated to one another in the SOM grid, because the input patterns (samples) corresponding to these two classes are the least spectrally similar in comparison with the spectral attributes of the other two land cover classes. While the SOM grid image is useful for visualizing the SOM mapping, it must be pointed out that in the method proposed here no interaction by the user is done on it.
After the SOM training, the prototype filtering process was applied. In this experiment, 4 SOM prototypes presented null activity, and 8 presented a high degree of spectral-textural heterogeneity, given that its *ENE* values exceeded the threshold defined in the equation (4). Thus, of the 64 total SOM prototypes, 12 of them were filtered, leaving 52 to be analyzed.

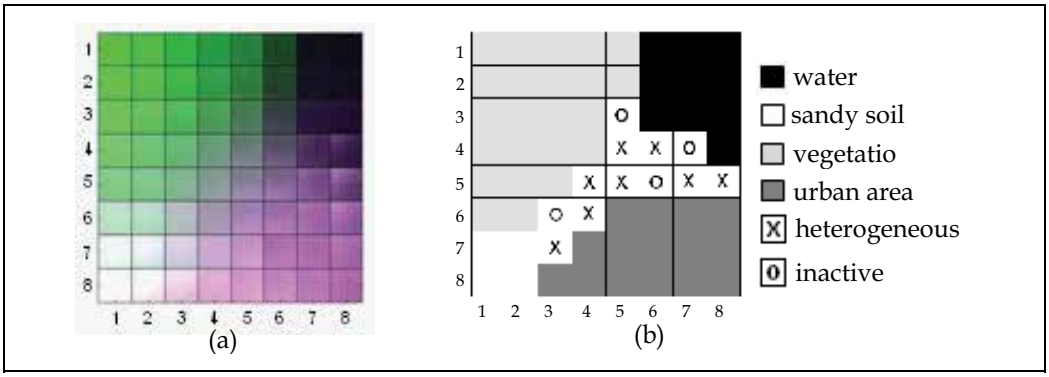


Fig. 4. (a) SOM grid image obtained from test image 1. (b) Classified SOM grid.

In sequence, the agglomerative hierarchical method was applied on the filtered SOM prototypes. Consequently, a dendrogram consisting of 52 levels was generated, each level with a different configuration of SOM prototype clusters. Fig. 5a shows the dendrogram obtained using the merging criterion employed by the proposed method. As can be viewed with the aid of the SOM grid image (Fig. 4a), the SOM prototypes corresponding to four land cover classes present in the image were appropriately clustered in a hierarchical way at all of the stages. Attempting to confirm the efficiency of the implemented merging criterion, the single linkage method also was utilized for clustering the SOM prototypes, but using only the spectral information of the SOM prototypes and respecting its topological relations in the grid. The single linkage method is the most popular method of hierarchical clustering. Single linkage defines the distance between any two clusters as the minimum distance between them (Xu & Wunsch II, 2005). The Fig. 5b shows the dendrogram produced by the single linkage method. For the test image 1 this is not a satisfactory result. The SOM prototypes corresponding to sandy soil class (located in the positions (7,1), (7,2), (8,1), and (8,2) of the SOM grid) were all merged with SOM prototypes corresponding to area urban class at the earliest levels. This occurs because the prototypes of these two land cover classes are spectrally very similar, and as the merging criterion of the single linkage method is exclusively based on minimum distance they are erroneously grouped into a single cluster. Since the merging criterion utilized by the proposed method uses more information about the data in addition to inter-cluster distances, it does not allow that prototypes of the sandy soil and area urban classes be merged at the initial levels. Although the spatial boundary index of these two classes has presented a low value (given that the spatial boundary between them is considerable), the CDbw and spatial compactness indices presented relatively high values, preventing thus an early fusion of the two classes.

Continuing with the application of the proposed classification methodology, the modified version of the CDbw cluster validation index was applied at all of the levels of the dendrogram to determine which cluster configuration of the 52 that were obtained is ideal. Fig. 6 shows the validation index values in the modified and the original form. The graph only shows the values between levels 38 and 51 of the dendrogram. For levels lower than 38 the index values are smaller than those on the graph, or even null, due to the existence of levels composed of clusters with a single prototype. The number of clusters or classes from level 38 to 51 decreases from 15 to 2, respectively.

As expected, the strategy of modifying the cluster validation index computation has approximation errors, but the variation of their values occurs in a similar way to the original

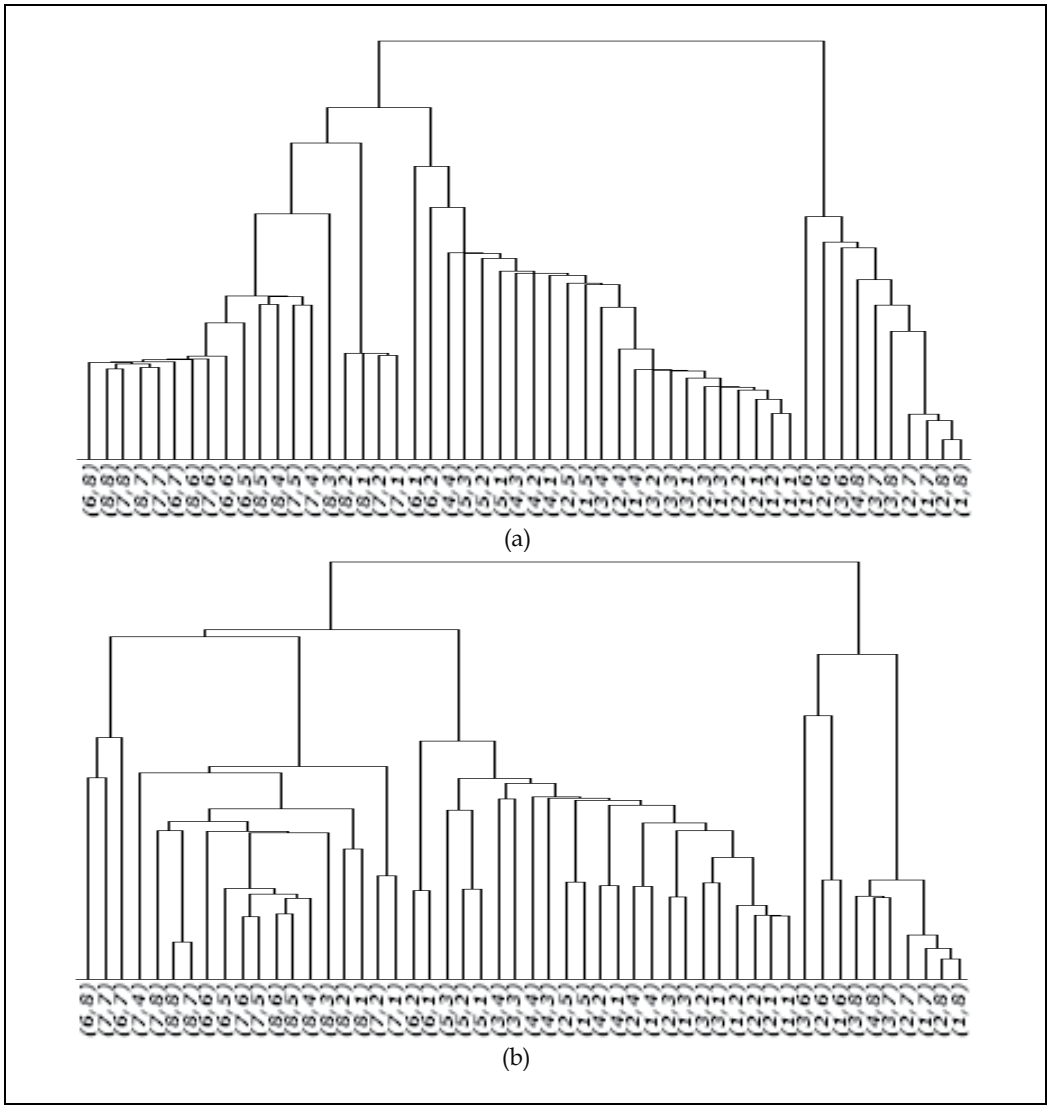


Fig. 5. Dendrograms of the SOM prototypes using different merging criteria. (a) Proposed merging criterion. (b) Single linkage method using only spectral information. The ordered pairs correspond to the positions of the SOM prototypes in the 8×8 grid.

version, not compromising the decision making process regarding the ideal number of data clusters. The main advantage of using the modified version of the index is its processing time. In this experiment, while the time to calculate the CDbw index in the original form (for the entire dendrogram) was 478 seconds, the modified version needed only 26 seconds. It is important to note that this significant difference occurs due to the volume of data considered for each of the ways to calculate the index. Calculation of the CDbw index in its original form was executed at each level of the dendrogram considering the 1521 samples collected in the image, while the modified version calculation considered only 64 SOM prototypes and their respective activity levels.

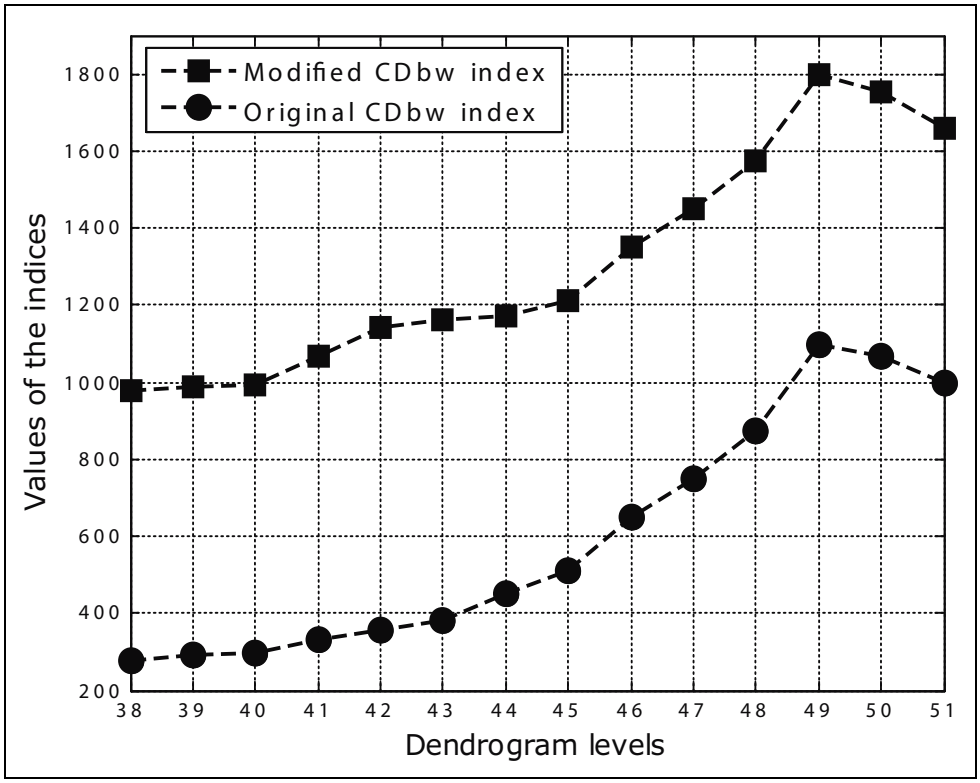


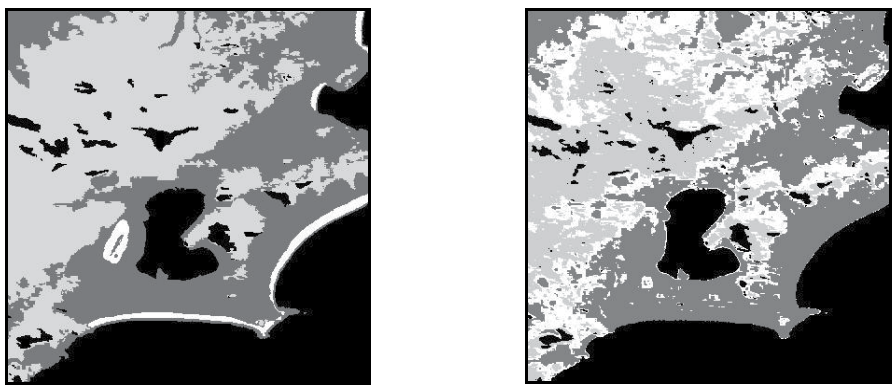
Fig. 6. CDbw index values in the modified and the original form

For the CDbw index, the greater its value, the better the result. Considering this, of all of the dendrogram levels obtained in this experiment, level 49 is the one that has the greatest value for the index (as can be seen in Fig. 6), and thus, the best cluster configuration for the SOM prototypes. Fig. 4b shows the SOM grid image classified in accordance with the clusters (classes) of level 49. Compared to the SOM grid image in Fig. 4a, it can be seen that the SOM prototypes are grouped into 4 classes, adequately corresponding to the 4 large land cover classes present in the image (water, sandy soil, vegetation, and urban area). The squares marked with "o" and "×" are, respectively, the inactive and heterogeneous prototypes that were discarded in the filtering process of the proposed method. It is interesting to see that these prototypes act as "interpolation units" and certainly contribute to the class separation process.

To finish, the original image was classified comparing 7×7 pixel windows with all of the labeled SOM prototypes (except the inactive prototypes). According to the proposed method, a total of 7643 pixels associated with the heterogeneous prototype class were reclassified using the neighboring pixel class that had the least (spectral) distance. Fig. 7a shows the classification result of test image 1 using the proposed methodology (the 4 classes are represented with the same colors used in Fig. 4b).

Fig. 7b shows the result of the image classification by the *K*-means algorithm considering the number of classes as being equal to 4. The algorithm was executed using five different initializations for the centroids of each of the classes and a maximum number of iterations equal to 100.

Unlike the proposed method, the *K*-means algorithm did not discriminate the sandy soil pattern in the image and classified the vegetation areas in two categories. The sandy soil pattern is only differentiated by the *K*-means algorithm if the number of clusters (*k*) is set to greater or equal to 7. Moreover, performing a visual analysis of the classification results obtained by the two methods it is possible to verify that the image classified by proposed methodology has a superior visual aspect. While the classification of the image by the *K*-means algorithm appears more dotted, the image produced by the proposed method is more homogenous in all of the classified areas. The difference in appearance is certainly due to the way the two methods extract the information from the image. While the *K*-means algorithm applied here uses a pixel-by-pixel approach, the proposed method works with pixel windows which incorporate the neighborhood information of pixels. In both classification results some vegetation areas were erroneously classified as water pattern due to topographic shadowing effects.



(a) (b)

Fig. 7. (a) Test image 1 classified by the proposed method. (b) Test image 1 classified by the K-means algorithm considering k=4.

Attempting to perform a less subjective analysis of the image classified by the proposed method, and considering the absence of terrestrial truth for the test image, the present work performed the classification of the test image in a supervised manner and considers these results as a reference (or "true"). The supervised classification, considering the 4 predominant classes in the image, was done by means of a multilayers *Perceptrons* neural net. This class of neural nets has been widely employed to perform supervised classification of remotely sensed images (Özkan & Erbek, 2003).

The Kappa agreement index was calculated by comparison of the image classified by the proposed method and the reference image (resulting from the supervised classification). Considering that the Kappa index reached the value of 0.89, we can consider that the classification result of the test image 1 by method presented in this work was satisfactory.

4.2 Experiment 2

The second experiment was performed on a segment taken from an IKONOS image. It has an IFOV of 4 m and is composed of three bands in the visible spectrum. The segment used in

this experiment has 350 lines and 384 columns. This imagery shows irrigation pivots in the region of Andaraí in the state of Bahia, Brazil, and presents six land cover classes: sparse vegetation, forest, two types of bare soil, and two types of growing crop. Fig. 8 shows the segment of the original image (denominated here as test image 2).

Following the procedures of the proposed method, 1292 sample windows of size 9×9 were collected from test image 2 and used to train a SOM composed of 144 neurons arranged in a 12×12 rectangular grid. Fig. 9a shows the SOM grid image after training in which it is possible to visualize prototype clusters that correspond to the land cover classes present in the original image. As in experiment 1, topological ordering and density matching properties of the mapping produced by the SOM can be seen. Land cover classes with similar spectral attributes are mapped to neighboring regions of the two-dimensional output grid and those that occupy larger areas in the original image are mapped to a larger number of prototypes of the grid.

Fig. 8. Color composite of the test image 2 (provided by Engesat/Brazil, © Space Imaging)

After the SOM training, the prototype filtering process was applied. In this experiment 15 SOM prototypes presented null activity, and 14 presented a high degree of spectral heterogeneity. Thus, of the 144 total prototypes, 29 of them were filtered, leaving 115 SOM prototypes to be analyzed.

In sequence, the agglomerative hierarchical method was applied on the filtered SOM prototypes. A dendrogram consisting of 115 levels was generated, each level with a different configuration of SOM prototype clusters. The modified version of the CDbw index was applied at all of the levels of the dendrogram to determine which of the 115 cluster configurations that were obtained was ideal. Level 110 had the greatest value for the validation index, and therefore, the best cluster configuration for the SOM prototypes. Fig. 9b shows the SOM grid image classified in accordance with the clusters (classes) of level 110. Compared to the SOM grid image in Fig. 9a, it can be seen that the SOM prototypes are grouped into 6 classes, adequately corresponding to the 6 land cover classes present in the original image.

As in experiment 1, it is important to point out that the correct segmentation of the SOM grid is only reached due to the merging criterion employed by the proposed method. If only the inter-cluster distances were used to decide about the cluster fusion, certainly different

prototype classes would be erroneously grouped into a single cluster in the SOM grid. For example, the SOM prototypes corresponding to forest and growing crop 1 classes are spectrally very similar. As can be viewed in the SOM grid image (Fig. 9a), they are close to one another in the upper left corner of the grid. These two prototype classes are correctly discriminated by the proposed method because the indices used in the merging criterion (especially the spatial boundary and spatial compactness indices) presented high values, preventing thus an early fusion of the two classes. In fact, as can be seen in the original image (Fig. 8), the areas corresponding to forest and growing crop 1 classes (situated respectively in the lower left corner and in the upper right corner of the image) do not have common boundary and are spatially compact. It is fundamental to note also that the adequate segmentation of the SOM is facilitated by the prototype filtering process applied on it. The inactive and heterogeneous prototypes that were discarded in the filtering process act as "borders" in the SOM grid contributing considerably to the separation of the clusters.

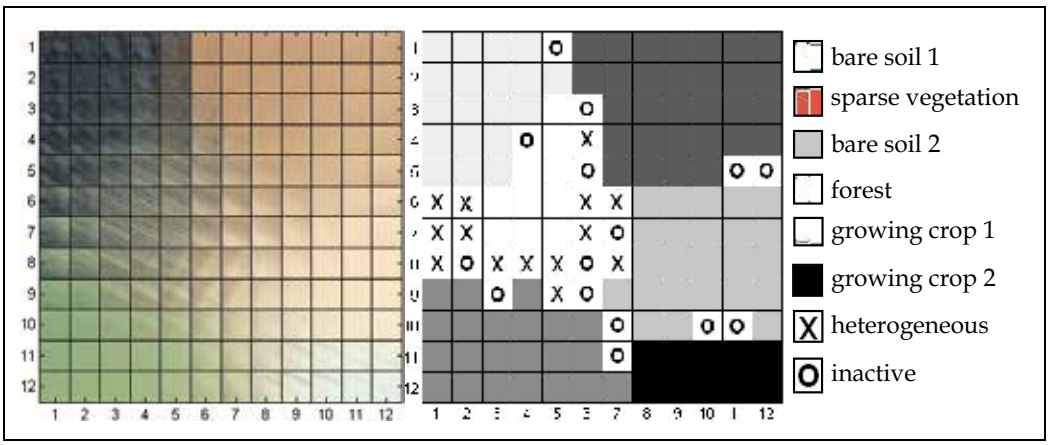
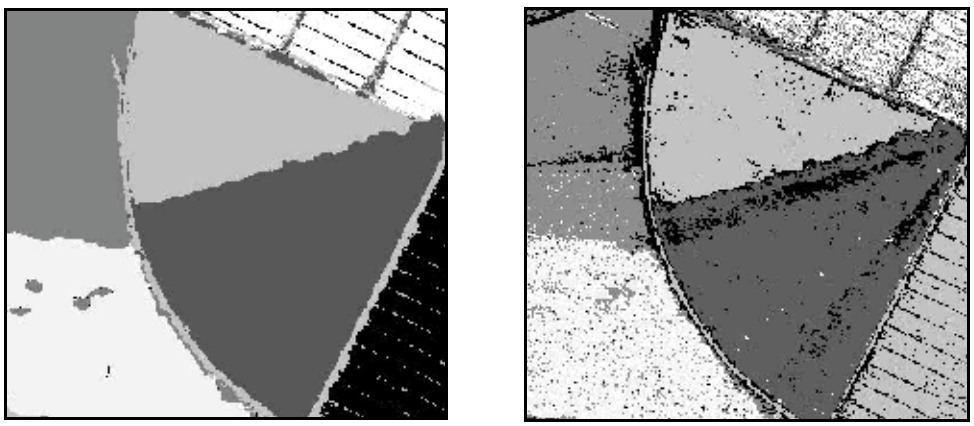


Fig. 9. (a) SOM grid image obtained from test image 2. (b) Classified SOM grid.

Fig. 10a shows the classification result of test image 2 using all of the labeled SOM prototypes, except the inactive prototypes (the 6 classes are represented with the same colors used in Fig. 9b). A total of 19717 pixels associated with the heterogeneous prototypes were reclassified using the neighboring pixel class that had the least (spectral) distance. Fig. 10b shows the result of the classification of the same image by K-means algorithm setting the number of classes at 6.

Performing a visual analysis of the results obtained by the two methods it is possible to clearly verify a large difference between the two classifications. The proposed method adequately classified the different land cover patterns present in the original image without any significant confusion between them. On the other hand, the K-means algorithm did not correctly discriminate the patterns and presented a relatively erroneous classification. It did not differentiate the growing crop 2 class (in the lower right corner of the image) from one of the bare soil patterns, and it also confused the forest class with the growing crop 1 class. Just as in the classification results of experiment 1, the classification of the test image 2 by the K-means algorithm were more speckled, while the image produced by the proposed method is considerably more homogeneous in all of the classified areas. This difference in appearance is more accentuated in this experiment because test image 2 presents a higher spatial resolution than test image 1. In test image 2, the land cover classes have more well defined

textural features, and because the proposed method uses the neighborhood information of the pixels, it produces a superior result to the K-means algorithm, which uses a pixel-by-pixel approach.



(a) (b)

Fig. 10. (a) Test image 2 classified by the proposed method. (b) Test image 2 classified by the K-means algorithm considering k=6.

Although the quality of the classification produced by the proposed method is visually superior, the image used in this experiment also was classified in a supervised manner and the result of this classification was used as a reference to evaluate the results obtained by the two methods. As in experiment 1, the supervised classification was done by means of a multilayers Perceptrons neural net. The Kappa index was then calculated individually for the images produced by the proposed method and by the K-means algorithm. Confirming the visual impression of the results, the image classified by the K-means algorithm had a Kappa value of 0.74, while the image classified by the proposed method had a value equal to 0.92.

4.3 Processing time

Table 1 shows the processing times spent in the two experiments by the main stages of the proposed classification methodology. All of the experiments presented here were performed with MATLAB software and a microcomputer with an AMD Athlon™ 2600+ 1,91GHZ processor and 1 GB of RAM. The global time consumed to perform the classifications of test images 1 and 2 were, respectively, 143 and 236 seconds.

A comparison between the processing times consumed by proposed methodology and the K-means algorithm to classify the test images is not appropriate here. As previously mentioned, the two methods have principles and characteristics that are very different, which should be considered in the evaluation of its calculation complexities.

An important difference between the methods is in the quantity of classes they analyze in order to classify the image. The proposed method evaluates different cluster configurations for the data, whereas the K-means algorithm performs the classification of the scene only for a single quantity of classes defined a priori. In the experiment 1, for example, the time spent by the proposed method to evaluate 52 different cluster

configurations was 26 seconds, using SOM prototypes to represent the image data and the CDbw validation index with modified calculation. If we were to use a similar strategy to the K-means algorithm, i.e., perform different classifications of the image varying the number of classes (k) and then apply the CDbw validation index to determine which was better, the processing time would be so high that the strategy would be considered impracticable. It is enough to take into consideration the time spent in applying the CDbw index to evaluate the classification produced by the K-means algorithm shown in Fig. 7b. It took 979 seconds, far more than the time spent by the proposed method to evaluate diverse cluster configurations for the image. Even though the CDbw validation index has a good level of computational complexity in comparison to others, the high number of points in remotely sensed images generates a very high processing volume to execute its calculations.

Taking these observations into account and also considering the possible benefits of the techniques and procedures employed in the methodology proposed, we can conclude that the processing time consumed by the method presented here is perfectly admissible, certifying its application viability.

Stages	Consumed time (in seconds)	
	Experiment 1	Experiment 2
Sampling	8	7
SOM training	34	56
Prototypes filtering	2	4
Agglomerative hierarchical analysis	29	96
Clusters evaluation	26	42
Image classification	44	31
Total	143	236

Table 1. Processing time consumed by the proposed method

5. Conclusions and final considerations

In this work, an unsupervised method of classifying remotely sensed images based on clustering of the SOM was presented. The key point of the proposed method is to perform the cluster analysis of the image through a set of SOM prototypes instead of working directly with the original patterns of the scene.

The proposed methodology presents benefits and potentialities that make it as a differentiated alternative to perform the unsupervised classification of remotely sensed images. Among these, we can point out:

1. The two-level clustering approach based on SOM significantly reduces the computational load of the classification process, making it possible to use methods that have not normally been considered computationally viable for the processing of

remotely sensed images, such as hierarchical clustering methods and cluster validation indices.

2. Another benefit in executing the clusters analysis of the image through a set of SOM prototypes is noise reduction. The SOM prototypes are local averages of the data, and therefore less sensitive to random variations than the original data of the image.
3. The method does not require a previous definition of the number of classes to perform the classification of the image. It does not occur in the majority of the conventional unsupervised classification methods, such as K-means.
4. The distributed representation of the classes by means of prototype groups gives the method the potential to discover geometrically complex and varied data clusters. Methods such as K-means use a single prototype (centroid) to represent each class and because of this are only capable of adequately detecting clusters that have spherical formats.
5. The simple use of pixel windows allows contextual information to be included without any explicit calculation of measure for it. This approach contributes to the quality of the resulting classification.
6. The proposed method has only two parameters that must be defined by the user (the size of the samples and the number of SOM neurons), and despite this it is very robust as far as these choices. Conventional methods (such as K-means and ISODATA) are very sensitive to a quantity of user-dependent parameters.
7. The utilization of an agglomerative hierarchical clustering method for clustering the SOM allows the user to observe the relationships that exist between the land cover patterns existing in the image at different cluster levels. It can be very helpful in applications where the structure of the information present in the image is not clearly known.
8. The proposed method employs an efficient merge mechanism that incorporates more information about the data in each cluster. Traditional clustering methods use only inter-cluster distance information to decide about the merging of clusters.
9. The method can determine, without any user intervention, the ideal number of clusters or classes for the image.

In addition to the test image utilized in the experiments shown here, the proposed method has also been applied to other high and medium resolution images, with satisfactory results. However, it is important to observe that the performance of the proposed method highly depends on the quality of the SOM mapping. The use of the procedures and techniques presented here assumes that the SOM has been successfully trained. In this way, questions about topological distortions and density approximation produced by the mapping of the net can be better investigated. Moreover, SOM variants were not used in this work because we wanted to select the most frequently used version of SOM. Therefore, new SOM developments could naturally be applied to further increase the potential of the proposed classification methodology.

6. References

Azcarraga, A. P., (2000) Assessing self-organization using order metrics, Proceedings of IEEE-INNS-ENNS International Joint Conference on Neural Networks, July, Como, Italy, 6:159-164.

Ball, G., and Hall, D., (1967) A clustering technique for summarizing multivariate data, Behavior Science, 12:153-155.

Bandyopadhyay, S., (2005) Satellite image classification using genetically guided fuzzy clustering with spatial information, International Journal of Remote Sensing, 26(3): 579-593.

Costa, J. A. F., and Netto, M. L. A. (1999) Estimating the Number of Clusters in Multivariate Data by Self-Organizing Maps. Intl. Journal of Neural Systems, vol. 9, pp. 195-202.

Costa, J. A. F., and Netto, M. L. A. (2001) Clustering of complex shaped data sets via Kohonen maps and mathematical morphology". In: Proceedings of the SPIE, Data Mining and Knowledge Discovery. B. Dasarathy (Ed.), Vol. 4384, pp. 16-27.

Duda, T., and Canty, M., (2002) Unsupervised classification of satellite imagery: choosing a good algorithm, International Journal of Remote Sensing, 23(11): 2193-2212.

Gonçalves, M. L., Andrade Netto, M. L., Costa, J. A. F., Zullo Jr., J., (2005) Automatic Remotely Sensed Data Clustering by Tree-Structured Self-Organizing Maps, Proceedings of IEEE International Geoscience and Remote Sensing (IGARSS'05), 25-29 July, Seoul, Korea, 4 p.

Gonçalves, M. L., Andrade Netto, M. L., Costa, J. A. F., Zullo Jr., J., (2006) Data Clustering using Self-Organizing Maps segmented by Mathematic Morphology and Simplified Cluster Validity Indexes: an application in remotely sensed images, Proceedings of IEEE International Joint Conference on Neural Networks, July, Vancouver, p. 8854-8861.

Hadjimitsis, D.G, Evangelou I., Lazakidou A., and Clayton C.R.I., (2003) Unsupervised Classification of Remotely Sensed Images Using Self-Organizing Maps for mapping land-cover changes, Proceedings of the Remote Sensing and Photogrammetry Society Annual Conference, September, Nottingham, UK, 7 p.

Halkidi, M., and Vazirgiannis, M., (2002) Clustering validity assessment using multi representatives, Proceedings of SETN Conference, April, Thessaloniki, Greece.

Haralick, R. M., Shanmugam, K., and Dinstein, I., (1973) Textural features for image classification, IEEE Transactions on Systems, Man and Cybernetics, 3(6): 610-621.

Huang, K., (2002) The use of a newly developed algorithm of divisive hierarchical clustering for remote sensing image analysis, International Journal of Remote Sensing, 23(16):3149-3168.

Ji, C. Y., (2000) Land-use classification of remotely sensed data using Kohonen self-organizing feature map neural networks, Photogrammetric Engineering and Remote Sensing, 66(12):1451-1460.

Ji, M., (2003) Using fuzzy sets to improve cluster labeling in unsupervised classification, International Journal of Remote Sensing, 24(4):657–671.

Kelly, M., Shaari, D., Guo, Q, and Liu, D., (2004) A comparison of standard and hybrid classifier methods for mapping hardwood mortality in areas affected by

"sudden oak death", Photogrammetric Engineering and Remote Sensing, 70(11):1229-1239.

Kohonen, T., (1997) Self-Organizing Maps, 2nd Ed., Berlim: Springer Verlag.

Liu, A., Bi-Cheng, L., Chen, G., and Zhangiu, X, (2005) A new ART neural networks for remote sensing image classification, Lecture Notes in Computer Science, 3611:37-42.

Magnussen, S., Boudewyn, P., and Wulder, M., (2004) Contextual classification of Landsat TM images to forest inventory cover types, International Journal of Remote Sensing, 25(12):2421-2440.

Marçal, A. R. S., and Borges, J. S., (2005) Estimation of the "natural" number of classes of a multispectral image. Proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGARSS'05), 25-29 July, Seoul, Korea, 6: 3788-3791.

Marçal, A. R. S., and Castro, L., (2005) Hierarchical clustering of multispectral images using combined spectral and spatial criteria, IEEE Geoscience and Remote Sensing Letters, 2(1):59-63.

Merkl, D., and Rauber, A., (1997) Alternative ways for cluster visualization in self-organizing maps. Proceedings of the Workshop on Self-organizing Maps (WSOM97), June, Finland, 106-111.

Özkan, C., and Erbek, F. S., (2003) A Comparison of activation functions for multispectral Landsat TM image classification, Photogrammetric Engineering and Remote Sensing, 69(11):1225-1234.

Sezgin, M., Ersoy, O. K. and Yazgan, B., (2004) Segmentation of remote sensing images using multistage unsupervised learning. Proceedings of SPIE, Applications of Digital Image Processing XXVII, August, Colorado, USA, 5558:616-623.

Shah, C. A., Arora, M. K., and Varshney, P. K., (2004) Unsupervised classification of hyperspectral data: an ICA misture model based approach, International Journal of Remote Sensing, 25(2):481-487.

Tran, T. N., Wehrens, R., and Buydens, L. M. C., (2003) Sparef: a clustering algorithm for multiespectral images, Analytica Chimica Acta, 490(1):303-312.

Tran, T. N., Wehrens, R., and Buydens, L. M. C., (2005) Clustering multispectral images: a tutorial, Chemometrics and Intelligent Laboratory Systems, 77(1-2):3-17.

Tso, B, and Olsen, R. C., (2005) Combining spectral and spatial information into hidden Markov models for unsupervised image classification, International Journal of Remote Sensing, 26(10):2113-2133.

Vesanto, J., and Alhoniemi, E., (2000) Clustering of the Self-organizing Map. IEEE Transactions on Neural Networks, 11(3):586-602.

Vilmann, T., Merenyi, E., and Hammer, B., (2003) Neural maps in remote sensing image analysis, Neural Networks, 16(3):389-403.

Wang, J., Delabie, J., Aasheim, H. C., Smeland, E., and Myklebost, O., (2002) Clustering of the SOM easily reveals distinct gene expression patterns: results of a reanalysis of lymphoma study, BMC Bioinformatics, 3:36.

Wilson, H. G., Boots, B., and Millward, A. A., (2002) A comparison of hierarchical and partitional clustering techniques for multispectral image classification, Proceedings

of IEEE International Geoscience and Remote Sensing Symposium (IGARSS'02), 24-28 June, Toronto, Canada, (3):1624-1626.

Wu, S., and Chow, T. W. S., (2004) Clustering of the Self-organizing Map using a clustering validity index based on inter-cluster and intra-cluster density, Pattern Recognition, 37:175-188.

Xu, R., and Wunsch II, D., (2005) Survey of clustering algorithms, IEEE Transactions on Neural Networks, 16(3):645-678.

17

Applications of Complex-Valued Self-Organizing Maps to Ground Penetrating Radar Imaging Systems

Akira Hirose and Yukimasa Nakano
Department of Electrical Engineering and Information Systems, The University of Tokyo
Japan

1. Introduction

We have been studying ground penetrating radars (GPRs) to visualize antipersonnel plastic landmines by dealing with complex-amplitude (amplitude and phase) data of scattered and/or reflected electromagnetic wave by using a complex-valued self-organizing map (CSOM). In GPRs, we cannot get rid of intense clutter because of the ground surface, soil clods, stones, unexploded ordnance, and exploded ones scattered as metal fragments. We need to discriminate plastic landmines from those other objects with a high correct discrimination rate. In our systems, we pay attention to complex-amplitude texture, instead of intensity used in conventional radar systems, in the spatial and frequency domains to extract textures. Then we feed the textural feature to a CSOM so that we classify the textures adaptively, and visualize the plastic landmines. We have been inspired by the human early vision and the self-organizing adaptability. In this Chapter, we present the system construction, explain the procedure and dynamics, and demonstrate the effectiveness in experiments.
The organization of this chapter[1] is as follows. In Section 2, we present the general background of plastic landmine detection. Section 3 explains the merits and demerits of three plastic landmine detection systems that utilize the electromagnetic wave. Section 4 describes the features of and merits in complex-valued neural networks including self-organizing maps. Then we present the processing flow of our plastic landmine detection system based on the CSOM and show the conventional and proposal methods in Section 5. In Section 6, we show experimental results with observed data. We summarize and conclude this chapter in Section 7.

2. Landmine detection and ground penetrating radars (GPRs)

GPRs are widely used for buried object detection, ruin investigation, groundwater assessment, and other various applications (Sato & Takeshita, 2000) (Moriyama et al., 1997). They are also expected to find nonmetallic landmines (Bourgeois & Smith, 1998)(Montoya & Smith, 1999) (Peters Jr. & Daniels, 1994) (Sato et al., 2004) (Sato et al., 2005)

[1] A part of this chapter was published in Nakano & Hirose (2009c): Y.Nakano and A.Hirose, "Adaptive ground penetrating radar systems to visualize antipersonnel plastic landmines based on local texture in scattering and reflection data in space and frequency domains," in *Radar Technology*, In-Tech, (Dec. 2009) pp.127-140, ISBN 978-953-307-029-2.

(Sato et al., 2006). A near future target is to find or visualize antipersonnel landmines with a high distinction rate between landmines and other objects. The conventional metal detectors, based on electromagnetic induction, use so low frequency that the electromagnetic field penetrates through the soil very deep, and the false negative probability is very small. However, because of its long wavelength, the resolution is limited, and they can tell just whether inductive material exists or not. They cannot distinguish landmines from other metal fragments. GPRs employ much higher frequency. Then the resulting higher resolution will be helpful to discriminate landmines.

Currently, there are two methods to remove the plastic landmines. One is a metal detector, and the other is a rotary cutter. The former detects a blasting cap made of metal in the landmine. Because the cap is very small, we must set the sensitivity at a high level. Then, the positive fault rate is as high as about 99.9% (specificity=0.1%), resulting in a lot of time to remove the landmines. The latter, rotary cutter, looks like a bulldozer, bores the ground and tries to clear the landmines by exploding them. The problems in this method are necessity of additional removal by human because of impossibility of perfect clearance, necessity of sufficient areas for the approach of the rotary cutter, and land pollution by the exploded powder.

Accordingly, though these methods have certain merits, they have also demerits. Therefore, new landmine detection systems based on GPRs attract attention and are studied by a dozen of researcher groups/laboratories presently to solve the problem. Most of the proposed methods employ high-frequency and wide-band electromagnetic wave to visualize a plastic landmine itself instead of the metallic blasting cap. In general, it is noted that the electromagnetic wave is reflected at boundaries between materials having different permittivity, and that the spatial resolution of the observation is almost the same as the wavelength. Therefore, it is possible to detect not only the metal but also the plastic body because the electromagnetic wave is reflected at the boundary of the soil and the plastic. In addition, the wide band electromagnetic wave has the possibility to observe the the accurate distance from the antenna to the target, physical property for electromagnetic wave and structural characteristics of the target. That is to say, a GPR system has a potential of detecting plastic landmines more strictly than the metal detector does.

However, high-frequency wave also induce a lot of clutter, which is caused by the roughness of the earth's surface and scattering substances other than the plastic landmines. Consequently, it is very difficult to extract significant features helpful for detecting the plastic landmines from the observed data by ignoring the clutter. Furthermore, it is also difficult to treat the extracted features effectively. A lot of processing methods were proposed so far. That is, we must resolve the following two steps to detect the plastic landmines. The first step is how to extract the features, and the second is how to treat the extracted features. To accomplish our goals, we must select or develop new effective methods.

Previously we proposed an adaptive radar imaging system to visualize plastic landmines using complex-valued self-organizing map (CSOM) (Hara & Hirose, 2004) (Hara & Hirose, 2005). The CSOM is a type of complex-valued neural networks (Hirose, 2006) (Hirose, 2009) (Hirose, 2003) (Aizenberg et al., 2000) (Nitta, 2009) (Mandic & Goh, 2009) (Hirose, n.d.) (Widrow et al., 1975), which will be described briefly in Section 4. With the system, we observe reflection and scattering to obtain a complex-amplitude two-dimensional image at multiple frequencies. In the resulting 3-dimensional (2-dimensional (space) $\times$ frequency) data, we extract local texture information as a set of feature vectors, and feed them to a CSOM for adaptive classification of the 3-dimensional texture. By using the system, we could visualize antipersonnel plastic landmines buried shallowly underground. We also constructed

a preliminary object identifier, which is a type of associative memory that learns the feature of the plastic-landmine class with adaptation ability to various ground conditions (Hirose et al., 2005) (Nakano & Hirose, 2009a). However, the system requires a long observation time because it employs mechanical scan. Long observation time is one of the most serious problems in high-resolution imaging systems. Some methods to overcome the problem have been investigated (Kobayashi et al., 2004) (Shrestha et al., 2004).
We then developed a portable visualization system with an antenna array to reduce the observation time (Masuyama & Hirose, 2007). The array has 12×12 antenna elements, resulting in about 144 pixels. The element aperture size is $28\text{mm}\times14\text{mm}$, which determines the spatial resolution. In texture evaluation and adaptive CSOM classification, a higher resolution leads to a better performance. We recently proposed a resolution enhancement method using a special antenna-selection manner in combination with elimination of direct coupling and calibration of propagation pathlength difference (Masuyama et al., 2007) (Masuyama et al., 2008). However, even with such resolution enhancement, the visualization performance is still worse than that obtained with the first mechanical-scanning system. The resolution is still insufficient, and the mutual coupling between antenna elements are not completely ignorable.
In this chapter, we describe two techniques to improve the visualization ability without mechanical scan, namely, the utilization of SOM-space topology in the CSOM adaptive classification and a feature extraction method based on local correlation in the frequency domain. In experimental results, we find that these two techniques improve the visualization performance significantly. The local-correlation method contributes also to the reduction of tuning parameters in the CSOM classification.

3. Conventional technology

Roughly speaking, there are two radar systems. One is the pulse radar, and the other is the stepped-frequency radar. In the following subsections, we briefly explain the characteristics of these methods respectively.

3.1 Pulse GPR

A pulse GPR observes the time response of the electromagnetic-wave pulse irradiated toward the ground. The time response represents the depth of a scatterer. When we sweep an antenna in two dimension horizontally, we can obtain the three dimensional data. Besides, as the electromagnetic wave is reflected at the boundary of materials that have different permittivity, it is possible to observe the reflection wave from the plastics that forms a landmine, not only the metal blasting cap. The pulse GPRs have another merit. As the pulse have a wide bandwidth, as wide as that of the stepped-frequency range mentioned below, we can observe not only the time response but also the frequency response through the Fourier transform, and these data may show characteristics of the target. Regarding the measuring time, we can conduct the measurement more speedily in comparison with the stepped-frequency GPRs.
However, as a maximum frequency component of the pulse radar is usually about $f = 6\text{GHz}$ at the most. That is, the shortest free-space wavelength is about 5cm. Then the pulse GPRs cannot observe sufficient amount of characteristics of plastic landmines whose size is typically the same as the wavelength. Besides, the ground surface is very rough, and the soil, including various scatterers, often causes serious clutter. Therefore, it is very difficult to obtain clear images. To solve this problem we need to utilize sharper pulse which consists of a wide-band

wave with a high power. However, as a high-peak pulse is distorted by the nonlinearity in transmitter circuits and switched antennas, the problem cannot be solved sufficiently.

3.2 Stepped-frequency GPR

Stepped-frequency GPRs observe the reflected continuous wave at a wide-range frequency points. This method does not need to output strong power instantaneously. Then, the electromagnetic wave has little influence on the nonlinearity of the circuits and the switches. As the results, stepped-frequency GPR accomplishes higher SN ratio than the pulse GPR. In addition, it is easier for the stepped-frequency radar to observe the high frequency wave and select bandwidth freely than a pulse radar system. Besides, we can obtain the time response, like what pulse radar provide, through the inverse Fourier transform of the observed frequency domain data.

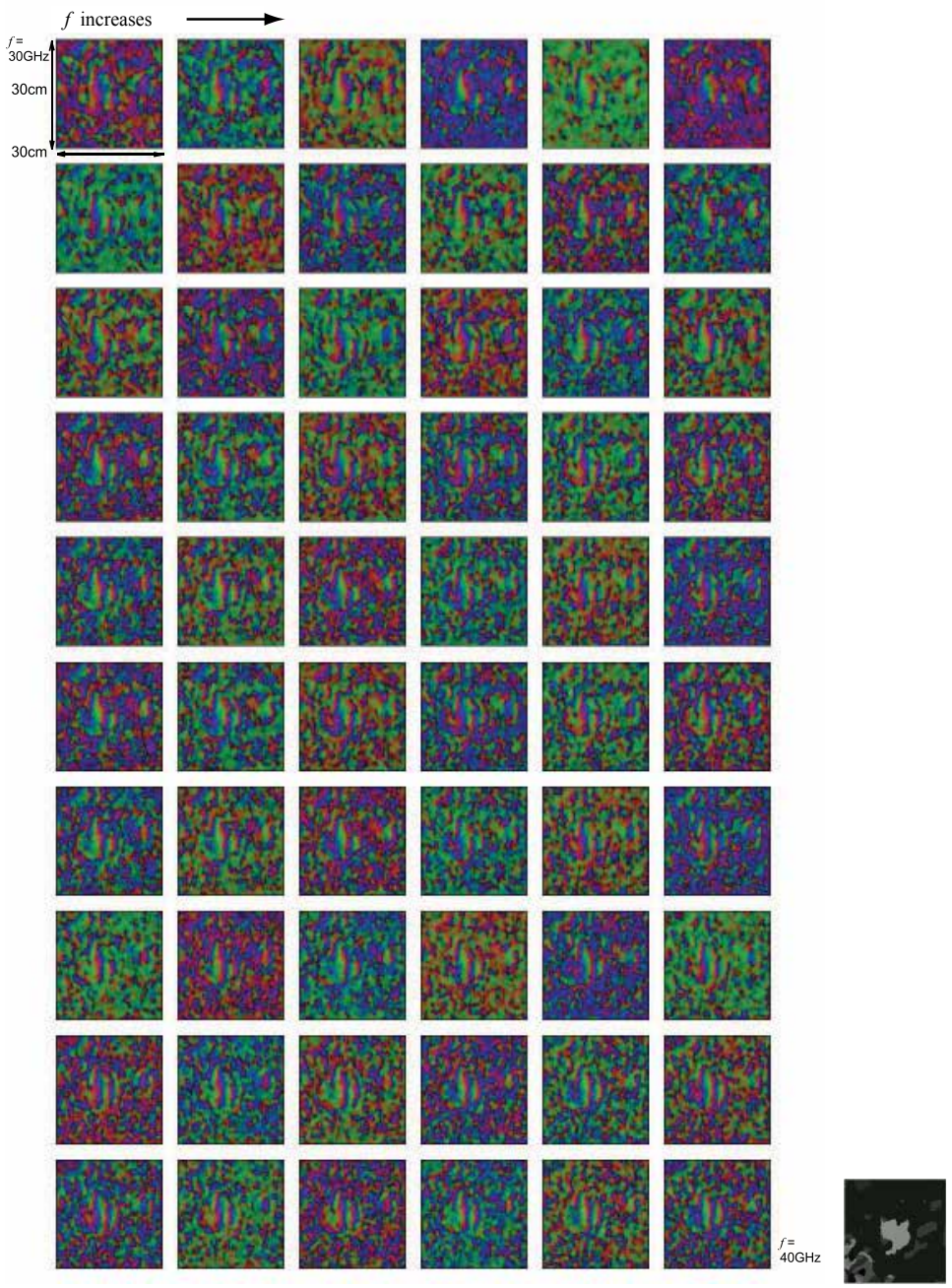
Here we present an example of raw data obtained by a stepped-frequency radar system having a relatively high frequency range with mechanical scanning in Fig.1(a) so that the readers can catch what type of data we are dealing with. This is a rather simple example for observation of 30cm$\times$30cm area of sand (rich in silica) soil. The frequency is changed stepwise from the top-left to bottom-right. The brightness shows the intensity, while the hue represents the phase. Though the complex-amplitude texture (pattern) seems almost at random, the randomness is varying place by place. In this experiment, we have a plastic landmine (TYPE-72 amtipersonnel plastic landmine) at the center of the observation area in this case. Readers may find a certain feature in the randomness at the central area. In addition, we can actually find similar changes there also in the frequency domain. The human beings are highly sensitive to such changes in the textural features. We extract the features specific to the landmine area based on the human early vision system in the complex-amplitude domain by using the CSOM to visualize the landmine area as shown at the bright area at the center in Fig. 1(b).

As above, stepped-frequency GPRs enables us to accomplish high SN ratio at the high frequency with a wide band, and obtain not only the time response but also frequency domain feature very effectively. It is true that even the stepped-frequency GPR has a drawback. Namely, it takes too long time to measure the scattering because of the time required for frequency sweeping. However, this problem will be solved by inventing new appropriate devices in the near future.

Then we can expect a higher precision with the system utilizing the stepped-frequency GPR than the conventional systems. To achieve this purpose, there are two important points we should consider carefully. One is to extract useful features from the obtained data, and the other is to fully utilize the features. However, a perfect technique has not been suggested yet. In the next section, we show the details of our CSOM-based signal processing published in our previous paper (Nakano & Hirose, 2009b).

4. Complex-valued neural networks including self-organizing maps

Complex-valued neural networks are the networks that deal with complex signals with complex variables in the complex domain (Hirose, 2006) (Aizenberg et al., 2000) (Hirose, 2003) (Nitta, 2009) (Mandic & Goh, 2009). They are also extensions of various complex-valued linear signal processing methods such as complex least mean square (LMS) algorithm (Widrow et al., 1975). The behavior of a N-dimensional complex-valued neural networks is different from that of $2N$-dimensional real-valued one (Hirose, 2009). The origin lies in the nature of multiplication in the complex domain. That is, the weight multiplication at synapses



(a) (b)

Fig. 1. (a) raw image showing three dimensional (two dimentional in space × one dimensional in frequency) complex-amplitude data of scattering / reflection and (b)a result of adaptive segmentation using the CSOM. In (a), the brightness represents the intensity, whereas the hue the phase, and in (b) the grayscale indicates textural class labels (Hara & Hirose, 2004).

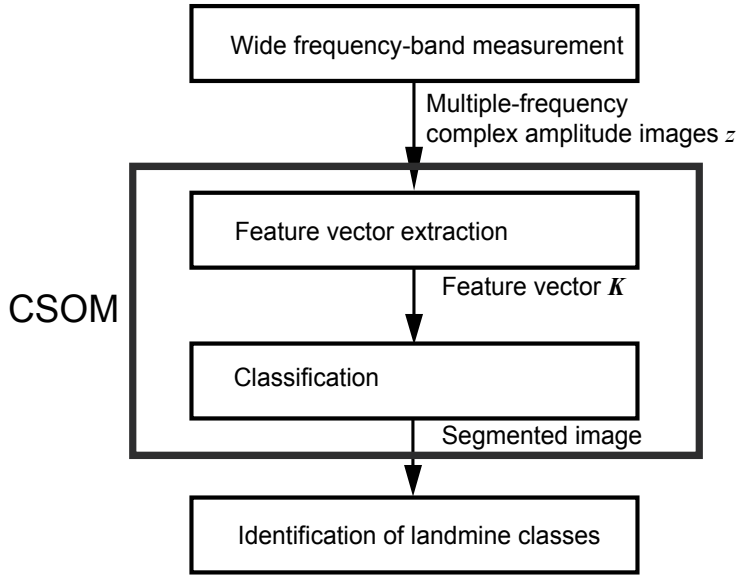


Fig. 2. The overall processing flow (Nakano & Hirose, 2009b).

yields the phase rotation as well as the amplitude amplification or attenuation. This type of multiplication reduces ineffective degree of freedom in the learning or self-organization to enhance the generalization characteristics in comparison with double-dimensional real-number networks, in spite of the fact that a complex number can be represented as an ordered pair of real numbers. The network dynamics consisting of this elemental rotation and amplification / attenuation leads to significant merits in total, originating from the consistency with the wave-related phenomena and information in the GPRs and other coherent systems (Kawata & Hirose, 2008) (Tay et al., 2008). For details, see (Hirose, 2006) (Hirose, n.d.).
Among various neurodynamics in the complex domain, the complex-valued self-organizing maps (CSOMs) may possess less features which reflect the complex multiplication, since SOM in general have two sub-processes in the operation, i.e., winner determination and weight update, both of which sub-processes mostly consist of addition and subtraction in its arithmetics. However, even in such SOM cases, the CSOMs are sensitive to coherence, which is a great advantage in GPRs and other systems that uses coherent waves. In addition, when we employ a complex inner product as the metric to determine the winner, it is found that the advantage is greatly enhanced (Aoyagi et al., 2010).

5. System construction

5.1 Overall construction

Figure 2 shows the processing flow in our plastic landmine visualization system. We describe the components briefly. First, we operate our system at a stepped-frequency mode to obtain wideband electromagnetic reflection and/or scattering images at multiple frequency points. The details of the system is given in Ref.(Masuyama & Hirose, 2007). We acquire complex amplitude images at 10 observation frequency points from 8 to 11.6GHz at an interval of 0.4GHz.
Next, we generate a spatially segmented image by using a CSOM that classifies local texture adaptively. The classification consists of two steps. In the first step, we extract feature vectors

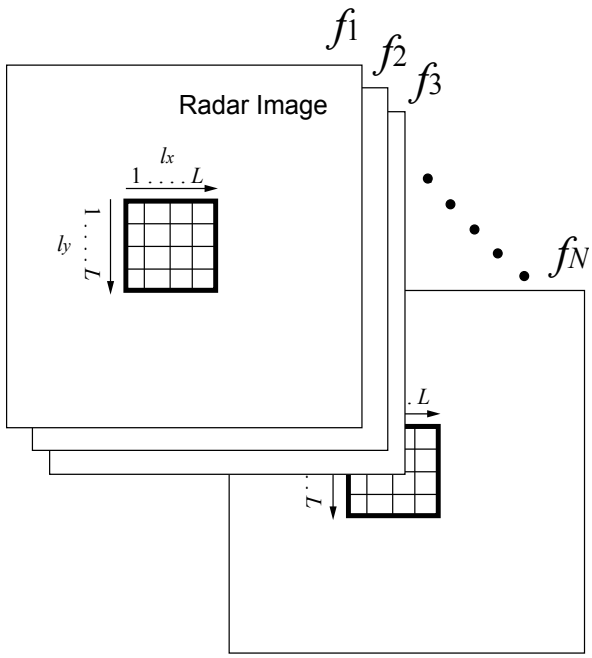


Fig. 3. The scanning window for local textural feature extraction (Nakano & Hirose, 2009b).

representing local complex-amplitude textural quantity in a local window that sweeps all over the image. As shown in Fig.3, we prepare a sweeping window in each frequency image at a synchronizing real-space location. We calculate correlations between pixel values in the window in terms of real-space relative distance and frequency-domain distance. We assume that the correlation values represent the texture at around the pixel at the window center, and we put the values at the center pixel as the textural feature. In the second step, we classify the extracted feature vectors adaptively by using a CSOM (Hara & Hirose, 2004). Then we color pixels correspondingly with the resulting classes to generate a segmented spatial image.
Lastly, we identify landmine classes included in the segmented image. We use an associative memory that learns respective features of landmine classes and other objects such as metal fragments, stones and clods. We prepare a set of teacher features for the learning beforehand by observing land pieces where we know the landmine locations.
In this paper, we propose two novel methods in the CSOM processing. One is a dynamics in the feature vector classification, and the other is a feature vector extraction method. The former is utilization of SOM-space topology in the CSOM adaptive classification by introducing a ring CSOM, and the latter is the extraction of local correlation in the frequency domain.

5.2 Local correlation method to extract textural features

One of the specific methods in our system is to adopt frequency-domain local correlation in the texture feature extraction. We extract textural features in the spatial and frequency domains as follows. This process corresponds to the human early vision using Gabor filter though we deal with the correlation values directly. In particular, we extend the "local correlation" to the frequency domain in the following manner (Nakano & Hirose, 2009b).

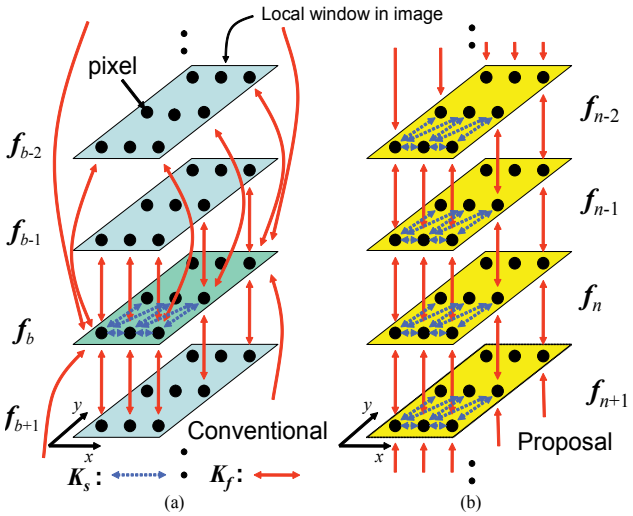


Fig. 4. (a)Conventional and (b)proposed textural feature extraction methods based on correlation in real-space and frequency domains (Nakano & Hirose, 2009b).

Figure 4(a) illustrates the conventional method, in which the feature vector $\boldsymbol{K}$ is calculated for complex pixel values $z(l_x, l_y, f)$, where the amplitude is represented in decibel (dB), as

$$\boldsymbol{K} = [M, \boldsymbol{K}_s, \boldsymbol{K}_f] \tag{1}$$

$$M = \frac{1}{L^2} \sum_{l_x=1}^{L} \sum_{l_y=1}^{L} z(l_x, l_y, f_b) \tag{2}$$

$$\boldsymbol{K}_s = [K_s(0,0), K_s(1,0), K_s(0,1), K_s(1,1)] \tag{3}$$

$$K_s(i,j) = \frac{1}{L^2} \sum_{l_x=1}^{L} \sum_{l_y=1}^{L} z(l_x, l_y, f_b) z^*(l_x + i, l_y + j, f_b) \tag{4}$$

$$\boldsymbol{K}_f = [K_f(f_1), \cdots, K_f(f_N)] \tag{5}$$

$$K_f(f_n) = \frac{1}{L^2} \sum_{l_x=1}^{L} \sum_{l_y=1}^{L} z(l_x, l_y, f_b) z^*(l_x, l_y, f_n) \tag{6}$$

where M, $\boldsymbol{K}_s$, and $\boldsymbol{K}_f$ are the mean, real-space-domain correlations, and frequency-domain correlations, respectively. Real-space discrete coordinate l_x and l_y determine pixel positions in the local window as shown in Fig.3.

We prepare a local observation window of $L \times L$ pixels to extract the local textural feature in the window by calculating correlations between pixel values. In (2)–(6), M is the average of pixel values in the window at a base frequency f_b, which we select among the 10-frequency observation points in advance. The vector $\boldsymbol{K}_s$ in (3) is the local correlations in the $L \times L$ real-space window at f_b, while $\boldsymbol{K}_f$ is the correlations between pixel values at f_b and other frequencies f_n at identical positions.

The effectiveness of $K_f(f_n)$ in (6) as a feature vector was suggested by the following frequency-dependent interference. A plastic landmine usually has parallelism among its

ceiling, bottom, and air gap inside, if any, which causes interference, whose spectral profile is periodic in the frequency domain. That is to say, we will observe a resonance at integral multiple of a certain frequency periodically in the frequency domain, resulting in a specific peak at certain f_n in $K_f(f_n)$. We intended to capture this phenomenon in (6). However, we found in our series of experiments that we normally observe only a single peak in the 8–12GHz band. If we expect multiple frequency peaks, we have to expand the observation bandwidth. However, very high-frequency electromagnetic wave cannot penetrate ground so deep.
Then we have changed our direction. Note that, in the spatial texture case described above, we paid attention to local correlation caused by the Markovian property. In the same way, also in the frequency domain, we decided to calculate the local correlation to observe the frequency space texture in a simple manner.
Figure 4(b) illustrates our proposal, namely the frequency-domain local correlation method, to extract the frequency-domain feature. We define our new K_f as

$$\boldsymbol{K}_f = [K_f(f_1), \cdots, K_f(f_{N-1})] \tag{7}$$

$$K_f(f_n) = \frac{1}{L^2} \sum_{l_x=1}^{L} \sum_{l_y=1}^{L} z(l_x, l_y, f_n) z^*(l_x, l_y, f_{n+1}) \tag{8}$$

where $\boldsymbol{K}_f$ is the feature vector representing the correlation coefficients between the data at adjoining frequency points. This method enables us to eliminate the base frequency f_b, which means that we do not need to choose a special frequency. To extend this f_b-free advantage further, we also modify M and $K_s(i,j)$ slightly as

$$M = \frac{1}{L^2 N} \sum_{l_x=1}^{L} \sum_{l_y=1}^{L} \sum_{n=1}^{N} z(l_x, l_y, f_n) \tag{9}$$

$$K_s(i,j) = \frac{1}{L^2 N} \sum_{l_x=1}^{L} \sum_{l_y=1}^{L} \sum_{n=1}^{N} z(l_x, l_y, f_n) z^*(l_x + i, l_y + j, f_n) \tag{10}$$

That is, M and $\boldsymbol{K}_s$ are averaged for the all used frequency data.
The frequency-domain local correlation method is suitable for the processing in this system. Instead of the radar cross section, we use the texture of complex amplitude when we distinguish plastic landmines and other objects such as metal fragments and stones. If we can obtain a sufficiently high resolution in real-space and frequency domains, we should take into account the correlation between one pixel and another at a certain distance. However, when the system has only a low resolution, it is more effective to concentrate on the correlation between neighboring pixels, in which we can expect useful correlation information.
Additionally, in the proposed method, it is a great merit that we do not need the base frequency f_b, which was required in the conventional method. Previously we have a number of possible f_b. As presented below, it is a problem that a different f_b results in a different segmentation image. The new method is free from this problem because we have only one way to construct $\boldsymbol{K}$.

5.3 Utilization of SOM-space topology in the CSOM adaptive classification

The other method specific to our new system is that we employ a SOM dynamics that utilizes SOM-space topology in the CSOM adaptive classification (Nakano & Hirose, 2009b). Figure 5 shows the CSOM structure, which forms a ring in the CSOM space. In the CSOM in our

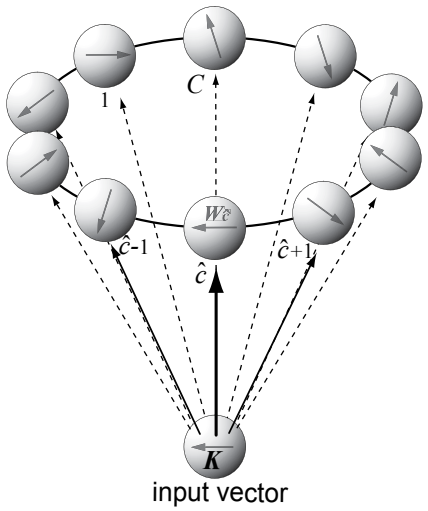


Fig. 5. The ring-CSOM structure. Balls represent reference vectors in the CSOM, and the directions of the arrows show the vector values, among which the winner $\boldsymbol{W}_{\hat{c}}$ and the two neighbors $\boldsymbol{W}_{\hat{c}\pm1}$ change in the self-organization.

previous system, we classified the feature vectors by using the K-mean clustering algorithm, which is the simplest SOM dynamics, as (Hara & Hirose, 2004)

$$\boldsymbol{W}_{\hat{c}}(t+1) = \boldsymbol{W}_{\hat{c}}(t) + \alpha(t)(\boldsymbol{K} - \boldsymbol{W}_{\hat{c}}(t)) \quad (11)$$

$$\alpha(t) = \alpha(0)\left(1 - \frac{t}{T}\right) \quad (12)$$

$\boldsymbol{W}_{\hat{c}}(t)$	: reference vector of the winner
$\boldsymbol{K}$	: input feature vector
t	: iteration number in self-organization
T	: maximum iteration number
$\alpha(t)$	: self-organization coefficient

where the winner $\boldsymbol{W}_{\hat{c}}(t)$ is the reference vector nearest to $\boldsymbol{K}$ among all the reference vectors $\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_c, \ldots, \boldsymbol{W}_C$. We update $\boldsymbol{W}_{\hat{c}}$ iteratively by presenting $\boldsymbol{K}$ sequentially. In the new system, we also introduce the self-organization of reference vectors at the winner neighbors $(c \pm 1)$ in the SOM space shown in Fig.5 as (Nakano & Hirose, 2009b)

$$\boldsymbol{W}_{\hat{c}\pm1}(t+1) = \boldsymbol{W}_{\hat{c}\pm1}(t) + \beta(t)(\boldsymbol{K} - \boldsymbol{W}_{\hat{c}\pm1}(t)) \quad (13)$$

$$\beta(t) = \beta(0)\left(1 - \frac{t}{T}\right) \quad (14)$$

where $\beta(t)$ is another self-organization coefficient for the neighboring classes, which is usually smaller than $\alpha(t)$. The classes $\hat{c} \pm 1$ are neighbors of the winner class $\hat{c}$ in the CSOM space. The CSOM space is used only to determine the winner neighbors, whereas the winner is determined in the $\boldsymbol{K}$ space.

The reason of the modification is as follows. In the previous method, we used the K-means algorithm (11), which is the simplest dynamics in the SOM. Because we have only about 10

Fig. 6. Photo of our field experiment in Siem Reap, Cambodia (Hirose, 2008).

classes in the adaptive vector quantization in the landmine visualization system, we paid less attention to the SOM-space topology. Nevertheless, we could obtain sufficiently meaningful classification results (Hara & Hirose, 2004). However, with the present portable visualization system with a lower spatial resolution at the antenna array (Masuyama & Hirose, 2007), the failure probability in the classification became slightly higher than the first laboratory system. We sometimes fail to extract texture features sufficiently because of the decrease in resolution. As described later, in such an insufficient extraction case, we found that only a small number of classes were used in the vector quantization.
We therefore proposed the utilization of the SOM-space topology in the CSOM adaptive classification, to activate most of the prepared reference vectors, by introducing additional self-organization at neighbor class vectors. In this paper, we change the values of only the two adjacent-class vectors $\boldsymbol{W}_{c\pm1}$ as shown in (13). The neighbor vector number is small because the number of the totally prepared classes is small, i.e., only about 10. The structure of the CSOM should also be suitable for the small size, namely, one-dimensional as shown in Fig.5.

6. Experiments and results

Figure 6 shows the field experiment conducted in Cambodia to evaluate our system for laterite soil which has a high permeability and hence a larger clutter. At the site, we also have frequent heavy rainfall, resulting in temporary higher permittivity of the soil lasting for several hours to a few days. Table 1 shows the parameters used in the following experiments. We have determined the values of $\alpha(0)$ and $\beta(0)$ empirically. We bury a mock plastic landmine under ground iteratively. We change the burial situation every time, including the ground surface and underground. The surface-roughness amplitude is about 2cm peak-to-peak. In any case, the landmine is buried at around the center of the observation area.
Figure 7(a) shows an experimental result (Result 1) obtained for lightly wet laterite soil. The numbers show the observation frequencies. The upper blue maps show the amplitude data, while the lower color maps show the phase data. Scales of amplitude and phase are shown at the top. The position in every map corresponds to the position in real space. As mentioned

Target (Plastic landmine)	
Size	78mm$^{\phi}$,40mm high
Burial depth	$2 \sim 3$cm
System	
Antenna height	$2 \sim 3$ cm above ground
Window size	$L = 4$
Frequency number	$N = 10$
Class number	$C = 10$
Initial learning coefficients	$\alpha(0) = 0.4$
	$\beta(0) = 0.1$
Maximum learning iteration	$T = 10$

Table 1. Parameters of target and system (Nakano & Hirose, 2009b).

above, we use these complex amplitude data obtained at the 10 frequency points. Figure 7 (b) shows segmented images generated with the previous method. The numbers are base frequencies f_b used respectively. We can choose feature vectors $\boldsymbol{K}$ in 10 ways because there are 10 possible f_b. Each gray level indicates one of the 10 classes. We can find a segmented area at the buried plastic landmine position at f_b=8GHz and 8.8GHz. However, we cannot at other f_b. Figure 7 (c) shows the result of segmentation by utilizing the SOM-space topology in the CSOM adaptive classification. We find that there are more classes used in the classification, i.e., 10 classes in most cases, than that in the case of the previous method. We can confirm that we can classify the landmine area appropriately at most f_b. For example, also at 8.4GHz and 9.2GHz, we are successful in the segmentation this time. These results reveal that we can improve the performance of classification by the utilization of the SOM-space topology in the CSOM. Figure 7 (d) shows the segmentation result obtained by the frequency-domain local correlation method as well as the utilization of SOM-space topology. As mentioned before, there is only one manner to extract feature vectors $\boldsymbol{K}$ in this proposed method because we have no f_b. Here we show four result examples for various initial reference vectors in the CSOM since the result of the CSOM may depend on the initial reference-vector values. In all the cases, the landmine area is segmented correctly. We confirm a high robustness of the present method with the two proposal.

Figure 8(a) shows a measurement result (Result 2) obtained for highly wet laterite soil just after a heavy rain, which means a more difficult situation than that of Fig.7(a). The landmine classification seems more difficult in this case than that of Fig.7 because the calibration of direct coupling components (Masuyama et al., 2007) is somewhat sensitive to noise, occasionally resulting in insufficient compensation of antenna-selection-mode dependent amplitude. Figure 8(b) shows the segmented images obtained with the previous method. We can classify the landmine area only when f_b=9.2GHz. We completely failed in the segmentation at other f_b. Figure 8(c) shows a result by utilizing the SOM-space topology in the CSOM. We can segment the landmine area only at 9.2GHz again. Figure 8 (d) shows the results obtained by employing the two proposed methods. It is confirmed that we can classify the landmine area perfectly. We show four results for various initialization again. The results indicate that we can segment landmine areas stably. In addition, we recognize that more classes are used for the classification in Fig.8(d) than in Fig.8(c) despite we use the same dynamics for the classification in the CSOM. For this reason, we can extract more characteristic feature quantities with the frequency-domain local correlation method than that with the previous one.

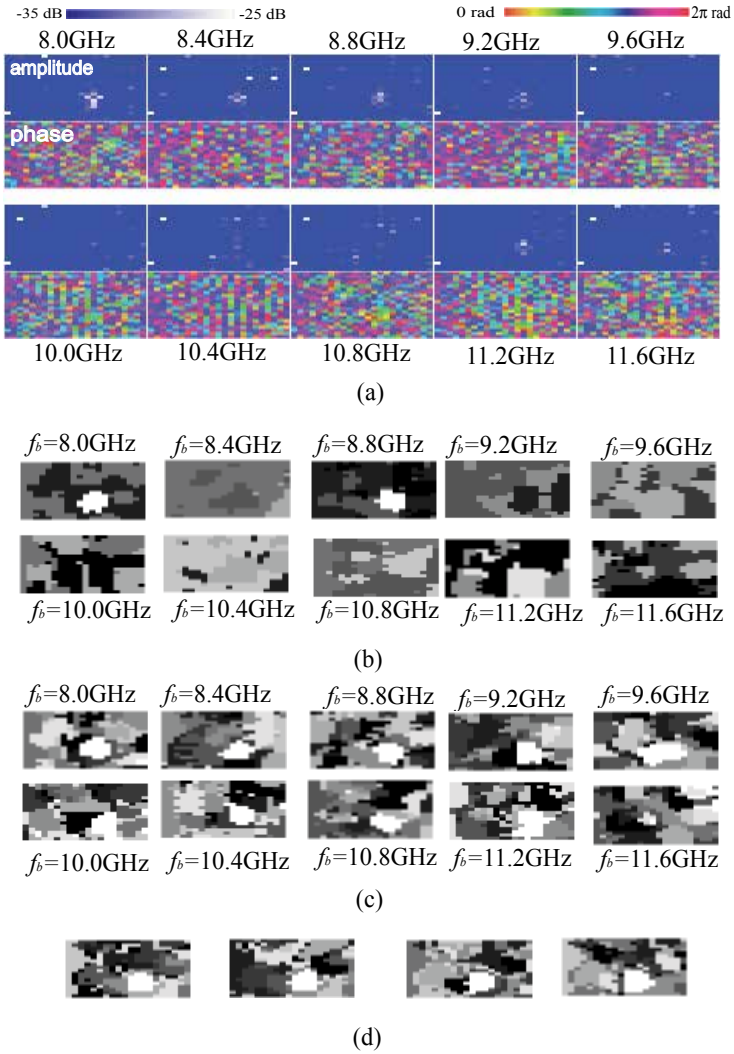


Fig. 7. Experimental results 1 for lightly wet laterite soil. (a)Amplitude and phase images at 10 frequency points, (b)classification results with the previous method. Numbers denote base frequency f_b, (c)classification results with utilization of SOM-space topology in the CSOM. Numbers denote base frequency. (d)Classification results with utilization of SOM-space topology in the CSOM and frequency-domain local correlation method. The four images show the results with various initial reference vectors in the CSOM (Nakano & Hirose, 2009b).

7. Summary

We reviewed recently proposed two techniques, used in stepped-frequency GPRs, to improve the performance of the GPR system to visualize plastic landmines. One is to employ local correlation as the feature vector components in the frequency domain as well as spatial domain. It extracts complex texture information better and, at the same time, eliminates the need of the base frequency, which had to be chosen by the user as one of the tuning parameters

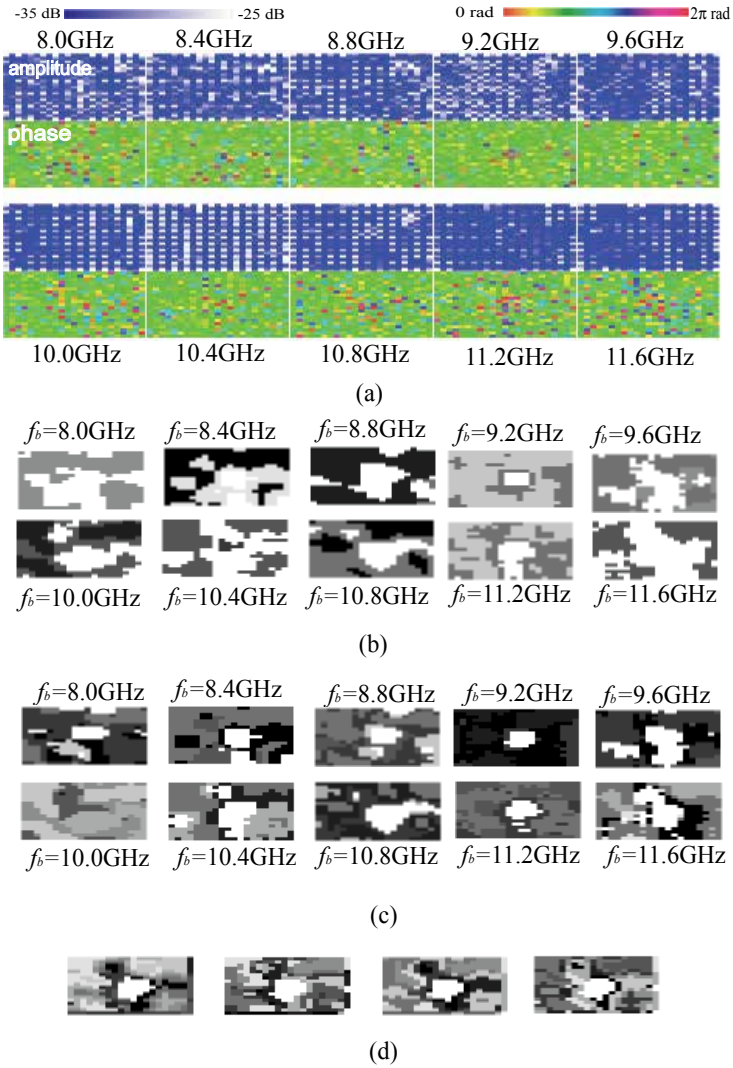


Fig. 8. Experimental results 2 for heavily wet laterite soil. (a)Amplitude and phase images at 10 frequency points, (b)classification results with the previous method. Numbers denote base frequency f_b, (c)classification results with utilization of SOM-space topology in the CSOM. Numbers denote base frequency. (d)Classification results with utilization of SOM-space topology in the CSOM and frequency-domain local correlation method. The four images show the results with various initial reference vectors in the CSOM (Nakano & Hirose, 2009b).

in the previous system. The other technique is to utilize SOM-space topology in the CSOM adaptive classification to stabilize the classification process. Unlike the K-means algorithm, we can use most of the prepared classes in the learning vector quantization. We presented the experimental results obtained in Cambodia to demonstrate better and stable visualization of the plastic landmine.

8. References

Aizenberg, I. N., Aizenberg, N. N. & Joos, V. (2000). *Multi-Valued and Universal Binary Neurons – Theory, Learning and Applications –*, Kluwer Academic Publishers, Boston / Dordrecht / London.

Aoyagi, T., Radenamad, D., Nakano, Y. & Hirose, A. (2010). Complex-valued self-organizing map clustering using complex inner product in active mmillimeter-wave imaging, *Proceedings of the International Joint Conference on Neural Networks (IJCNN) 2010 Barcelona*, IEEE/INNS, Barcelona, pp. 1346–1351.

Bourgeois, J. M. & Smith, G. S. (1998). A complete electromagnetic simulation of the separated-aperture sensor for detecting buried land mines, *IEEE Transactions on Antennas and Propagation* 46(10): 1419–1426.

Hara, T. & Hirose, A. (2004). Plastic mine detecting system using complex-valued self-organizing map that deals with multiple-frequency interferometric images, *Neural Networks* 17(8-9): 1201–1210.

Hara, T. & Hirose, A. (2005). Adaptive plastic-landmine visualizing radar system: effects of aperture synthesis and feature-vector dimension reduction, *IEICE Transactions on Electronics* E88-C(12): 2282–2288.

Hirose, A. (2006). *Complex-Valued Neural Networks*, Springer-Verlag, Heidelberg, Berline, New York.

Hirose, A. (2008). An adaptive ground penetrating radar imaging system based on complex-valued self-organizing map – recent progress and experiments in cambodia –, *Proceedings of the World Congress on Computational Intelligence (WCCI) 2008 Hong Kong – International Joint Conference on Neural Networks (IJCNN) 2008 Hong Kong –*, pp. 1610–1615.

Hirose, A. (2009). Complex-valued neural networks: The merits and their origins, *Proceedings of the International Joint Conference on Neural Networks (IJCNN) 2009 Atlanta*, IEEE – INNS, pp. 1237–1244.

Hirose, A. (n.d.). Complex-valued neural networks (Tutorial, International Joint Conference on Neural Networks (IJCNN) 2009 Atlanta) (IEEE Computational Intelligence Society (CIS) video archive.
URL: *http://ewh.ieee.org/cmte/cis/mtsc/ieeecis/video_tutorials.htm*

Hirose, A. (ed.) (2003). *Complex-Valued Neural Networks: Theories and Applications*, World Scientific Publishing Co. Pte. Ltd.

Hirose, A., Toh Jiayun, A. & Hara, T. (2005). Plastic landmine identification by multistage association, *IEICE Tech. Rep.* (NC2004-156).

Kawata, S. & Hirose, A. (2008). Frequency-multiplexing ability of complex-valued Hebbian learning in logic gates, *International Journal of Neural Systems* 12(1): 43–51.

Kobayashi, T., Feng, X. & Sato, M. (2004). A gpr for landmine detection using an array antenna, *International Symposium on Antennas and Propagation (ISAP) Sendai*.

Mandic, D. P. & Goh, V. S. L. (2009). *Complex Valued Nonlinear Adaptive Filters – Noncircularity, Widely Linear and Neural Models*, Wiley.

Masuyama, S. & Hirose, A. (2007). Walled LTSA array for rapid, high spatial resolution, and phase sensitive imaging to visualize plastic landmines, *IEEE Transactions on Geoscience and Remote Sensing* 45(8): 2536–2543.

Masuyama, S., Yasuda, K. & Hirose, A. (2007). Removal of direct coupling in a walled-LTSA array for visualizing plastic landmines, *International Symposium on Antennas and Propagation (ISAP) 2007 Niigata*, pp. 1238–1241.

Masuyama, S., Yasuda, K. & Hirose, A. (2008). Multiple mode selection of walled-ltsa array elements for high resolution imaging to visualize antipersonnel plastic landmines, *IEEE Geoscience and Remote Sensing Letters* 5(4): 745–749.

Montoya, T. P. & Smith, G. S. (1999). Land mine detection using a ground-penetrating radar based on resistively loaded vee dipoles, *IEEE Trans. Antenna and Propagation* 47(12): 1795–1806.

Moriyama, T., Nakamura, M., Yamaguchi, Y. & Yamada, H. (1997). Radar polarimetry applied to the classification of target buried in the underground, *Wideband Interferometric Sensing and Imaging Polarimetry*, Vol. 3210 of *Proc. of SPIE*, pp. 182–189.

Nakano, Y. & Hirose, A. (2009a). Estimation of landmine class likelihood by evaluating the degree of matching of features in ring-CSOM space, *Technical report*, IEICE Technical Report (NC2008-169).

Nakano, Y. & Hirose, A. (2009b). Improvement of plastic landmine visualization performance by use of ring-CSOM and frequency-domain local correlation, *IEICE Transactions on Electronics* E92-C(1): 102–108.

Nakano, Y. & Hirose, A. (2009c). *Radar Technology*, In-Tech, chapter Adaptive ground penetrating radar systems to visualize antipersonnel plastic landmines based on local texture in scattering and reflection data in space and frequency domains, pp. 127–140.

Nitta, T. (ed.) (2009). *Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters*, IGI-Global.

Peters Jr., L. & Daniels, J. J. (1994). Ground penetrationg radar as a surface environmental sensing tool, *Proceedings of the IEEE* 82,No.12: 1802–1822.

Sato, M., Fujiwara, J., Fenga, X. & Kobayashi, T. (2005). Dual sensor ALIS evaluation test in Afghanistan, *IEEE Geoscience and Remote Sensing Society Newsletter* pp. 22–27.

Sato, M., Hamada, Y., Feng, X., Kong, F.-N., Zeng, Z. & Fang, G. (2004). GPR using an array antenna for landmine detection, *Near Subsurface Geophysics* 2: 7–13.

Sato, M., Takahashi, K., Feng, X. & Kobayashi, T. (2006). Stepped-frequency handheld demining dual sensor alis, *Proceeding of 11th International Conference on Ground Penetrating Radar*, p. UXO.10.

Sato, M. & Takeshita, M. (2000). Estimation of subsurface fracture roughness by polarimetric borehole radar, *IEICE Transactions on Electronics* E83-C(12): 1881–1888.

Shrestha, S. M., Arai, I. & Tomizawa, Y. (2004). Landmine detection using impulse ground penetrating radar, *Internationall Symposium on Antennas and Propagation (ISAP) Sendai*.

Tay, C. S., Tanizawa, K. & Hirose, A. (2008). Error reduction in holographic movies using a hybrid learning method in coherent neural networks, *Applied Optics* 47(28): 5221–5228.

Widrow, B., McCool, J. & Ball, M. (1975). The complex lms algorithm, *Proceedings of the IEEE* 63: 719–720.

18

Automated Mapping of Hydrographic Systems from Satellite Imagery Using Self-Organizing Maps and Principal Curves

Marek B. Zaremba
Université du Québec en Outaouais
Canada

1. Introduction

A fully automatic and high-precision cartographic mapping of terrain features such as forests, rivers or roads from multispectral satellite or aerial images is a challenging problem in remote sensing, largely due to the fact that it requires an adequate representation of irregular and discontinuous objects. Being able to provide sub-meter resolution multispectral images, high-resolution satellites such as QuickBird or Ikonos broaden the application possibilities of satellite imagery and offer the possibility of making them sensors of choice for a variety of environmental applications. Their high spatial and radiometric resolution facilitates visual interpretation. Temporal resolution of image databases can be largely increased, due to the known revisit time and pointing capabilities of the satellite platform, which facilitates large-scale change-detection and monitoring of selected areas in order to keep natural environment databases up-to-date. Certain analyses involving spectral change detection and dynamically obtained maps can be performed more easily and in a more automated fashion. Finally, high-resolution satellite images offer the potential for being ever more competitive in terms of price with the aerial images.

This chapter presents methods based on Self-Organizing Maps (Kohonen et al., 1996) developed in efforts to fully automate the generation of hydrographic maps from remotely sensed imagery. The complexities of generating cartographic representations of hydrological objects, such as rivers and lakes, from satellite and aerial images consists generally of two categories of tasks: the first involves the extraction process of the required linear or a real feature while the second involves generation of a suitable representation in a form appropriate for cartographic map presentation. The presented approach applies the technology of Self-Organizing Maps (SOM) at both stages of the hydrological mapping process, i.e., the detection of water bodies from multispectral images and the subsequent tracing of hydrological systems or networks.

The first task can be approached by applying a classification technique or through a scene analysis method. A number of different methods have been reported in the literature. Conventional image processing techniques typically apply edge detection algorithms (Ma & Manjunath, 2000) in efforts to define water regions. A similar problem of road detection from satellite images was discussed in (Auclair Fortier et al., 2001). A rule-based approach to segmentation of satellite images was presented in (Ton et al., 1991). Selection of

multispectral bands for segmentation of low-resolution ocean images was discussed in (Ainsworth & Jones, 1999). Availability of three-dimensional information can enhance the mapping process. In addition to providing the raw elevation data, the Digital Elevation Models (DEMs) have been processed (Jenson & Domingue, 1988) to extract hydrographic features, such as basins or flowlines. The conventional methods generally used are sensitive to noise and usually provide only a rough shape description or present extractions that are discontinuous suggesting erroneous heterogeneity of objects. The extraction process described in this chapter makes use of the topological relationships between or among the objects - automatically extracted by SOM.

The second task requires derivation of objects suitable for representation in map form. The conventional approaches resort to classical image processing techniques, such as skeletal thinning algorithms or splines. Extraction of rivers from digital SPOT images using the approximation of the river shape by the snake method was reported in (Dillabaugh et al., 2002). A two stage, multi-resolution procedure was applied to produce a detailed outline of the channel banks and, subsequently, fit high resolution imagery to the channel banks using the dynamic contour technique. The calculation of the snake shapes is based on dynamic programming and is computationally expensive. Another disadvantage of the snake-based method is that the resulting shape can often be located outside of the area of the river. A powerful technique for obtaining skeletons of planar objects turned out to be neural network architectures. A self-organizing map was initially investigated for skeletal shape extraction in (Datta et al., 1996). An initial linear topology was able to evolve to circular and - depending on the angle between the neuron and its neighbors - forked forms. However, the skeletonization results were not invariant to the rotations of the image because of the flow-through version of SOM adopted in that work. By using the batch formulation (Ritter et al., 1996) of the SOM algorithm, this problem was solved in (Sing et al, 2000).

Many of the conventional approaches, such as skeletal thinning algorithms or splines, will fail. This is as a result of the quality of the derived objects from the input images which generally contain noise or objects that are rendered discontinuous as a result of object scarcity. One of the objectives of this study was to develop techniques able to deal with images in the form of sparse data sets. Sparse data sets refer to spatial discontinuity of a homogeneous and topologically related object within an object class. There are known several methods to cope with the sparse shape and occlusions, the most popular of them (by the compromise of efficiency and simplicity) are based on a median filtering and morphological set operations such as opening and closing (Haralick & Shapiro, 1992). Afterwards, a standard thinning algorithm is applied to extract the shape skeleton. Such methods usually introduce a large amount of distortions when the shape sparseness and occlusion is significant.

In order to effectively deal with sparse and discontinuous remote sensing data at the stage of tracing hydrographic systems, the principal curve approach was adopted. In this approach, the skeletal description of point sets is extracted directly, without a sequential thinning procedure. Principal curves are one of the nonlinear generalizations of principal components. They were first defined in (Hastie & Stuetzle, 1989) as self-consistent smooth curves defined by the property that each point of the curve is the average of all data points that project to it, i.e., for which that point is the closest point on the curve. Consequently, the curve passes through the "middle" of a multi-dimensional probability distribution or data cloud. It is shown in (Duchamp & Stuetzle, 1996) that Hastie & Stuetzle principal curves are saddle points of the distance function (MSE) which explains why cross-validation is not a

viable method for choosing the complexity of principal curve estimates. The definition of principal curves in (Kégl et al, 2000) assures that, assuming the distribution has finite second moments, principal curves always exist. This makes it possible to carry out a theoretical analysis of learning principal curves from training data. Based on their definition, the authors developed the Polygonal Line Algorithm. The algorithm starts with a straight line segment and, in each iteration, increases the number of segments by one by adding a new vertex to the previously produced polygonal curve. After adding a new vertex, the positions of all vertices are updated in an inner loop so that the resulting curve minimizes a penalized distance function. The algorithm is extended in (Kégl & Krzyzak, 2002) to find principal graphs of data sets, and is applied to handwritten character skeletonization.

In many aspects principal curves are related to several other unsupervised learning methods, including Vector Quantization and Generative Topographic Mapping (Chang & Ghosh, 2001). It was demonstrated in (Mulier & Cherkassky, 1995) that discretized principal curves are essentially equivalent to SOMs. In (Sun & Yang, 2007), a new definition of principal curves - Principal Curve with Feature Continuity (PCFC) - was proposed. PCFC focuses on both reconstruction error minimization and feature continuity. It builds a continuous mapping from samples to the extracted features so the features preserve the inner structures of the sample set. The properties of PCFC make it a powerful tool for the tasks of feature extraction for signal representation. Principal curves and surfaces were investigated in (Einbeck et al., 2010) in the context of multivariate regression modeling. The situation was analyzed where the intrinsic dimensionality of the data tends to be very small due to the high redundancy induced by complex dependency patterns between the involved variables. It is then useful to approximate the high-dimensional predictor space through a low-dimensional manifold (i.e., a curve or a surface), and use the projections onto the manifold as compressed predictors in the regression problem. The authors used the local principal curve algorithm for the compression step, and provided an algorithm extendible to manifolds of arbitrary dimension.

One of the first applications of principal curves was identification of ice flows and their outlines in satellite images (Banfield & Raftery, 1992). Initial estimates of floe outlines come from the erosion-propagation (EP) algorithm, which combines erosion from mathematical morphology with local propagation of information about floe edges. The edge pixels from the EP algorithm are grouped into floe outlines by grouping them about principal curves rather than points on lines. Local Principal Curve algorithm was used on the Beaver County, PA, digitalized floodplain data to reconstruct rivers in the valleys (Einbeck et al., 2005). The curve moves along the data cloud with along first local principal components. First local principal component is considered to be a (biased) approximation of the tangent to the crest line of the estimated density, gives locally the best fit. The principal graph algorithm proposed in (Kégl & Krzyzak, 2002) was used to obtain principal curves that could be served as skeletons of a fingerprint (Miao et al., 2007). Based on the obtained principal curves, a minutiae extraction algorithm is proposed to extract minutiae of the fingerprint. An application aiming at the understanding the potential impact of the microbicide on HIV viral transmission by assessing the kinetics of a microbicide lubricant was presented in (Caffo et al., 2007). The experiment was conducted by imaging a radiolabeled lubricant distributed in the subject's colon. The tracer imaging was conducted via single photon emission computed tomography (SPECT), a non-invasive, in-vivo functional imaging technique. A novel principal curve algorithm was developed to construct a three dimensional curve through the colon images. The algorithm was tested on several two dimensional images of familiar curves. An approach based upon the theory of principal

curves to find vessel lumen centerlines for the application to 3D CT angiography (CTA) was reported in (Wang et al., 2009).
In the approach presented in this chapter, the location of each vertex of piecewise linear generating curves that represent skeletons of the objects corresponds to the position of a particular SOM unit, treated here as a centre of the neural gas architecture (Martinetz et al., 1993). The proposed method makes it possible to extract the object skeletons and to reconstruct the planar shapes of sparse objects based on the topological constraints of generating lines and the estimation of local scale (Zaremba & Palenichka, 2002). In the final stage of the mapping process, the skeletal description of point sets (extracted directly, without a sequential thinning procedure) is used for tracing the final hydrographic shape by applying the principal curve algorithm between consecutive vertices of the skeleton. The experimental test results using satellite images demonstrate the accuracy of the proposed approach and its utility for fully automated mapping of hydrological objects.

2. Satellite data

2.1 Source image data

In the case of multi-dimensional image data, such as multi-spectral satellite imagery, several images of the same scene can be provided simultaneously. In remote sensing, they represent, for example, a view of the earth surface obtained by using electromagnetic radiation of different wavelengths. A pixel with image coordinates (i,j) of a multi-spectral image is a vector $\boldsymbol{b}_{i,j} = [b_1, b_2, \ldots, b_T]$ of T intensity values corresponding to T wavelength bands. The extraction and tracing of hydrological objects was performed on a data set consisting of medium-resolution Landsat imagery. Images obtained from the Landsat-7 satellite using the Enhanced Thematic Mapper Plus (ETM+) scanning radiometer instrument are produced with seven bands of reflected energy and one band of emitted energy. Since one of the eight ETM+ channels is a panchromatic channel, in Landsat-7 imagery $T = 7$. In many application areas, multi-dimensional image data are also obtained from a single image by extracting local intensity properties with respect to each image point (i,j). This approach is frequently applied to images with textured objects and backgrounds. In the case of Landsat-7, such extraction could be performed on the panchromatic channel data. The characteristics of the Landsat-7 ETM+ satellite sensor are given in Table 1.

Fig. 1. Landsat-7 image of the Eastern Ontario area

Spectral Band	Half-Amplitude Bandwidth (μm)	Ground Resolution (m)
1	0.450±0.005 - 0.515±0.005 (Blue)	30
2	0.525±0.005 - 0.605±0.005 (Green)	30
3	0.630±0.005 - 0.690±0.005 (Red)	30
4	0.775±0.005 - 0.900±0.005 (Near IR)	30
5	1.550±0.010 - 1.750±0.010 (Mid IR)	30
6	10.40±0.100 - 12.50±0.100 (Thermal)	60
7	2.090±0.020 - 2.350±0.020 (IR)	30
8	0.520±0.010 - 0.900±0.010 (Panchromatic)	15

Table 1. Spectral and spatial characteristics of Landsat-7

Landsat-7 images were pre-processed at level 1G: The 1G level image is a radiometrically and geometrically corrected raw level 0R image. The main scene of the study covers the area of the Ottawa region (Fig. 1), with a diversified river system: the Ottawa River with wide sections and islands of different size, the Gatineau River in the northern part of the image, and the Rideau river with some sections narrower than the image resolution.

3. Extraction of water regions

3.1 Topology of the network

The extraction is performed using pixel-by-pixel multispectral classification using unsupervised image segmentation and a set of labelled examples. The mapping from the space of spectral features to the space of terrain classes is performed by a self-organizing feature map (SOM) architecture.

The classical (SOM) is a two-layer network, where each input layer neuron x_j has a feed-forward connection w_{ij} to each output layer neuron y_i [Kohonen et al., 1996]. A feature map is created by adjusting the connection weights from m-dimensional-vector input nodes to n-dimensional interconnected output nodes ($n < m$). The topology of a three-dimensional self-organizing feature map is shown in Fig. 2. The resulting network has the ability to adapt its

behavior in such a way that the location (k, l, m) of the activated output neuron in the network becomes specific to certain characteristic features (a_k, b_l, c_m) of the input signal X.

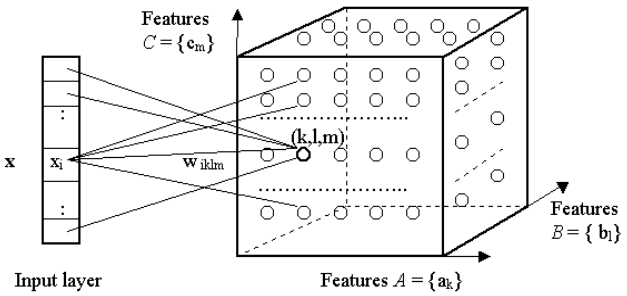


Fig. 2. Topology of a three-feature self-organizing map

In this application, the definition of the input vector consists in finding a discriminative set of spectral bands from Table 1. The selection is performed based on the distribution of pixels for the water and the non-water regions. An example of the distribution for the Near-IR band (Band 5) is shown in Fig. 3.

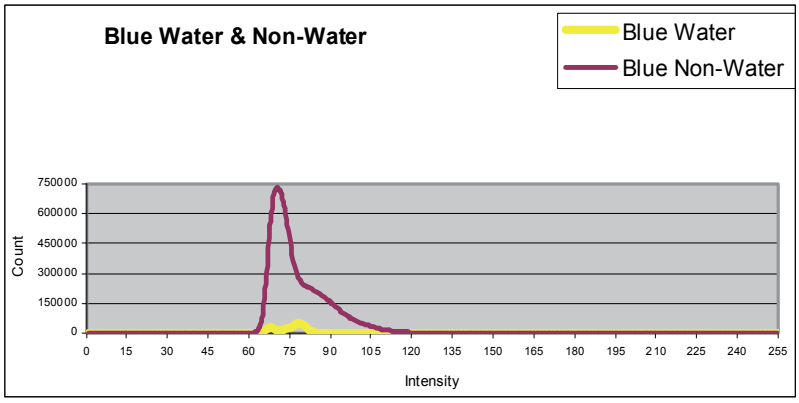


Fig. 3. Distribution of pixel blue-channel intensity for water and non-water regions

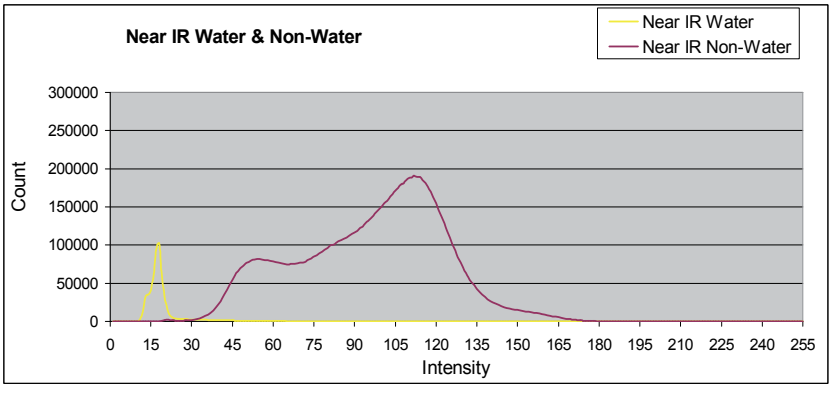


Fig. 4. Distribution of pixel Near IR-channel intensity for water and non-water regions

The distributions in Figures 3 and 4 show that the Near IR channel is more discriminative than the Blue channel. Combinations of channels offer better classification results than using a single channel. For the final architecture of the classification network, three features were selected, corresponding to the Green, Blue and Near IR bands.

3.2 Learning method

The lesrning method consists of two phases: the SOM network learning phase, enhanced by the use of fuzzy logic paradigm, and the merging and evaluation phase.

Network learning phase

The learning of the network consists in updating the connection weight vector **w**. Let, for a 3-D network, $\mathbf{x} = (x_1, x_2, x_3)^T$ represent the input vector and w_{ij} represent the weight matrix for the weights between unit *j* of the input layer (j = 1,2,3) and unit *i* of the output layer (let us assume, for simplicity, a linear order of the output neurons). The basic learning algorithm takes place in three steps.

1. Computing of matching values for each unit in the output layer. If we define the winner with index c, the neuron with the smallest network input

$$|| x - w_c || = \min\{|| x - w_{ij} ||\} \tag{1}$$

   is declared the winner neuron.

2. Updating the weights. The weights corresponding to the winner unit and its neighboring units are updated according to the formula:

$$\begin{aligned} w_j(k+1) &= w_j(k) + \frac{\sum_{i=1}^{n} \alpha_{ij}(k)^{m(k)} (x_i(k) - w_j(k))}{\sum_{i=1}^{n} \alpha_{ij}(k)^{m(k)}} && \text{for } i \in \text{topological neighbourhood} \\ w_j(k+1) &= w_j(k) && \text{otherwise} \end{aligned} \tag{2}$$

   where

$$w_i = \frac{\sum_{i=1}^{n} x_i (\alpha_{ij})^m}{\sum_{i=1}^{n} (\alpha_{ij})^m} \tag{3}$$

   The topological neighborhood, may be defined in several ways. In a typical choice for the neighborhood function it falls off with distance between the output hypercube units. Thus nearby neurons receive stronger updates and end up responding more strongly to nearby input patterns. This arrangement enables the topological information to be fed into the network (Villmann et al., 1997). The presence of uncertainty and non-Gaussian, nonstationary statistics the methodology of fuzzy sets offers significant advantages. In the fuzzy set paradigm, a pixel can belong to several clusters with different certainty α mapped to a range [0,1]. Fuzzy C-means was the first algorithm to incorporate fuzzy login in the learning process. It is an iterative algorithm used to find the grades of membership α_{ij} and cluster centers w_j to minimize the criterion

$$J = \sum_{i=1}^{n}\sum_{j=1}^{c}(\alpha_{ij})^m \mid x^i - w_j \mid^2 \tag{4}$$

where c is the number of clusters, and n represents the number of input samples. A major drawback of the algorithm is that it becomes time-consuming when the number of clusters is large. The combination of SOM and fuzzy C-means can overcome the drawbacks of both methods and improve the performance of classification. In the fuzzy SOM (FSOM), the learning rate of SOM can be replaced by the membership values calculated as

$$\alpha_{ij} = \left[\sum_{k=1}^{c}\left(\frac{\mid x^i - w_j \mid}{\mid x^i - w_k \mid}\right)^{\frac{2}{m-1}}\right]^{-1} \tag{5}$$

The rate is related to the distance between the centers of the clusters and the samples. The clusters closest to the sample (the winner) take the largest membership value. Unlike fuzzy C-means, which calculates all membership values, for fuzzy SOM the k in (5) relates to the size of neighbourhood, so only membership values of the winner and its neighbourhood are calculated. In order to reduce the computational time, the parameter m in equation (5), which is equivalent to the neighbourhood size, changes with time according to:

$$m(k) = m_0(1 - e^{-E(k)}) + m_{\min} \tag{6}$$

where m_0 and m_{min} are positive constants greater than one and E(k) is the Euclidean distance between successive weight values.

3. Calculate Euclidean distance

$$E(k) = \max\{ \|w_j(k+1) - w_j(k)\|^2 \} \tag{7}$$

If $E(k)<\varepsilon$, stop. Else, go to Step 2.

Merging and evaluation phase

An essential question that has to be addressed in the design of a SOM network is the size of the network, i.e., the number of neurons. The network can not be too small so that patterns which are important but can be distinguished only among a large number of patterns are not missed. On the other hand, an excessive number of neurons may hinder the detection of a pattern by splitting it into several sub-patterns. The optimal size of the network can be determined by initially selecting a large network, and subsequently limiting its size through merging closely located neurons.

Assume two neurons are defined by the weights $\mathbf{w}_i = (w_{i1}, w_{i2} \ldots w_{in})$ and $\mathbf{w}_j = (w_{j1}, w_{j2} \ldots w_{jn})$. If $\| w_i - w_j \| < T$, where T is a pre-specified threshold, then the two corresponding clusters can be merged. The new neurons will be defined by

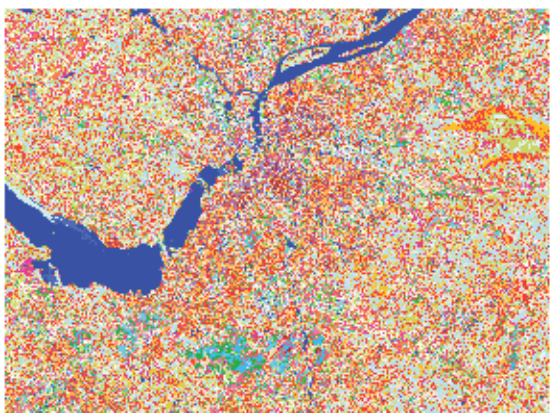
$$\mathbf{w}_{new} = \frac{n\mathbf{w}_i + m\mathbf{w}_j}{n+m}, \tag{8}$$

where n and m are the hit rates (input vectors corresponding to the neuron). The more the input data activate the neuron, the greater the value of the weight assigned.

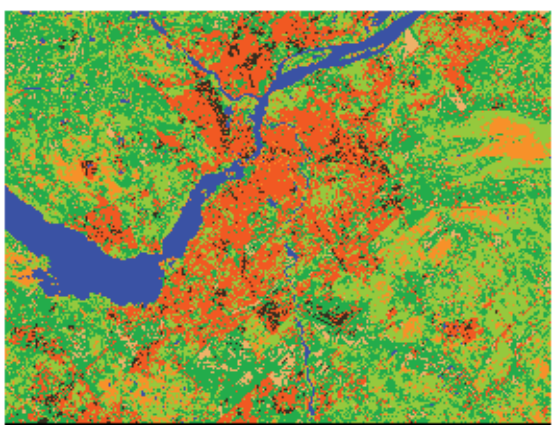
A cluster validity method is required in order to measure the quality of the merging results. We defined the following quality function:

$$V = \frac{\frac{1}{n}\sum_{i=1}^{c}\sum_{x\in X_i}||x_k - w_i||}{\min_{i,j}||w_i - w_j||} \tag{9}$$

where $c > 2$ is the total number of neurons, and n is the total number of samples. The optimal number of neurons is that for which V becomes minimum. The parameter V is the ratio of the average distance between the inputs and their corresponding neuron to the minimum distance between neurons. The inputs should be made as close to their corresponding neurons as possible, and the neurons distributed as far apart as possible. The procedure of merging neurons continues as long as the distance between the neurons remains smaller than a given threshold value. Different thresholds can be selected, with a subsequent use of V to evaluate the quality of merging results.



a)



b)

Fig. 5. Clustering results a) 100 nodes, b) 7 nodes

In summary, the procedure for merging neurons consists of the following steps:

1. Select T; load the learned neurons and the hit rates of the neurons.
2. Calculate the smallest Euclidean distance between pairs of neurons. If the distance is smaller than T, merge the two neurons according to equation (8).
3. Repeat Step 2 based on the values of neurons after merging. If the smallest distance is larger than T, stop the iteration.

The number of clusters that has the smallest V, for different T, is the final optimal number of clusters.

Typical results of the merging procedure are depicted in Fig. 5. Figure 5a shows the segmentation of the satellite image into 100 classes. The results of merging the neurons into a network with 7 units are shown in Fig. 5b.

The optimum results were obtained for 25 nodes. The resulting water areas are shown in Fig. 6.

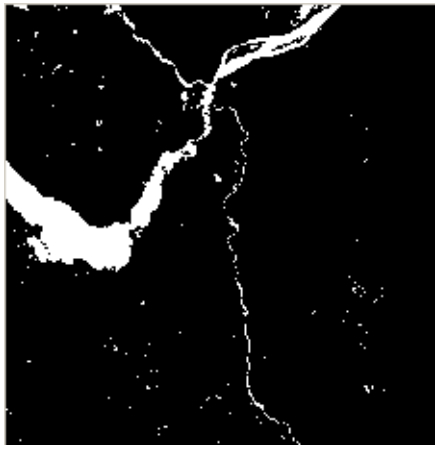


Fig. 6. Binary map of the water bodies

4. Mapping and tracing of hydrographic networks

The basic idea of the hydrographic network tracing algorithm consists in generating principal curves between the vertices of the network skeleton graph. In the first stage of the process, the set of vertices of the principal curve skeleton graph is constructed by applying a SOM-based algorithm. The algorithm consists in iterative updating of vertex coordinates in the data space while progressively decreasing the span of a SOM kernel function. The resulting piece-wise linear skeleton is constructed as follows:

Step 1: Initialization. The input data vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N$ represent the image plane coordinates of input binary image $\phi(m,n)$ obtained by the segmentation process, such as shown in Figure 6. The SOM units $u_1, \ldots, u_K$ are initialized in the two-dimensional data space, with weight vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K$ representing the grid topology. The initial number of units K should be comparable with the maximal number of vertices in the object skeletons. The scale attributes of all units are set to the maximal value, which is comparable with the image size. This means that each unit is initially connected to its q nearest neighboring SOM units, where q is the maximal number of vertices.

Step 2: Determination of vertex neighborhood connectivity. The SOM units with the same scale value are connected into a local structure by checking the connectivity between the

neighborhood units. During the first iteration, the full connectivity is established, the unit scales being all equal and maximal in size.

Step 3: Matching point coordinates with vertices. With the Euclidean distance used as a similarity measure, this step corresponds to the Voronoi tessellation of input data set. The number a SOM unit in the topological space closest to the *i*th data point is determined by:

$$z_i = \arg\min_r \{ \| \boldsymbol{v}_i - \boldsymbol{w}_r \| \}, \qquad i = 1, \ldots, N \tag{10}$$

As a result, the object support regions of the input binary image $\phi(m,n)$ will be partitioned into N Voronoi regions $\{V_i\}$. The new weights of the rth unit will be determined based on pixels belonging to the rth Voronoi region and the regions neighboring to V_r.

Step 4. Evaluation of vertex scale. The scale attribute of the rth SOM unit, ρ_r, is estimated by considering the Voronoi regions for the SOM units, and is proportional to the number of points in the region V_r. The more precise scale estimate is made by finding the size of the maximal disk inscribed into the rth Voronoi region:

$$\rho_r = \arg\max_k \left\{ \max_{(u,v)\in V_k} \left\{ \left| \frac{1}{|S_k|} \sum_{(m,n)\in S_k(u,v)} \varphi(m,n) - \frac{1}{|R_k|} \sum_{(m,n)\in R_k(u,v)} \varphi(m,n) \right| \right\} \right\} \tag{11}$$

where $R_k(u,v)$ is kth disc and $S_k(u,v)$ is a ring around it. The ring width is equal to the minimal distance between two objects considered as isolated. Such a scale estimate is used because of the assumption of shape sparseness (the direct method of maximal inscribed disk produces poor results). The point $\boldsymbol{t}_r = (u_r, v_r) \in V_r$, which corresponds to the maximal value of scale (Eq. 10), is called the *attraction point* of region V_r. The use of such a scale estimate and attraction point is useful when calculating conditional expectation of unit coordinates in the next step of this algorithm. Figure 7 depicts the detected vertices represented as circles proportional to the vertex scales.

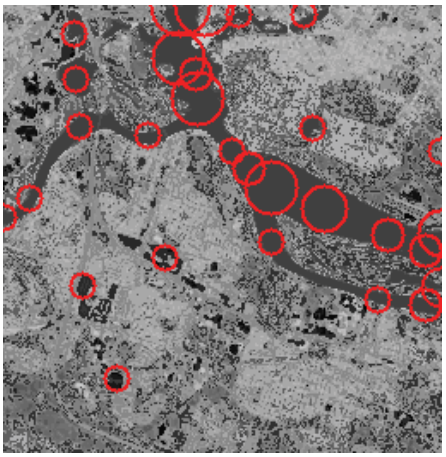


Fig. 7. Vertices represented as circles of different scale

Step 5: Updating of coordinates using conditional expectation. The weights of SOM units, i.e., the vertex coordinates, are updated at this stage. The approach of conditional expectation [Singh et al., 2000] is used here, modified in order to take into account the current unit scale:

$$\boldsymbol{w}_r = \frac{\sum_{i=1}^{K} \boldsymbol{v}_i \Phi(\boldsymbol{w}_r, \boldsymbol{w}_i, \boldsymbol{t}_r)}{\sum_{i=1}^{K} \Phi(\boldsymbol{w}_r, \boldsymbol{w}_i, \boldsymbol{t}_r)}, \quad r = 1, \ldots, K \tag{12}$$

where $\Phi(\boldsymbol{w}_r, \boldsymbol{w}_i, \boldsymbol{t}_r)$ is a monotonically decreasing kernel function defined indirectly in terms of two intermediate variables: distance $\Delta_1(\boldsymbol{w}_r, \boldsymbol{w}_i)$ between the positions of the *r*th and the *i*th SOM units, and distance $\Delta_2(\boldsymbol{w}_r, \boldsymbol{t}_r)$ between the position of unit *r* and the attraction point $\boldsymbol{t}_r \in V_r$. A Gaussian kernel function has been used as the smoothing kernel function Φ(.) of the total distance $\Delta_1+\Delta_2$:

$$\Phi(\Delta_1 + \Delta_2) = \exp\left(-\frac{(\Delta_1 + \Delta_2)^2}{2\sigma^2}\right) \tag{13}$$

where σ is the span of the kernel function. The distance function $\Delta_1(\boldsymbol{w}_r, \boldsymbol{w}_i)$ is defined between SOM neighborhood units connected to u_i. The simultaneous use of two variables in the kernel function allows the algorithm to adapt to the segments of objects of different scales of interest. The smoothing function (Eq. 13) represents the conditional probability that the point $\boldsymbol{v}_i$ belongs to unit u_r at the current iteration step.

Step 5. Determination of SOM vertices connectivity graph. During this step, the connectivity is determined between all non-connected vertices and the local structures. Vertices and elementary straight-line segments are also determined. If a vertex is connected to a line segment, then this vertex is added to the skeleton and a new SOM unit with its local connectivity attributes is inserted into the list of units. Such a vertex insertion provides more exact approximation of skeletons with crossing lines and other higher-order connections.

In order to avoid ragged skeleton lines for elongated thick objects and false connections between skeleton vertices of disconnected object parts, which often result from applying the Minimum Spanning Tree algorithm (Singh et al., 2000), The decision about the vertex connectivity is made by using a context-dependent connectivity test. The test is based on the Markov random chain model of vertices belonging to the same elementary curve and the Bayesian principle of a decision-making process. The Markov chain model is used here to provide the probability of the position of the next vertex with respect to the positions of the two immediately preceding and connected vertices. This model of the curve skeleton macro-growth process can be defined by the conditional probabilities of the new vertex position with respect to the positions of the previous two adjacent vertices on the same curve. Let u_s, u_l and u_k be three consecutive vertices which generate segments $[u_s, u_l]$ and $[u_l, u_k]$. The Markov chain model of a skeleton piece-wise linear graph *G* can be defined in terms of a conditional probability of the slope $\theta_{l,k}$. of line segment $[u_l, u_k]$ with respect to the slope $\theta_{s,l}$ of its preceding straight-line segment $[u_s, u_l]$:

$$\{P(\theta_{l,k}, [u_l, u_k] \subset G) = P(\theta_{l,k} / \theta_{s,l}, \forall\{[u_s, u_l], [u_l, u_k]\} \subset G\}, \tag{14}$$

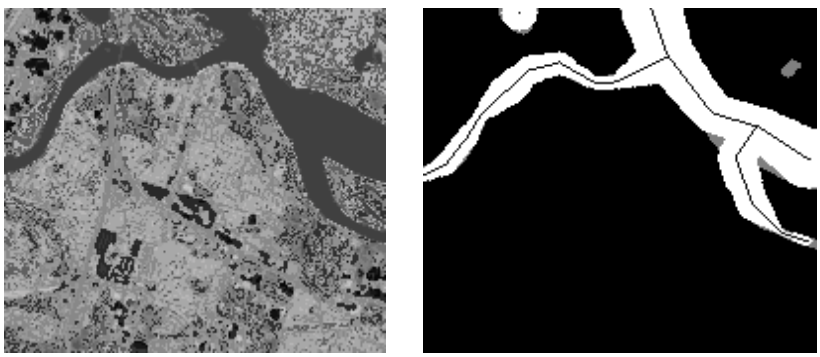
where $P(\theta_{l,k} / [u_k, u_l] \in G)$ is the unconditional probability of the slope of the straight-line segment between vertices *k* and *l* of the same line *G*, and $P(\theta_{k,l} / \theta_{s,l}, \forall\{[u_s, u_l], [u_l, u_k]\} \subset G\}$ is the conditional probability of the slope of the straight-line segment between vertices *k* and *l*

of the same generating line G provided that the slope of straight-line segment $[u_s, u_l] \subset G$ equals $\theta_{s,l}$. After adopting a limit probability, δ, in the connectivity test by a probability thresholding, i.e., $P(u_k \wedge u_l / x) \geq \delta$, the connectivity test can be written in an explicit form:

$$\frac{P(x / u_k \wedge u_l)}{P(x / u_k \,||\, u_l)} \geq \frac{\delta}{(1-\delta)} \cdot \frac{P(u_k \,||\, u_l)}{P(u_k \wedge u_l)}, \tag{15}$$

where $P(u_k \,||\, u_l) = 1 - P(u_k \wedge u_l)$ is the prior probability of no connection between vertices u_k and u_l, and $P(x / u_k \,||\, u_l)$ is the conditional probability of the connectivity feature x provided vertices u_k and u_l are not connected. The connectivity test by Eq. (15) guarantees that the error due to a wrong connection of vertices will be less than $(1 - \delta)$.

Figure 8 gives an example of the results of skeletonization of a network of hydrographic objects. The piece-wise linear approximation of the principal curve is shown. The skeleton demonstrates the utility of the presented approach in a situation of multiple rivers. In addition to showing the skeleton graph, the white form in Fig. 8b depicts also the results of a reconstruction of the hydrographic shape in Fig. 8a from the skeleton. The reconstruction consists in interpolating the scales of adjacent vertices along the skeleton segments. Only large values of the vertex scales (see Step 4) were used.



a) b)

Fig. 8. Skeleton graph (b) for the water basin in (a)

In the second stage of the hydrographic network tracing process, the vertex points are used for tracing the river by connecting them by a principal curve. A curve **C** is said to pass through the middle of a dataset if every point **x** on the curve is the average of the observations projecting onto it. The Hastie and Stuetzle definition of principal curves is based on probability densities. Let **X** denote a two-dimensional random vector distributed according to a probability density p, and let $\mathbf{C} \subset \mathbf{R}^2$ be a smoothly embedded closed interval. For each point $\mathbf{x} \in \mathbf{R}^2$, let $d(\mathbf{x}, \mathbf{C})$ denote the distance from **x** to **C**. Because **C** is compact, for each $\mathbf{x} \in \mathbf{R}^2$ the distance $d(\mathbf{x}, \mathbf{C})$ is realized by at least one point of **C**. The *projection map*

$$\pi_{\mathbf{C}} : \mathbf{R}^2 \rightarrow \mathbf{C} \tag{16}$$

is the map which assigns to each $\mathbf{x} \in \mathbf{R}^2$ a point $\pi_{\mathbf{C}}(\mathbf{x}) \in \mathbf{C}$ realizing the distance from **x** to **C**, that is,

$$d(\mathbf{x}, \mathbf{C}) = \| x - \pi_{\mathbf{C}}(\mathbf{x}) \|. \tag{17}$$

We can now formalize a definition of a principal curve. A curve **C** is called *self-consistent* or a *principal curve* of a density p if $E(\mathbf{X} \mid \pi_C(\mathbf{X}) = \mathbf{x}) = \mathbf{x}$ for almost every $\mathbf{x} \in C$.
The notion of projection also leads to a natural definition of the distance between a random vector **X**, or its associated density, and a curve **C**:

$$d^2(\mathbf{x}, \mathbf{C}) = \mathrm{E}(\|\mathbf{X} - \pi_C(\mathbf{X})\|^2). \tag{18}$$

As proved in (Hastie & Stuetzle, 1989), principal curves are critical points of the distance in the variational sense. Since all principal curves are saddle points of the distance, there are no local minima. In order to better illustrate this point, let us set this property against the regression problem, where the conditional expectation $\mathrm{E}(Y \mid x)$ minimizes the expected squared distance $\mathrm{E}(Y - f(X))^2$ among all functions f. Although principal curves are not local minima, in general, they are local minima of the distance for "low frequency variations" (Duchamp & Stuetzle, 1996). The definition of "low-frequency" depends on the principal curve itself, and not only on the underlying density. In practical situations, and for the type of shapes required to trace hydrographic objects, this limitation of principal curves is not critical.
An example of a river traced using a polygonal principal curve is shown in Figure 9. The examples shown in Figures 8 and 9 illustrate the versatility and good performance of the SOM-based approach to solve the problem of tracing hydrographical systems. The presented method works on long stretches of rivers. It can also effectively deal with a system of connected, tributary rivers.

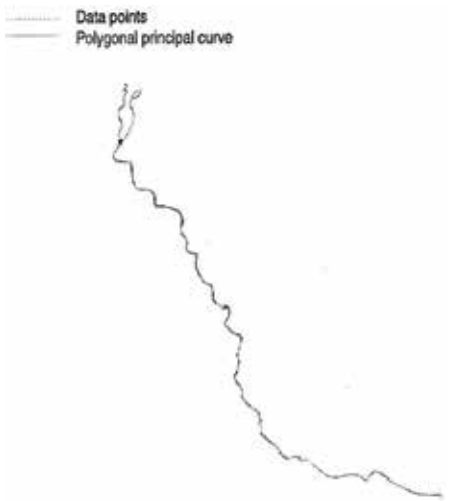


Fig. 9. A river mapping example

5. Conclusion

A semi-automated approach to the extraction and tracing of river networks was presented, where SOM networks are used in both stages of the hydrological mapping process, i.e., the detection of water bodies from multispectral images and the subsequent tracing of hydrological systems or networks. By adopting the principal curve approach, the algorithms precented in this chapter permit to effectively deal with sparse image data. The tracing

method makes it possible to extract the river skeletons based on the topological constraints of river segments and the estimation of local scale. A distinctive feature of the proposed SOM-based approach is the introduction of such attributes as local scale, used in the scale-based updating phase of SOM units, and connectivity of SOM units. A probabilistic model-based connectivity test is performed while connecting adjacent SOM units, i.e., skeleton vertices. The presented method for the extraction of the skeletal shape and the tracing of hydrographic objects by using structured self-organizing maps was tested on Landsat-7 ETM+, SPOT and QuickBird satellite images, demonstrating the utility of the presented approach.

6. Acknowledgement

The work presented in this chapter was supported by the GEOIDE Network of Centres of Excellence and the Natural Sciences and Engineering Research Council of Canada.

7. References

Ainsworth, E.J. & Jones, I.S.F. (1999). Radiance spectra classification from the ocean color and temperature scanner on ADEOS, *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 37, No. 3, pp. 1645-1656.

Auclair Fortier, M.-F.; Ziou, D.; Armenakis C. & Wang, S. (2001). Automated Correction and Updating of Roads Databases from High-Resolution Imagery, *Canadian. J. Remote Sensing*, Vol. 27, No.1, pp. 76-89.

Banfield, J. D. & Raftery, A. E. (1992). Ice floe identification in satellite images using mathematical morphology and clustering about principal curves, *Journal of the American Statistical Association*, Vol. 87, No. 417, pp. 7-16.

Caffo, B.S.; Crainiceanu, C.M.; Deng, L. & Hendrix, C.W. (2007). A case study in pharmacologic imaging using principal curves in Single Photon Emission Computed Tomography, *Working Paper 143* (May 2007), John Hopkins University, Dept. of Biostatistics.

Chang, K.-Y and Ghosh, J. (2001). A unified model for probabilistic principal surfaces, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 23, No. 1, pp. 22-41.

Datta, A.; Parui, S.K. & Chaudhuri, B.B. (1996). Skeletal shape extraction from dot patterns by self-organization, *Proc. 13th Int. Conf. Pattern Recognition*, Vol. 4, 1996, pp. 80-84.

Dillabaugh, C.R.; Niemann, K.O. & Richardson, D. (2002). Semi-Automated Extraction of Rivers from Digital Imagery, *Geomatica*, Vol. 6, No. 3, pp. 263-284.

Duchamp, T. & Stuetzle, W. (1996). Extremal properties of principal curves in the plane. *The Annals of Statistics*, Vol. 24, No. 4, pp. 1511-1520.

Einbeck, J.; Tutz, G. & Evers, L. (2005). Local Principal Curves. *Statistics and Computing*, Vol. 15, pp. 301-313.

Einbeck, J.; Evers, L. & Powell, B. (2010). Data compression and regression through local principal curves and surfaces. *International Journal of Neural Systems*, Vol. 20. pp. 177-192.

Haralick, R.M. & Shapiro, L.G. (1992). *Computer and Robot Vision*, Addison-Wesley, Reading, MA.

Hastie, T. & Stuetzle, W. (1989). Principal curves, *Journal of the American Statistical Association*, Vol. 84, No. 406, pp. 502-516.

Jenson, S.K. & Domingue, J.O. (1988). Extracting topographic structure from digital elevation data for geographic information system analysis. *Photogrammetric Engineering and Remote Sensing*, Vol. 54, pp. 593-600.

Kégl, B.; Krzyzak, A.; Linder, T. & Zeger, K. (2000). Learning and design of principal curves, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 3, pp. 281-297.

Kohonen, T.; Oja, E.; Simula, O.; Visa, A. & Kangas, J. (1996). Engineering applications of the self-organizing map. *Proceedings of the IEEE*, Vol. 84, pp. 1358-1384.

Ma, W.-Y. & Manjunath, B.S. (2000). EdgeFlow: A Technique for Boundary Detection and Image Segmentation, *IEEE Trans. Image Processing*, Vol. 9, pp. 1375-1388.

Martinetz, T.; Berkovich, S. & Schulten, K. (1993). Neural-gas network for vector quantization and its application to time-series prediction, *IEEE Trans. on Neural Networks*, Vol. 4, No. 4, pp. 558-569.

Miao, D.Q; Tang, Q.S. & Fu, W.J. (2007). Fingerprint minutiae extraction based on principal curves, *Pattern Recognition Letters*, Vol. 28 , Issue 16 (December 2007), pp. 2184-2189.

Mulier, F. & Cherkassky, V. (1995). Self-organization as an iterative kernel smoothing process, *Neural Computation*, Vol. 7, No. 6, pp. 1165-1177.

Ritter, H.; Martinetz, T. & Schulten, K. (1992). *Neural Computation and Self-Organizing Maps: An Introduction*. Addison-Wesley, Reading, MA, 1992.

Singh, R.; Cherkassky, V. & Papanikopoulos, N. (2000). Self-Organizing Maps for the Skeletonization of Sparse Shapes, *IEEE Trans. on Neural Networks*, Vol. 11, pp. 241-248.

Sun, M. & Yang, J. (2007). Principal curves with feature continuity. Proceedings of the 11th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, Nanjing, China, May 22 - 25, 2007. In: *Lecture Notes In Artificial Intelligence*. Zhou, Z., Li, H. & Yang, Q. (Eds.), pp. 785-792, Springer-Verlag, Berlin, Heidelberg.

Ton, J.; Sticklen, J. & Jain, A.K. (1991). Knowledge-Based Segmentation of Landsat Images, *IEEE Trans. on Geoscience and Remote Sensing,*Vol. 29, pp. 222-231.

Villmann, T.; Der, R.; Herrmann, M. & Martinez. T.M. (1997). Topology preservation in self-organizing feature maps: Exact definition and measurement. *IEEE Trans. on Neural Networks*, Vol. 8, No. 2, pp. 256-266.

Wong, W.C. K.; So, R.W. K. & Chung, A.C. S. (2009). Principal Curves: a Technique for Preliminary Carotid Lumen Segmentation and Stenosis Grading, *The MIDAS Journal*, July 29, 2009. http://hdl.handle.net/10380/3096.

Zaremba, M.B. & Palenichka, R.M. (2002). Probabilistic morphological modelling of hydrographic networks from satellite imagery using self-organizing maps. *Control and Cybernetics*, Vol. 31, No. 2, pp. 343-369.

Part 6

Application of SOM in Medical and Biological Sciences

19

Computational Approaches as a Tool to Study Developmental Biology in New World Primates

Maria Bernardete Cordeiro de Sousa, Allan Medeiros, Dijenaide Chaves de Castro, Adriano de Castro Leão and Adrião Duarte Dória Neto
Universidade Federal do Rio Grande do Norte, Brazil

1. Introduction

Kohonen neural network or SOM (Self-Organizing Maps) and k-means are effective computational techniques for extracting information from a large dataset, in some cases, more suitable than traditional statistical methods (such as regression) for detecting multivariate data patterns. We used these two computational approaches to study the developmental biology of a large dataset of weight versus age in common marmosets, seeking to provide a more precise description of the onset and end of behaviors during ontogenesis. Both techniques perform clustering, a technique that falls into a group of undirected data mining tools. The goal of undirected data mining is to discover structure in the data as a whole, separating or discovering classes or groups that are not easily detected from direct data inspection. Both k-means algorithms and self-organizing maps are clustering methods that classify patterns without the need of previous information on data distribution, as occurs in supervised methods. When used in feature datasets (for instance, age and weight), these techniques allow a data mining process that identifies stages and substages of the data distribution being used. In particular, the SOM algorithm is able to map features from a high- dimensional space into a one- or two- (in fact any dimension less than the dimension of the data) dimensional space with preservation of topology. Thus, these tools are better for modeling and identifying stages in a sample pool and they enable us to visualize data distribution in a two-dimensional space, even when the number of features in the data is high (or the topology is complex). For a same database we found that both approaches converged to a same result, enabling the process of classification, thus achieving the goal of this study. Biomedical research using the common marmoset (*Callithrix jacchus*) continues to grow, through the investigation of different topics such as infectious diseases, neuroscience and development, toxicology and drug development, reproductive biology and behavior. Using this classification we found that the infantile stage of the current classification could be divided into three stages and that feeding behavior is statistically different among them. We also applied a statistical method, Principal Component Analysis (PCA) to evidence the sex difference. Using data collected from 9 animals (4 males and 5 females) during development evidenced a sex difference, where male behavior features are more dispersing than those of females, indicating that behavior is more changeable in males. These findings indicate that the use of clustering techniques provides a more precise description of changes during ontogenesis in common marmosets

in addition to being useful in refining the follow-up of the ontogenetic development of common marmosets in terms of behavioral and physiological variables.

2. Computational tools and the study of animal behavior

There is currently great interest in computational intelligence and statistical methods applied to the study of animal behavior. These groups of techniques are very useful and powerful in analyzing natural behavior, simulating physiological models and measuring the complex sequential or isolated neural circuits of motor activities, vocalization or sensory processing among others.

According to Enquist & Ghirlanda (2005), the use of artificial neural networks (ANN) began as an effort to understand the nervous system and behavior based on the fundamental arrangement of the cytoarchitectural components of a system formed by a complex network of neurons. Moreover, this knowledge is multidisciplinary, encompassing mathematics, theories of intelligent machines and philosophy. Although artificial neural networks have provided evidence that this model is a powerful computational tool, applicable to a wide range of phenomena, Harris-Warrick & Marder, (1991), seeking to develop a model for an invertebrate brain, observed that high computational processing is necessary, even for these simple animals. However, the overwhelmingly promising findings compensate for these effects, and today many laboratories, mostly engineering, are working in this area. Enquist & Ghirlanda, (2005) also pointed out that students of animal behavior should have embraced the study of neural network models, but unfortunately, this did not occur, since traditional ethological and animal psychological thinking does not include a discipline related to this matter. Another important question involved with the use of integrative neural networks in neuroscience is the need to make a critical comparison of available theoretical approaches (Reeke and Sporns, 1993), considering their many applications. The contribution of neurophysiology to the development of artificial neural networks is very important and many situations, such as the demonstration of visual receptive fields by Hubbell & Wiesel (1962), the invertebrate models of learning developed by Kandel & Pittenger (1999) and the recent study by Tin et al., (2008) where the authors developed a Bayesian linear regression algorithm that confirms recent findings on the organization of the motor cortex, providing an incremental real-time version for real-time interfaces between brains and machines.

The use of statistical techniques to analyze animal behavior has many applications in a wide range of biological activities, from molecular mechanisms to complex animal behavior. Very recent studies using black lemurs (*Eulemur macaco*: Pozzi et al., 2010) showed that ANN's describe all seven vocal categories and are more suitable than other statistical analysis methods, extracting from both temporal and spectral characteristics. Matthews (2009) studied the socially learned traditions white-fronted capuchin monkeys in the field (*Cebus albifrons*) by combining variations in multiple foraging techniques into pairwise behavioral similarity matrices. He found that cluster analysis makes it possible to predict specific features from the social learning hypothesis.

Animals living under natural conditions have to deal with social and environmental changes and unpredictable situations that trigger behavioral adjustments. A number of ethologists and comparative psychologist have studied animal behavior during adult life, when animals exhibit a more stable pattern of responses and the quantification of these variables is more predictable. During ontogenesis, i.e. during the development individuals

during their lifetime, a detailed approach is necessary to provide a more accurate picture of developing features, the onset and end of transient behaviors such as weaning and being carried, as well as the start and stabilization of most adult behaviors. However, even when the behavioral repertoire is fully developed, changes will occur owing to new challenges demanding requirements, becoming the expression of behavior that is flexible enough to allow survival in hostile environmental conditions. Thus, in both cases, in adult life and, mainly during ontogeny, individual behaviors also change with experience and when animals are faced with unpredictable situations.

Common marmosets *(Callithrix jacchus* - Linnaeus, 1758) are small-bodied non-human primates, endemic to Brazil with an adult weight of around 320 g. They are considered omnivores and in the last 10-15 years have extended their habitats beyond secondary forests to exotic species plantations, peridomiciliary areas and public parks. They live in free-ranging groups composed of between 5-17 different- aged individuals, including adult males and females (Rylands, 2006; Araujo, 1996). They exhibit a flexible mating system, depending on ecological constraints, showing that common marmosets are preferentially monogamous but on some occasions a polygynous system may arise (Arruda, 2005; Sousa et al., 2005).

Both mature and immature males and females emigrate from the natal group to build new social units, but this behavior is more frequent in females (Sousa et al., 2009). Subordinates are expelled by breeding (dominant) females or leave the group voluntarily showing or not functional ovulatory cycles (Albuquerque et al., 2001; Sousa et al., 2009; Yamamoto et al., 2010). Studies on social dynamics within free-ranging common marmoset groups have shown that breeding males and females form a nuclear structure (Koening and Rothe, 1991; Araujo, 1996) surrounded by satellite non-breeding individuals that are inspected in terms of reproductive condition. Breeding pairs try to maintain reproductive exclusivity using reproductive inhibition by means of aggressive behavioral mechanisms that limit the mate's copulation opportunities with other individuals (Araujo & Yamamoto, 1993; Yamamoto et al., 2010). Aggression is frequent when the home ranges of neighboring free-ranging groups overlap. Males and females become more aware, increasing scent marking, vocalizations and vigilant behaviors near territorial borders. On these occasions, competition is intense and sexual contact between males and females and extra-group individuals is frequently recorded (Lazaro-Perea et al., 2000; Lazaro-Perea, 2001). Dimorphic behavior in captive common marmoset males and females has been recorded in relation to infant carrying behavior, where males carried more during the first weeks of development, and in the physiological characteristics of females that may become pregnant again in the first weeks of the postpartum period (Kendrick & Dixson, 1986). Other studies using captive marmosets described additional sexual dimorphism in physiological variables, with females showing three times higher cortisol levels (Johnson et al., 1996; Raminelli et al., 2001) and higher aggressive behavior toward same sex individuals (Galvão-Coelho et al., 2008) than males.

The common marmoset is a very useful model in stress-induced situations. Galvão-Coelho et al., (2008) recently demonstrated that common marmosets exhibit behavioral changes during challenging situations. In adult subjects, basal cortisol profiles varied in a population (n= 48; 24 males), allowing their use as a model for studying mental disorders associated with high and low responsiveness of the hypothalamic-pituitary-adrenal (HPA) axis. Since the social behavior of common marmosets has been extensively studied in both situations (free-ranging and captive groups), computational tools seem to be a promising

path to the discovery of new components or features of behavior that may provide fresh insights into the sociobiology of this species.
Clustering tools such as k-means and principal component analysis techniques are alternatives to using artificial neural networks. In this chapter we present the results obtained regarding the effectiveness of k-means applied to the analysis of weight to characterize the infantile stage, corresponding to a period of rapid ontogenetic development in mammals. We also show the usefulness of PCA in identifying sex differences in the pooled data of individual and social behaviors expressed by common marmosets during development.

3. Is weight a good measure to characterize the early developmental stages of common marmosets?

Leão et al. (2009) have demonstrated that classifying the lifetime of common marmosets (*Callithrix jacchus*) using SOM and k-means techniques in experimental protocols where the variables "weight" and "age" are critical could generate new possibilities for refining data analysis during developmental stages previously characterized by behavioral recordings. The authors employed clustering techniques, where data mining is used to discover structure in the data as a whole, and substages in the infantile and juvenile stages described by Yamamoto, 1993. Both k-means algorithm (Duda et al., 2000) and self-organizing maps (Kohonen, 1982) are clustering methods that classify patterns without the need of previous information on data distribution, in contrast supervised methods. These techniques allow a data mining process that identifies stages and substages of data distribution. These tools are better for modeling and identifying stages in a sample pool and they enable us to visualize data distribution in a two- dimensional space, even when the data form high- dimensional spaces, i.e. when the input vector is high (Leão et al., 2009). Using these techniques, no target variable is required and, therefore, no distinction is made between independent and dependent variables. Thus, they are useful tools for identifying and classifying patterns immersed in a mass of data.
Figure 1 illustrates the distribution curve pattern for weight based in the Leão et al., (2009) dataset (n= 9,200 weight entries) and the 4 developmental stages used by Yamamoto (2003) (upper) and the curve pattern generated by using k-means to the data (below). The animals used by Leão et al., (2009) were living in outdoor cages at the Núcleo de Primatologia of the Federal University of Rio Grande do Norte. A database with recorded weights across the ontogenetic development of captive *Callithrix jacchus* born between 1985 and 2003 was used. The reference data used the age in days of the animals, as proposed by Yamamoto (1993), where infant class = 0-150 days; juvenile = 151 to 300 days; subadult = 301-600 and adult > 600 days. We also present in Figure 1 the application of k-means to the other classifications described in Ingram (1977), and Abbott et al. (2003) and Stevenson & Rylands (1988) using behavioral and physiological variables. Cluster techniques show that developmental stages cannot be divided based exclusively on weight information. Comparing the results with the behavioral classification that was used by the different authors, we see that clustering based only on weight generally tends to spread age classification further than conventional behavioral classification. For instance, an infantile stage based on a range of weights could be confounded with an older animal, suggesting that weight is not directly (linearly) related to age stage classification. Even though the animal's behavior stabilizes in adulthood, ceasing to gain weight between 15 and 20 months, there is still enough time (in the lifespan of the animal) to allow the division of stages according to weight, i.e. weight no longer characterizes behavioral development.

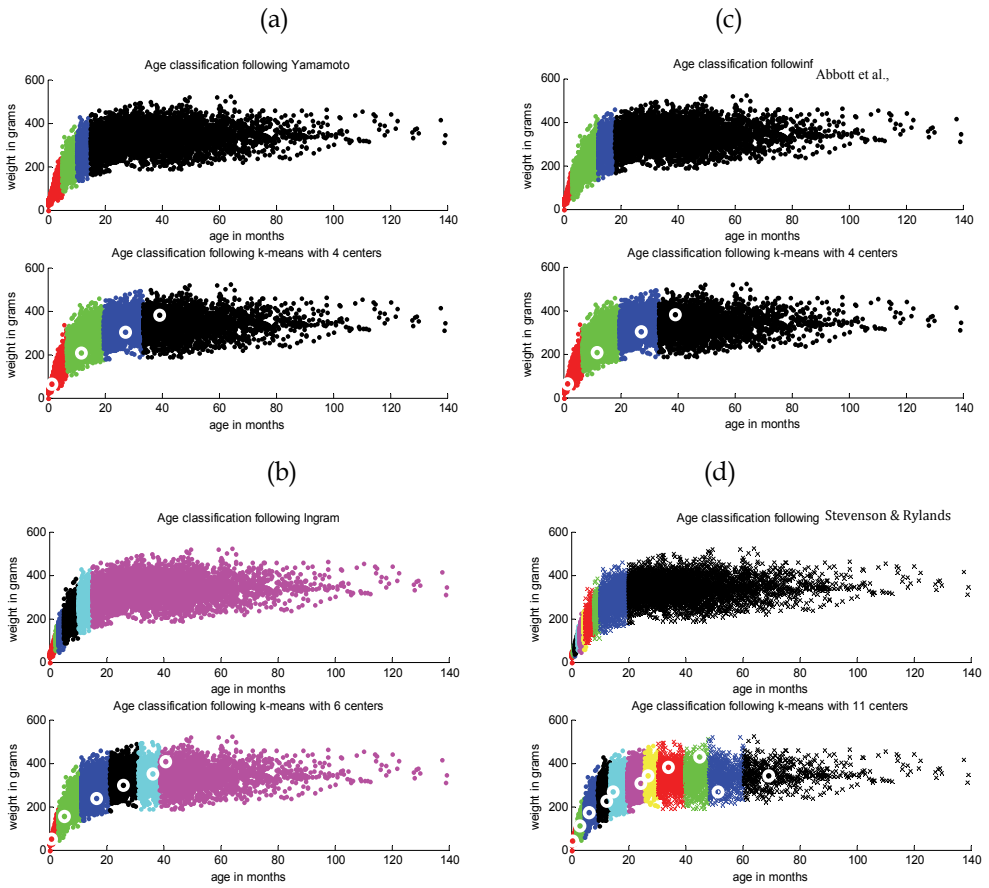


Fig. 1. Comparison plot of age classification with k-means using the stages proposed by Yamamoto, 1993 (a); Ingram, 1977 (b); Abbott et al., 2003 (c), and Stevenson & Rylands, 1988 (d). Colors indicate the infantile to adult stages. Age limit values are shown in Table 2.

Ingram and Stevenson and Rylands describe substages within the infantile stage. Using the number of infantile substages according to Ingram (n=3) and Stevenson & Rylands (n=7), characterized by behavioral and physiological data, k-means reveals that 3 different substages might be considered within the first 4 months, since statistical analysis of the 3 centroids shows significance among them. Furthermore, 2 substages were also found within the juvenile stage. Thus, according to Leão et al., (2009) using the 4 age stages proposed by Yamamoto (1993) resulted in 5 substages as follows: Infantile stage: 3 substages - infantile I (0 to 1 month) with mean weight of 40g (SD ± 11.79g); infantile II (>1 to 3 months) with mean weight of 98g (SD ± 26g) and infantile III (>3 to 4 months) with mean weight of 165g (SD ± 20g) and Juvenile stage: 2 substages – juvenile I (>5 to 7 months), with mean weight of 197.68g (SD ± 36g) and juvenile II (>7 to 10 months), with mean weight of 255.43g (SD ± 34g). These findings suggest that increased weight in early development follows behavioral progress in common marmosets. To explore this possibility we started to behaviorally characterize the infantile stage in common marmosets fitted to these 3 new infantile period substages.

Computational technique	Author/age in months
k-means	Yamamoto, 1993
0 to 11.8	0 to 5
11.8 to 27.2	5 to 10
27.2 to 39.2	10 to 15
39.2 and above	15 and abovee
k-means	Abbott et al, 2003
0 to 11.8	0 to 3
11.8 to 27.2	3 to 12
27.2 to 39.2	12 to 18
39.2 and above more	18 and above
k-means	Ingram, 1977
0 to 3.13	0 to 2
3.13 to 10.94	2 to 3
10.94 to 21.20	3 to 5
21.20 to 31.19	5 to 10
31.19 to 38.63	10 to 15
38.63 and above	15 and above
k-means	Stevenson & Rylands, 1988
0 to 1.46	0 to 0.5
1.46 to 3.82	0.5 to 0.93
3.82 to 7.93	0.93 to 1.17
7.93 to 13.63	1.17 to 2.1
13.63 to 17.79	2.1 to 2.8
17.79 to 22.61	2.8 to 4
22.61 to 29.38	4 to 5
29.38 to 38.70	5 to 8
38.70 to 48.07	8 to 10
48.07 to 60.05	10 to 20
60.05 and above	20 and above

Table 1. Age limits (in months) to classify stages according to different classifications available in the literature and k-means clustering.

Feeding behavior during the 3 infantile substages of captive common marmosets

We monitored 9 common marmosets from birth to the young adult stage (16 months) to characterize the behavioral correlates of the classification proposed by Leão et al., (2009). We followed 9 captive common marmosets (5 females and 4 males) from the first to the 16th week of life. Data sampling occurred twice a week for 30 min, in the morning and afternoon, totaling 4 sessions per week in the infantile stage (0-4 months). According to this classification, in which the infant stage was divided into 3 substages, weaning takes place at the end of the infantile II stage (twelve weeks) and the first bouts of solid food ingestion for marmosets were recorded in the infantile I substage (Table 2). We also found significant

statistical differences between food episode frequency across the three stages (ANOVA: infantile I x infantile II $p < 0.001$; infantile II x infantile III, $p = 0.022$; infantile II x infantile III, $p < 0.001$) (see Figure 2), suggesting that the infantile stage must be investigated as, at least, a tripartite phase, since behavioral changes occur rapidly and the onset of most individual and social behaviors occurs at this stage. These results suggest that this classification provides more detailed information about the surge of behavioral development, providing new information for the management and care of these small primates in early stages of development.

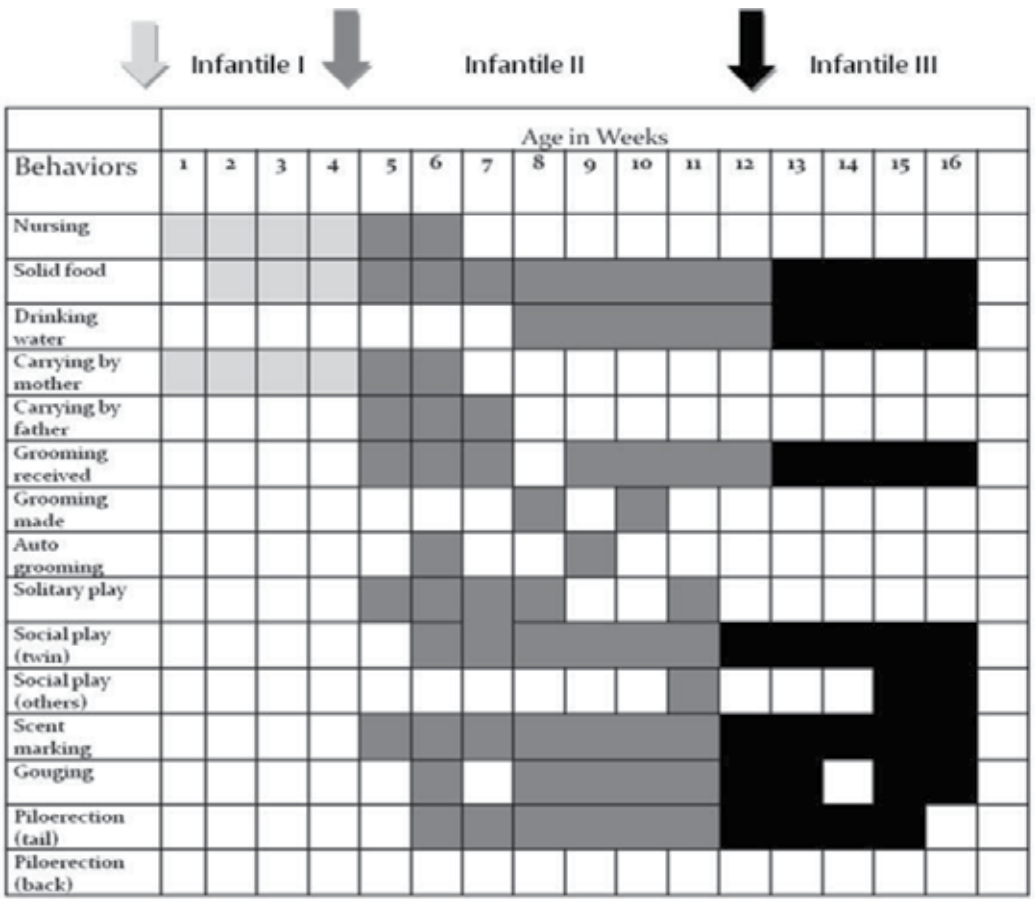


Table 2. Developmental stages in common marmosets. Different colours represent the three successive substages (I, II and III) of the infancy phase.

4. Data analysis of the ontogenetic development of common marmosets using principal component analysis

The main goal in data analysis is to represent the information contained in the data. In most cases, the data consist of a table in which the rows are measurements of some process (the information source) and the columns are the attributes that are being measured (information container).

To perform this rotation, one must find the proper rotation angles (in several dimensions). PCA finds a new set of coordinates that correspond to the eigenvectors of the covariance matrix of the data, which can be computed as

$$\mathbf{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$$

Where $\mathbf{x}_i$ is the vector representing the features in the i'th column (i'th measure) and $\boldsymbol{\mu}$ is the mean of the entire table, computed as

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$$

In both cases, N is the number of points (or measures). The superscript T stands for matrix transpose. The entire table can be represented as matrix $\mathbf{X}$ as follows:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix}$$

PCA can now be performed by projecting (or simply multiplying) the whole dataset by the transformation $\mathbf{U}$ given by the eigenvectors of $\mathbf{\Sigma}$ (each eigenvector of $\mathbf{\Sigma}$ is a column of $\mathbf{U}$).

$$\mathbf{Y} = \mathbf{XU}$$

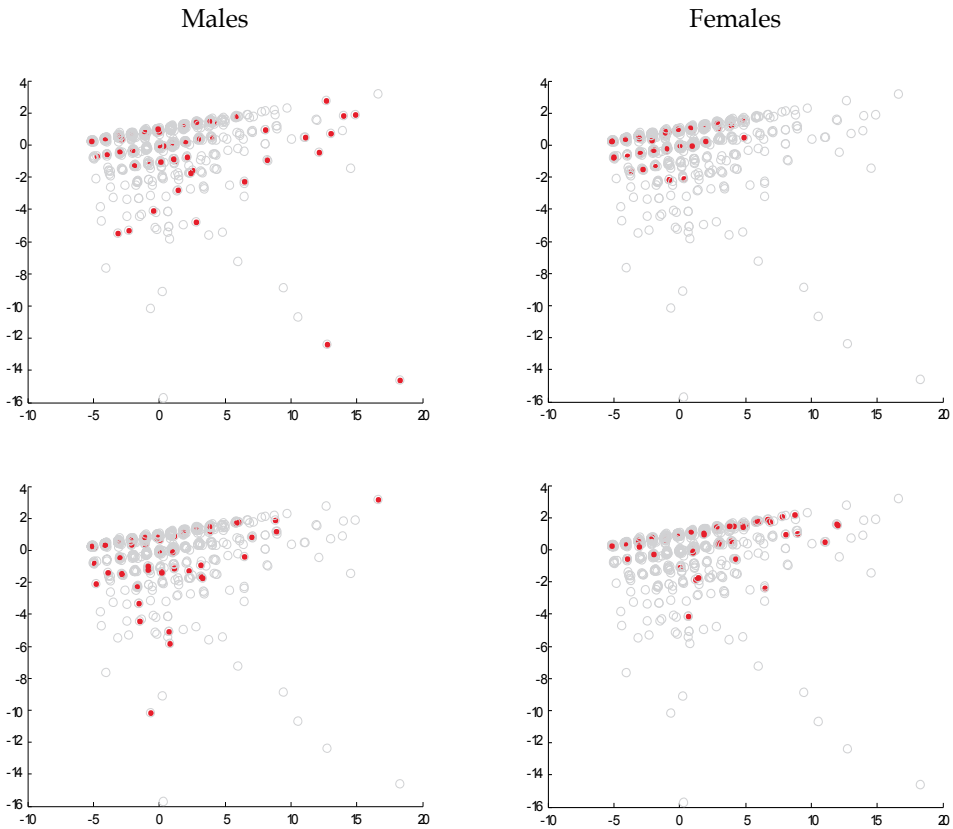
Each column in the new table $\mathbf{Y}$ corresponds to the new feature. Since the transformation matrix $\mathbf{U}$ is formed by the eigenvectors of $\mathbf{\Sigma}$, each column corresponds to an eigenvalue. Matrix $\mathbf{\Sigma}$ is symmetric (by formation), implying that the eigenvalues are all real and positive values. It turns out that the size of the eigenvalue corresponds to the variance of the respective column in the new dataset. Therefore, the larger the eigenvalue the larger the amount of information contained in the new column. The columns where the eigenvalues are small can now be discarded.

Sex signature using behavioral development data of common marmosets using PCA applied to the dataset

To perform PCA analysis the data were collected from 9 common marmosets (n= 5 females) from birth to young adulthood (16 months) as previously described. All the frequencies of social and solitary behaviors measured, classified as feeding (solid ingestion of food), affiliation (allogrooming), agonistic (scent-marking and piloerection) and individual (autogrooming) behaviors were typed into a template and Principal Component Analysis (PCA) was applied. One of the goals of applying PCA to this dataset is to perform feature extraction. We performed PCA in the dataset without the "sex" feature (removing the sex column) using only behavioral variables. Thus, only the first two columns of the

transformed data (two largest eigenvalues) were used. We then plotted the new features for each individual. Figure 2 illustrates that behavior features of common marmoset males (plots on the left) after PCA are more dispersing than females (plots on the right), suggesting that the variance in behavior in the original data is greater in males. Thus, analysis showed that computational tools exhibit behavioral differences between sexes during development.

For instance, males are more involved with territorial defense than females and adult males are the main caregivers, providing both transportation and sharing food with the young animals (Yamamoto et al., 2010). Females show more competitive behavior (agonistic and physiological inhibition) to other same- sex conspecifics (Abbott, 1984; Abbott et al., 1993; Araujo & Yamamoto, 1993; Alencar et al., 2006). Basal cortisol is higher in adult females than in males (Johnson et al., 1996 Raminelli et al., 2001) and a positive correlation is recorded between cortisol and both agonistic and anxiety behaviors when nonrelated females are living in same-sex dyads (Galvão-Coelho et al, 2008). With respect to the reproductive strategies of both sexes, it is considered that females compete and males cooperate to achieve reproductive success. As suggested by Yamamoto et al., (2010) the success of the common marmosets' social system seems to depend on an equilibrium between both strategies.



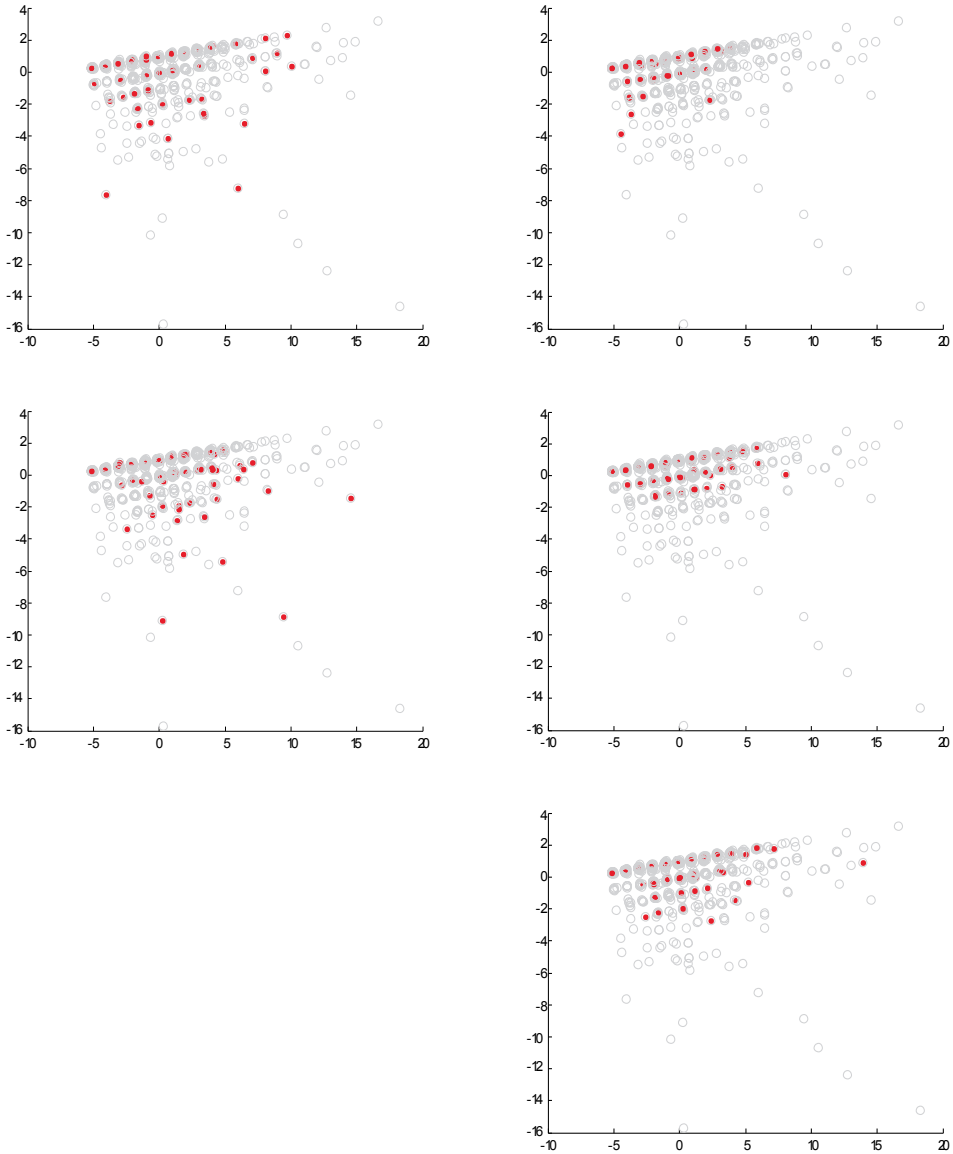


Fig. 2. Plots for the first two features extracted with PCA from behavioral data of common marmosets (4 males and 5 females). First column shows plots of male subjects and the second column shows the same features for females. The gray circles are the features for the entire dataset and the red dots are the features for each animal (each one in a different plot).

Although the use of PCA on behavioral data of common marmosets during development did not identify which type of behavior differs between males and females using sociobiological and ecological approaches, it is known that common marmoset males and

females diverge in their strategies, as previously demonstrated. Further studies might reveal other behavioral differences between males and females in post hoc analysis.

5. Future directions

Given that the behavioral repertoire in developing common marmosets was recorded across the juvenile, subadult and adult phases, it would be very interesting to use computational tools to describe the subtle changes occurring in juvenile I and juvenile II substages. For instance, preliminary data pointed out that solitary and both types of social play - between the twins and between the focal animal and the older siblings - are very intense in the juvenile II substage. The association between behavioral measurement and hormone cortisol dosage and sexual steroids, mainly during the juvenile stage, will provide more detailed information on the endocrine modulation of the behaviors, since the puberty mechanisms are triggered in primates at this time.

Another very interesting point to be analyzed using cluster techniques might be the behavioral data obtained by continuous recording during the entire light phase and circadian rhymicity could be demonstrated for the behaviors we are already measured and by adding other behavioral categories such as moving and stationary behaviors.

Moreover, characterizing the growing stages of common marmosets using motor performance and sensory skill tests at different ages will allow developmental researchers to be aware of opportunities for studying critical periods for the acquisition and maturation of motor and sensory competencies.

6. References

Abbott, D.H. (1984). Behavioural and physiological suppression of fertility in subordinate marmoset monkeys. *American Journal of Primatolology, 6*: 169-186.

Abbott, D.H. (1987). Behaviourally mediated suppression of reproduction in female primates. *Journal of Zoology, London, 213*: 455-470.

Abbott, D.H; Barrett, J. & George, L.M. (1993). Comparative aspects of social suppression of reproduction in female marmosets and tamarins. In A.B Rylands (ed.), *Marmosets and Tamarins: Systematics, Behaviour and Ecology*. Oxford University Press, pp. 152-163.

Abbott, D. H., Barnett, D. K., Colman, R. J., Yamamoto, M. E. & Schultz-Darken, N. (2003). Aspects of common marmoset basic biology and life history important for biomedical research. *Comparative Medicine, 53*: 339 - 350.

Albuquerque A.C.S.R., Sousa M.B.C., Santos H.M. & Ziegler T.E. (2001). Behavioral and hormonal analysis of social relationship between oldest (reproductive and non-reproductive) females in a wild monogamous group of common marmoset, *Callithrix jacchus. International Journal of Primatology*, 22:631-645.

Alencar, A.I., Sousa, M.B.C., Abbott, D.H. & Yamamoto, M.E. (2006). Contested dominance modifies the anovulatory consequences of social subordination of female marmosets. *Brazilian Journal of Medical and Biological Research*, 39: 647-658.

Araujo, A. & Yamamoto, M.E. (1993). Reação a intrusos da mesma espécie em *Callithrix jacchus*: influência do status social. In: M.E. Yamamoto & M.B.C. Sousa (Eds.) *A*

Primatologia no Brasil, Vol 4. Natal: Sociedade Brasileira de Primatologia/Editora Universitaria, pp 15-34.

Araújo, A. (1996). Influence des facteurs écologiques, comportementaux et démografhiques sur la dispersion de *Callithrix jacchus*. Doctoral thesis. Paris: Université Paris-Nord (Paris XIII). 234 p.

Arruda, M.F., Araujo, A., Sousa, M.B.C., Albuquerque, F.S. &Yamamoto, M.E. (2005) Two breeding females within free-living groups may not always indicate polygyny: Alternative subordinate female strategies in common marmosets (*Callithrix jacchus*). *Folia Primatologica*, 76: 10-20.

Duda, R. Hart, P. and Stork, D. (2000). *Pattern Classification*, Second Edition, Wiley.

Enquist, M. & Ghirlanda, G. (2005) Neural networks and animal behavior. Princeton University Press. Pp. 1-29.

Galvão-Coelho, N.L., Silva, H.P.A., Leão, A.C. & Sousa, M.B.C. (2008). Common marmosets (*Callithrix jacchus*) as a potential animal model for studying psychological disorders associated with high and low responsiveness of the hypothalamic-pituitary-adrenal axis. *Reviews in the Neuroscience*, *19*: 187-201.

Harris-Warrick, R. M. & Marder, E. (1991) Modulation of neural networks for behavior. *Annual Reviews in Neuroscience,* 14: 39-57.

Hubel, D.H. & Wiesel, T.N. (1962). Receptive fields, binocular interaction and functional architecture in cats visual cortex. *Journal of Physiology, London*, 160: 106-154.

Ingram, J.C. Interactions between parents and infants, and the development of independence in the common marmosets (*Callithrix jacchus*). *Animal Behaviour*, 25:811-827.

Johnson, E.O., Kamilaris, T.C., Carter, C.S., Calogero, A.E., Gold, P.W. & Chrousos, G.P. (1996). The behavioral consequences of psychogenic stress in a small, social primate (*Callithrix jacchus jacchus*). *Biology Psychiatry*, 40:317-337.

Kandel, E.R. & Pittenger, C. (1999). The past, the future and the biology of memory storage. *Philosophical Transactions of The Royal Society of London Series B-Biological Sciences*, 2027-2052.

Koenig, A. & Rothe, H. (1991). Social relationships and individual contribution to cooperative behaviour in captive common marmosets (*Callithrix jacchus*). *Primates, 32*: 183-195.

Kohonen T. (1982). Self-organised formation of topographically correct feature maps. *Biological Cybernetics*. *43*: 59–69.

Lazaro-Perea, C. 2001. Intergroup interactions in wild common marmosets, *Callithrix jacchus*: territorial defense and assessment of neighbours. *Animal Behaviour*, 62, 11-21.

Lazaro-Perea, C., Castro, C. S. S., Harrison, R., Araújo, A., Arruda, M.F. & Snowdon, C. T. (2000). Behavioral and demographic changes following the loss of the breeding female in cooperatively breeding marmoset. *Behavioral Ecology and Sociobiology, 48*: 137-146.

Kendrick, K.M. & Dixson, A.F. (1986). Neurohormonal control of sexual-behavior in the female common marmoset (*Callithrix jacchus*). *Applied Animal Behaviour Science*, 15 189-190.

Leão, A. C., Doria Neto, A. D. & Sousa, M. B. C. (2009). New developmental stages for common marmosets (Callithrix jacchus) using mass and age variables obtained by K-means algorithm and self-organizing maps (SOM). *Computers in Biology and Medicine*, 853-859.

Matheus, L. J. (2009). Intragroup behavioral variation in white-fronted capuchin monkeys (*Cebus albifrons*): mixed evidence for social learning inferred from new and established analytical methods. *Behaviour,* 146: 295-324.

Pozzi, L., Gamba, M. & Giacoma, C. (2010). The use of artificial neural networks to classify primate vocalizations: a pilot study on black lemurs. *American Journal of Primatology*, 72: 337-348.

Raminelli, J. L. F., Sousa, M. B. C, Cunha, M. S. & Barbosa, M. F. V. (2001). Morning and Afternoon Patterns of Fecal Cortisol Excretion among Reproductive and Non-Reproductive Male and Female Common Marmosets, *Callithrix jacchus. Biological Rhythm Research,* 32: 159-167.

Reeke, G.N. and Sporns, O. (1993). Behaviorally based modeling and computational approaches to neuroscience. *Annual Reviews in Neuroscience*, 16: 597-623.

Rylands, A. B. (1996). Habitat and evolution of social and reproductive behavior in Callitrichidae. *American Journal of Primatology*, *38:* 5-18.

Saltzman, W., Prudom, S.L., Schultz-Darken, N.J., Wittwer, D.J. & Abbott, D.H. (2004). Social suppression of cortisol in female marmoset monkeys: Role of circulating ACTH levels and glucocorticoid negative feedback. *Psychoneuroendocrinology, 29*: 141–161.

Sousa M.B.C, Albuquerque A.C.S.R., Albuquerque F.S., Araújo A., Yamamoto M.E. & Arruda M.F. (2005). Behavioral strategies and hormonal profile of dominant and subordinate common marmoset (*Callithrix jacchus*) females in wild monogamous groups. *American Journal of Primatology*, 67: 37-50.

Sousa, M.B.C., Albuquerque, A.C.S., Yamamoto, M.E., Araújo, A. & Arruda, M.F. (2009). Emigration as a reproductive strategy of the common marmoset (*Callithrix jacchus*). In S. M. Ford, L. M. Porter & L. C. Davis (Eds), *The Smallest Anthropoids: The Marmoset/Callimico Radiation*. New York: Springer, pp 167–182.

Stevenson, M. F. & Rylands, A. B. (1988). The marmosets, genus *Callithrix*. *In*: Mittermeier R. A., Rylands, A. B., Coimbra-Filho A. e Fonseca G. A. B. (eds.). *Ecology and Behavior of Neotropical Primates*, vol. 2. Contagem: Littera Maciel. p. 131-222.

Tin JA, D'Souza A, Yamamoto K., Yoshioka T., Hoffman D., Kakei S., Sergio L., Kalaska J., Kawato M., Strick P. & Schaal S. (2008). Variational Bayesian least squarest: An application to brain-machine interface data. *Neural Networks*, 21: 1112-1131.

Yamamoto, M.E. (1993). From dependence to sexual maturity: The behavioural ontogeny in Callithrichidae. In: Rylands, A.B. (ed.) Marmosets and Tamarins. Systematics, Behaviour and Ecology. Oxford University Press, Oxford, pp.235-254.

Yamamoto, M. E., Arruda M.F., Alencar A.I., Sousa M.B.C. & Araújo, A. (2009). Mating systems and female–female competition in the common marmosets, *Callithrix jacchus*. In S. M. Ford, L. M. Porter & L. C. Davis (Eds), *The Smallest Anthropoids: The Marmoset/Callimico Radiation*. New York: Springer, pp 119–133.

Yamamoto. M.E., Araujo. A., Sousa. M.B.C. & Arruda, M.F. (2010). Social organization in *Callithrix jacchus*: cooperation and competition. In: Regina Macedo Editor: Advances in the Study of Behavior, vol. 42, Burlington: Academics Press, pp. 259-273.

20

Clustering Genes, Tissues, Cells and Bioactive Chemicals by Sphere SOM

Yuh Sugii[1], Takayuki Kudoh[1], Takayuki Otani[1], Masashi Ikeda[1],
Heizo Tokutaka[2] and Masaharu Seno[1]
[1]Okayama University,
[2]SOM JAPAN Inc.
Japan

1. Introduction

The technology of high throughput screening is nowadays widely available in life science especially in the fields of molecular diagnosis and drug discovery. This is due to the establishment of microarray procedure, which deals with huge number of genes at one time. Cluster analysis is usually performed on the results of DNA microarray experiments. However, the routine procedure of data mining dealing with the huge number of signal information obtained from microarray is not fixed yet. We can find various way of clustering in hierarchical and non-hierarchical methods, which are applied for the analyses. The most popular ones appear non-hierarchical clustering such as k-means (MacQueen, 1965), partitioning around medoids (Kaufman and Rousseeuw, 1990) and cluster affinity search technique (Ben-Dor et al., 1999).

We have employed spherical self organizing map (sSOM), which is also a non-hierarchical clustering, to cluster genes by gene expression profiles of cells and tissues (Tuoya et al., 2008). Analyzing various types of carcinoma cells and normal tissues, we could find interesting cell surface molecules, which should serve as the molecular markers. This procedure, which we are demonstrating, is rather new to the data analyses of gene clustering from the gene expression profiles obtained from DNA microarray technique. Flexible arrangement of the data obtained allows us to cluster cells and tissues as well as genes to find definitely fantastic direction of further advancement of study.

Furthermore, we applied sSOM to classify bioactive chemical compounds by their mechanism of action (MOA), which should enable us virtual screening in silico (Reddy et al., 2007). In recent years, many intriguing methods for virtual screening have been developed in this field (Gasteiger et al., 2003; Melville et al., 2009). Especially, ligand-based method is suitable for selecting drug candidates from enormous compounds library because it simply requires computational resources, which are less expensive. Although SOM has partly been used as a tool of ligand-based methods to classify compounds by their properties in chemoinformatics, spherical SOM has not been used in chemoinformatics to the best of our knowledge (Brüstle et al., 2002; Schneider & Nettekoven, 2003; Schneider & Schneider, 2003; Wang et al., 2005; Kaiser et al., 2007; Renner et al., 2007; Li & Gramatica, 2010). We propose here the extended application of sSOM to classify bioactive compounds by their MOA together with their structural information. In the future this procedure should

be extremely helpful in the field of drug discovery as well as those of molecular biology and oncology.
This chapter is dedicated to introduce our concept of the application of sSOM procedure.

2. Materials, methods and tools

2.1 Cell lines and cell culture

Human breast cancer derived cell lines Hs-578T, MCF-7, MDA-MB-134, MDA-MB-231, SK-BR-3, T-47D and ZR-75-1 were obtained from American Type Culture Collection (ATCC, VA). Hs-578T cells were cultured in DMEM containing 10 % fetal bovine serum (FBS), 10 μg/mL insulin and 2 mM L-glutamine. MCF-7 cells were cultured in MEM containing 10 % FBS, 10 μg/mL insulin and 2 mM L-glutamine. MDA-MB-134 cells were cultured in Leibovitz-15 containing 10 % FBS, 2 mM L-glutamine buffered with 10 mM HEPES. MDA-MB-231 cells were cultured in DMEM containing 10 % FBS and 2 mM L-glutamine. SK-BR-3 cells were cultured in RPMI 1640 containing 20 % FBS and 2 mM L-glutamine. T-47D cells cultured in RPMI 1640 containing 10 % FBS, 2 mM L-glutamine, 10 μg/mL insulin and 30 ng/mL EGF. ZR-75-1 cells were cultured in RPMI 1640 containing 10 % FBS. All cells were maintained at 37 ˚C in a humidified 5 % CO_2 atmosphere except MDA-MB-134 cells, which were maintained in 100 % air.

2.2 Preparation of total RNA and cDNA synthesis

Total RNA was extracted from the cells used in this study. Cells were harvested at a confluence of 80% for preparation using RNeasy Mini kits (Qiagen), following the manufacturer's instructions. Total RNA from human normal breast and mouse normal tissues was purchased from Stratagene (CA). RNA integrity and purity were assessed by $OD_{260/280}$ measurements and by the ratio of 28S and 18S rRNA with Experion system (BioRad Labs, VA). The total RNA was further treated with DNase and purified. The integrity of template RNA was assessed by $OD_{260/280}$ measurements. Twenty micrograms of total RNA was used to synthesize cDNA in the presence of aminoalkyl-dUTP. To monitor the efficiency of cDNA synthesis and hybridization control RNAs were added in the reaction as describe previously (Tuoya et al., 2008; Abou-Sharieha et al. 2009). Cy3-labeled cDNAs were prepared by indirect labeling method adapted from the Brown Web site (http://cmgm.stanford.edu/pbrown/protocols).

2.3 Microarray analysis

We originally proposed DNA microarray, which focused cell membrane-bound proteins to identify cell surface marker specific to the cells or tissues of interest (Tuoya et al., 2008; Abou-Sharieha et al. 2009). Two different microarrays were designed to contain 1,795 oligonucleotide probes corresponding to human genes and 1,405 corresponding to mouse genes, respectively. These genes were limited to those coding membrane-bound proteins so as to cover cell surface proteins. To avoid the effect of alternative splicing, the coding sequence for the membrane-bound region or GPI-anchor modified region was focused to design the oligonucleotide probes. The probes were conjugated on the slide glass coated with diamond-like carbon as described previously (Tuoya et al., 2005).
The Cy3-labeled cDNA synthesized above was hybridized to the cell surface marker DNA microarray in 5x SSC/0.5 % SDS solution at 55 °C for 15 h. After washing, arrays were scanned on a FLA8000 scanner (Fuji Film, Japan). Intensity for each spot of the array was

captured by GenePix® Pro5.1 image analysis software (Axon Instrument). The fluorescent intensity of each spot referred as ''relative fluorescent intensity (RFI)'', which represented the expression level of each gene. Gene expression levels were compared to one another by RFI value to identify differentially expressed genes.

2.4 Data filtering in breast cancer cell

In order to eliminate genes that did not change significantly between cancer cell lines and normal tissue, each gene was given a score S by a formula:

$$S = |N - C| - V_C$$

, where N, C and V_C denote the expression level of the gene in normal breast, the average of the expression levels of the gene in the seven cancer cell lines and the standard deviation of the gene expression level in the seven cancer cell lines, respectively. Genes were eliminated from further consideration when S<0 or S=0, since only the genes with a score greater than a threshold (i.e., zero) are deemed potentially significant (Tuoya et al., 2008).

2.5 sSOM analysis of gene expression

The expression levels of each gene were normalized among the breast cancer cell lines and normal breast tissue and among mouse normal tissues. First, the maximal RFI value of each gene was taken as 1, the minimum RFI was taken as 0 and other RFI values were linearly calculated into the values between 0 and 1. Secondly the average expression levels of each gene were calculated and each average was divided by the maximal average value. The resultant values were further multiplied to each normalized value calculated above. The normalized data were clustered and displayed by sSOM software Cluster Blossom (Ver. 1.0.2, SOM Japan Co-Ltd., http://www.somj.com/). The training of Cluster Blossom were performed 50 times. Other parameters were automatically set by the software. Then the dendrograms were drawn from the final map after training by group average method with a glyph value 1.0.

2.6 Datasets for chemicals

The dataset analyzed in this study was taken from the previous report, in which 131 compounds were classified by the self organizing map with screening data against the 60 human cancer cell lines as input vectors (van Osdol et al., 1994). All these compounds structure data were downloaded from NCI databases by using Enhanced NCI Database Browser (http://129.43.27.140/ncidb2/). The names of compounds analyzed in this study are listed in Table 2 with NSC Nos. and MOA.

2.7 Descriptors of chemicals

All downloaded structures were submitted to the chemical descriptor calculation software, CDK Descriptor Calculator GUI (ver. 1.0.5; http://rguha.net/code/java/cdkdesc.html) to calculate 283 theoretical descriptors, including molecular descriptors, bond descriptors and atom descriptors (Steinbeck et al., 2003).

2.8 Descriptor scaling and selection

All above calculated descriptors were normalized by each row that they have mean 0 and variance 1 by the function of “normalize” in the “som package" of statistical software R

(Windows Ver. 2.9.0; "Self-Organizing Map" R package Version 0.3-4, URL http://cran.r-project.org/; R Development Core Team, 2007, http://www.R-project.org/). All errors were deleted from this dataset and normalized dataset (116 compounds-by-215 descriptors matrix) was obtained.

2.9 sSOM analysis of chemicals

Clustering were performed with the software Cluster Blossom (Ver. 1.0.3, SOM Japan Co-Ltd.). The trained sSOM was developed using the dataset above mentioned as input vectors. The same training parameters of Cluster Blossom were used as described above. Similarly, the dendrogram was drawn from the trained map as described above. The accuracy of clustering *A* was calculated as following.

$$A = Nt_{Gi} / N_{Gi} \times 100$$

where Nt_{Gi} is the number of compounds correctly assigned in cluster *Gi* and N_{Gi} is the number of compounds assigned in cluster *Gi* where *i* depicts the number of cluster.

3. Results and discussion

3.1 sSOM clustering of human breast cancer cell lines

We performed DNA microarray gene expression analysis in order to screen genes commonly and specifically expressed in the seven cell lines derived from breast cancer when compared to normal breast. As the result of data filtering, 840 genes were found to suffice the criteria described in "2.4 Data filtering in breast cancer cell". The expression levels of these genes were then normalized and clustered by sSOM. The gene expression profiles were visualized on the sphere surface map and the dendrogram indicating themselves classified by the origin of the cells (Fig. 1). It is interesting to note that Hs-578T and MDA-MB-231 cells, which are derived from basal-like breast cancer known to have poor prognosis, are clustered in the same group (Ray et al., 2010). T-47D, ZR-75-1, MCF-7 and MDA-MB-134 cells, which are derived from luminal breast cancer, are well known to have good prognosis. Since SK-BR-3 cells are Her2 positive, which is an efficient target for the cancer therapy, and derived from breast cancer of medium level of prognosis. Thus, the gene expression profiles were successfully visualized by the sSOM clustering, suggesting the clusters of prognosis. From the patterns, cells derived from luminal breast cancer appear to be clustered into three groups of "close to normal", "medium" and "poorer". Namely, it might be possible to diagnose SK-BR-3 cells as "close to normal" while MDA-MB-134 as poorer than the other luminal derived cells.

In order to find genes highly expressed in all the seven cancer cell lines, sSOM was performed with an assumptive gene inserted into the dataset of the 840 genes. The assumptive gene stood for an ideal point IP, which was supposed to be expressed in all the breast cancer cell lines analyzed in this study but not in the normal breast tissue, so that the genes clustered close to IP should be potential diagnostic markers of breast cancer. In the result of sSOM clustering, IP was mapped in the red part of the pattern in all the seven cancer cell lines (Fig. 1) but blue in normal breast tissue. Since this mapped position of IP is consistent with the assumption, the genes close to IP should be selected as candidates of cancer-specific genes on the sSOM. Each spot on the surface of sSOM contains a group of clustered genes (Fig. 2). The spots mapped close to IP are shown in Fig. 2 and the candidate genes clustered in each spot are listed in Table 1. It is noteworthy that ErbB3 and ROBO2 have been nominated as potential diagnostic

markers here and some reports are found describing their relationships with breast cancers (Lemoine et al., 1992; Gasparini, 1994; Quinn et al. 1994 Travis et al., 1996; Naidu et al., 1998; Fogel et al., 1999; Holbro et al., 2003; Barnes et al., 2005; Schabath et al., 2006; Shiau et al. 2008). MUC1 is also known as a diagnostic marker in various cancers including breast cancers (Singh et al., 2008). Considering the results that contain these potential candidates, the other genes listed in Table 1 could be a potential candidate for the diagnostic marker of breast cancers still unknown.

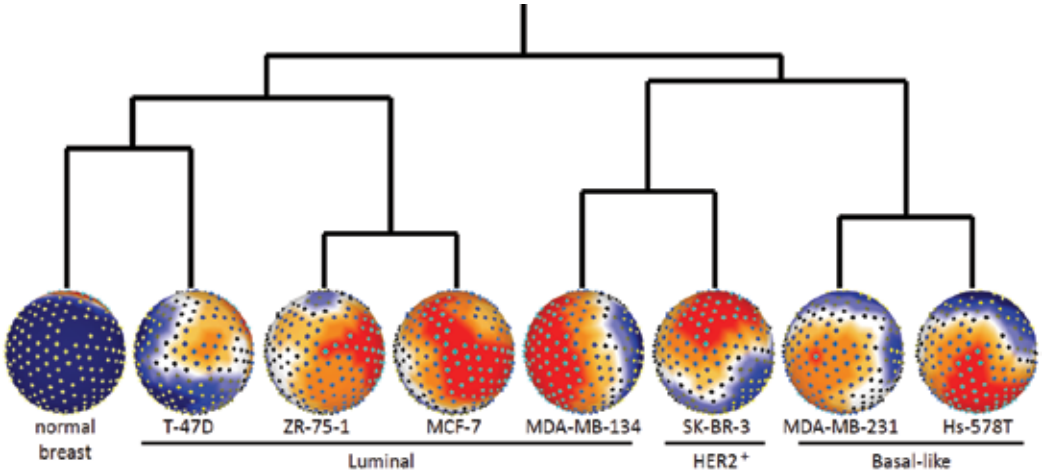


Fig. 1. The gene expression profiles analyzed by sSOM for cancer derived cell lines and normal breast. The normalized data set was clustered and visualized by Cluster Blossom. Each position of genes is fixed on the global surface. The colors indicate the expression level for each gene. Red, high; yellow, slightly high; white, median; light blue, slightly low; deep blue, low. See text for the names of cell lines and diagnostic levels. The alignment of cells is the result of sSOM clustering, which was drawn by dendrogram

Gene No.	GenBank Accession No.	Gene Name
1586	NM_032038	spinster-like protein
1423	NM_016372	seven transmembrane domain orphan receptor
1784	AH006947	vitelliform macular dystrophy protein 2
1777	M29366	v-erb-b2 erythroblastic leukemia viral oncogene homolog 3 (ErbB3)
1682	NM_012471	transient receptor potential cation channel, subfamily C, member 5
734	NM_002099	glycophorin A (includes MN blood group)
1399	AF040991	roundabout, axon guidance receptor, homolog 2 (ROBO2)
163	NM_001188	BCL2-antagonist/killer 1
247	NM_001218	carbonic anhydrase XII
1699	NM_003271	transmembrane 4 superfamily member 7
241	NM_022131	calsyntenin 2
1015	NM_002456	mucin 1, transmembrane (MUC1)

Table 1. Candidate genes for the potential diagnostic marker for breast cancer as picked up from genes commonly expressed in all the cancer derived cells studied in this paper

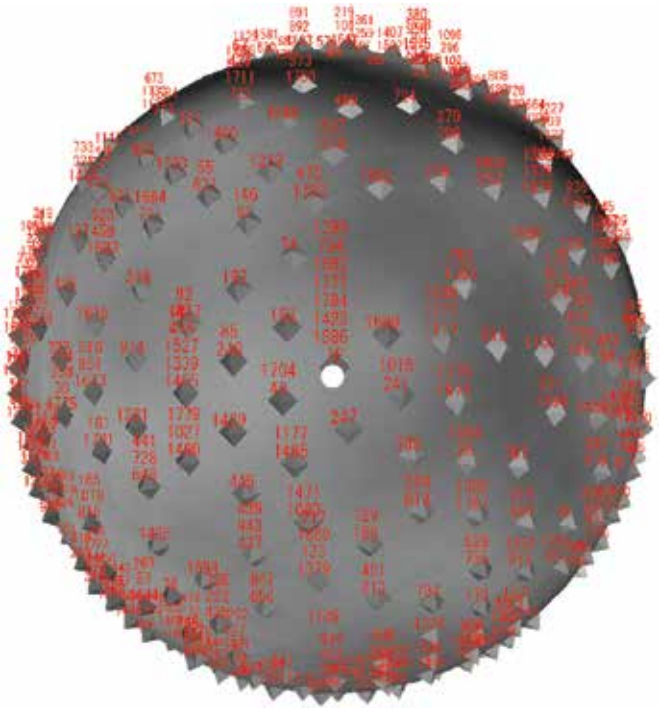


Fig. 2. The locations of ideal point IP (white spot) clustered by sSOM. This global surface is the same with those in Fig. 1 without colors. Each gray spot on the surface of sSOM contains a group of clustered genes, which were depicted with the gene numbers in red. The genes clustered close to IP are summarized in Table 1

3.2 sSOM clustering of mouse normal tissues

In this section, gene expression profiles of normal tissues in mouse were clustered by sSOM. Since the breast cancer cell lines were successfully clustered, we expected normal tissues should be also clustered with the features of each tissue. Clustering of brain, colon, heart, kidney, liver, lung, muscle, small intestine, spleen, stomach, testis and thymus was performed and the resultant gene expression profiles were aligned on the anatomical sketch of mouse body (Fig. 3). The relationship between each tissue was shown in a global map obtained by sSOM (Fig. 4). In this map, each distance between the nodes was not adjusted to a sphere surface (glyph =0) but reflected the distance when calculated by SOM (glyph =1.0) resulting in a meteoritic form of map.

The alignment of gene expression profiles around the body sketch reveals some similarities between the tissues. The similarity of the profiles between colon and intestine appears consistent. The similarity of profiles between spleen and thymus also sounds reasonable because of the deep relationship of these tissues with immunological system. The similarity is also found in heart, liver and lung. Although it is difficult to explain their close relationship from the embryonic development of tissues in mouse, it might be important to try to make viewpoints shared in these three tissues but not in other tissues as suggested by the sSOM clustering. Further application of sSOM on the gene expression profiles comparing with normal tissues and diseased tissues would lead to a challenging opportunity to find novel diagnostic markers in the future.

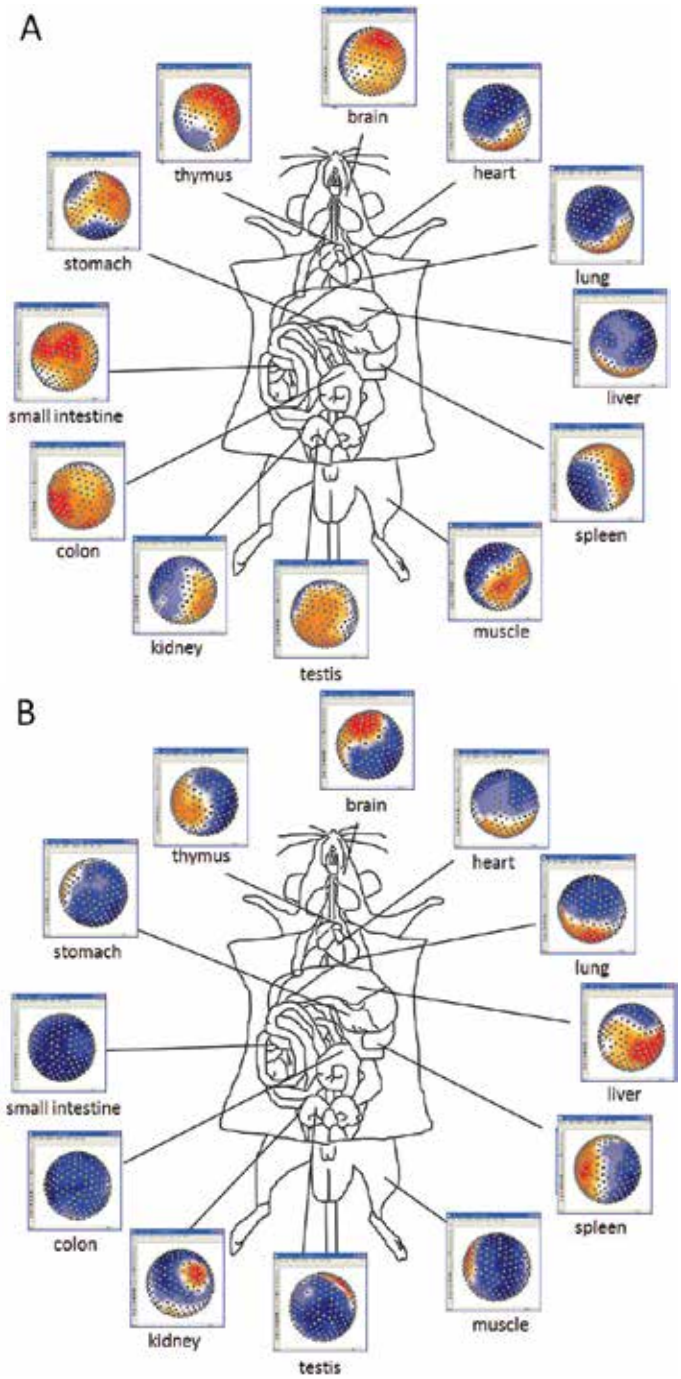


Fig. 3. Gene expression profiles of mouse normal tissues clustered by sSOM. The normalized data set was clustered and visualized by Cluster Blossom. Each position of genes is fixed on the global surface. See Fig. 1 for the colors indicating the expression level for each gene. Views of clustered global map from front side (A) and back side (B)

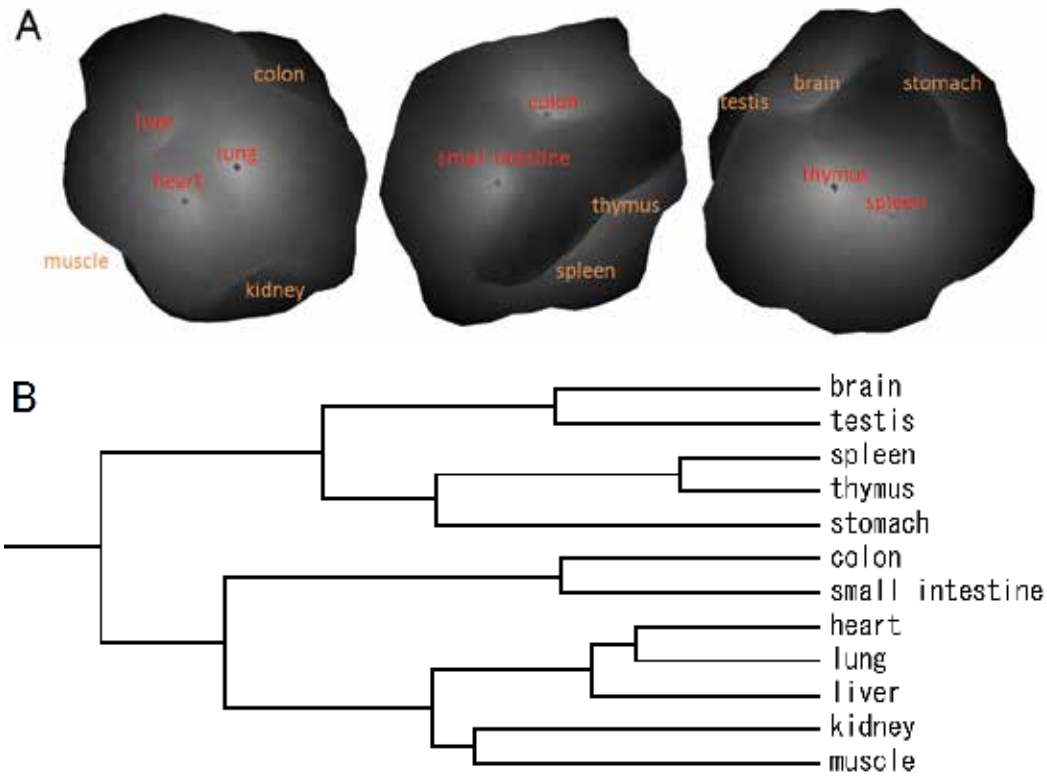


Fig. 4. Meteoritic map clustering normal mouse tissues (A) and dendrogram of clustered normal mouse tissues (B). Each distance between the nodes were calculated with glyph value 1.0. (A) Views from three different sides of the map. Tissues depicted in red letters are on the front side and those in orange letters are on the other side. The farther the relationship between the tissues, the darker the shadow is. (B) Dendrogram was calculated from the results of clustering by sSOM

3.3 sSOM clustering of bioactive chemicals

The bioactive compounds previously screened for anti-cancer reagents were evaluated for clustering in this study. The compounds were clustered by sSOM. The dendrogram was drawn based on the trained map by group average method to obtain 9 clusters of compounds, which were colored by their clusters on the surface of global map (Fig. 5). The compounds in the dataset are summarized in Table 2 with their assigned MOA and the cluster groups.

Table 3 shows the clustering results of compounds. The accuracy of clustering was overall 86.2%, ranging from 60 to 100% in each cluster. The alkylating agents, AC, A7, and AI, are misclassified relatively in higher frequency than other agents. It is interesting to note the anti-DNA agents, DI, DP, and DR, and the inhibitors of nucleotide synthesis, RI, RO, and R, are clustered into the same group. This might be the result due to the character of these agents associating with the enzymes associated with nucleotide metabolisms.

In this study, 16 compounds (ID 5, 17, 29, 33, 44, 46, 52, 64, 81, 86, 88, 92, 93, 96, 99) were misclassified. These results suggest that they might have another activity other than those experimentally defined because small organic compounds frequently exhibit

polypharmacology. In fact, trimetrexate (ID 88) and DUP785 (ID 96) might have topoisomerase inhibiting activity because both of them have resemble planar heteroaromatic ring, which is the feature of topoisomerase inhibitor. Additionally, mitzolamide (ID 89) has also heteroaromatic ring implying DNA interacting ability. The chemical structures of these three compounds are shown in Fig. 6. Currently, exploring new targets and activity of already approved drugs is fascinating strategy to develop novel therapeutic drugs with less risks of the clinical trial (Keiser et al., 2009). Although further investigation is needed, sSOM would be a comprehensive and useful tool to classify the compounds and to find novel activities in themselves.

ID	NSC No.	Drug Name	MOA	cluster
1	NSC740	Methotrexate	RF	G1
2	NSC750	Busulfan	A7	G6
3	NSC752	Thioguanine	DI	G7
4	NSC755	Thiopurine	DI	G7
5	NSC757	Colchicine	TU	G9
6	NSC762	Mechlorethamine	A7	G6
7	NSC1895	Guanazole	DR	G7
8	NSC3088	Chlorambucil	A7	G6
9	NSC6396	Thiotepa	A7	G7
10	NSC8806	Melphalan	A7	G6
11	NSC9706	Triethylenemelamine	A7	G7
12	NSC19893	Fluprouracil	R	G7
13	NSC25154	Pipobroman	A7	G6
14	NSC26980	Mitomycin	A2	G7
15	NSC27640	Floxuridine	DP	G7
16	NSC32065	Hydroxyurea	DR	G7
17	NSC33410	Colchicine derivative	TU	G9
18	NSC34462	Uracil mustard	A7	G6
19	NSC49842	Vinblastine sulfate	TU	G3
20	NSC51143	Pyrazoloimidazole	DR	G7
21	NSC56410	Porfiromycin	A2	G7
22	NSC63878	Cytarabine	DP	G7
23	NSC67574	Vincristine sulfate	TU	G3
24	NSC71261	beta-2'-Deoxythioguanosine	DI	G7
25	NSC71851	alpha-2'-Deoxythioguanosine	DI	G7
26	NSC73754	Fluorodopan	A7	G6
27	NSC79037	Lomustine	AC	G6
28	NSC82151	Daunorubicin	T2	G2

29	NSC83265	Trityl cysteine	TU	G9
30	NSC94600	Camptothecin	T1	G9
31	NSC95382	Camptothecin derivative	T1	G9
32	NSC95441	Semustine	AC	G6
33	NSC95466	PCNU	AC	G7
34	NSC95678	3-Hydroxypicolinealdehyde thiosemicarbazone	DR	G7
35	NSC100880	Camptothecin derivative	T1	G9
36	NSC102627	Yoshi-864	A7	G6
37	NSC102816	Azacytidine	RO	G7
38	NSC107124	Camptothecin derivative	T1	G9
39	NSC107392	5-Hydroxypicolinaldehyde thiosemicarbazone	DR	G7
40	NSC118994	Inosine glycodialdehyde	DR	G7
41	NSC122819	Teniposide	T2	G2
42	NSC123127	Doxorubicin	T2	G2
43	NSC125973	Paclitaxel derivative	TU	G3
44	NSC126771	Dichloroallyl lawsone	RO	G6
45	NSC127716	5-Aza-2'-deoxycytidine	DI	G7
46	NSC132313	Dianhydrogalactitol	A7	G7
47	NSC132483	Aminopterin	RF	G1
48	NSC134033	Aminopterine derivative	RF	G1
49	NSC135758	piperazinedione	A7	G6
50	NSC139105	Baker's soluble antifolate	RF	G5
51	NSC141540	Etoposide	T2	G2
52	NSC142982	Hycanthone	AI	G9
53	NSC143095	Pyrazofurin	RO	G7
54	NSC145668	Cyclocytidine	DP	G7
55	NSC14895	Ftorafur	R	G7
56	NSC153353	L-Alanosine	RO	G7
57	NSC153858	Maytansine	TU	G5
58	NSC163501	Acivicin	RI	G7
59	NSC164011	Zorubicin	T2	G2
60	NSC167780	Asaley	A7	G5
61	NSC172112	Spiromustine	A7	G6
62	NSC174121	Methotrexate derivative	RF	G1
63	NSC176323	Camptothecin derivative	T1	G9
64	NSC178248	Chlorozotocin	AC	G7

65	NSC182986	Diaziridinylbenzoquinone	A7	G7
66	NSC184692	Aminopterin derivative	RF	G1
67	NSC224131	N-(phosphonoacetyl-L-aspartic acid, tetrasodium salt)	RO	G7
68	NSC249910	Camptothecin derivative	T1	G9
69	NSC249992	Amsacrine	T2	G9
70	NSC264880	5,6-Dihydro-5-azacytidine	RO	G7
71	NSC267469	Deoxydoxorubicin	T2	G2
72	NSC268242	N,N-Dibenzyl daunomycin	T2	G2
73	NSC269148	Menogaril	T2	G2
74	NSC295500	Camptothecin derivative	T1	G9
75	NSC295501	Camptothecin derivative	T1	G9
76	NSC296934	Teroxirone	A7	G7
77	NSC301739	Mitoxantrone	T2	G2
78	NSC303812	Aphidicolin glycinate	DP	G4
79	NSC308847	Amonafide	T2	G9
80	NSC329680	Hepsulfam	A7	G6
81	NSC330500	Geldanamycin	DP	G9
82	NSC332598	Rhizoxin	TU	G5
83	NSC337766	Bisantrene	T2	G9
84	NSC338947	Clomesone	AC	G6
85	NSC344007	Piperazine alkylator	A7	G6
86	NSC348948	Cyclodisone	AC	G7
87	NSC349174	Oxanthrazole	T2	G2
88	NSC352122	Trimetrexate	RF	G9
89	NSC353451	Mitozolamide	AC	G7
90	NSC354646	Morpholino adriamycin	T2	G2
91	NSC355644	Anthrapyrazole derivative	T2	G2
92	NSC357704	Cyanomorpholinodoxorubicin	AI	G2
93	NSC361792	Thiocolchicine	TU	G9
94	NSC364830	Camptothecin derivative	T1	G9
95	NSC366140	Pyrazoloacridine	T2	G9
96	NSC368390	DUP785 (brequinar)	RO	G9
97	NSC374028	Camptothecin derivative	T1	G9
98	NSC376128	Dolastatin 10	TU	G5
99	NSC406042	Allocolchicine	TU	G9
100	NSC409962	Carmustine	AC	G6

101	NSC603071	Camptothecin derivative	T1	G9
102	NSC606172	Camptothecin derivative	T1	G9
103	NSC606173	Camptothecin derivative	T1	G9
104	NSC606497	Camptothecin derivative	T1	G9
105	NSC606499	Camptothecin derivative	T1	G9
106	NSC606985	Camptothecin derivative	T1	G9
107	NSC608832	Paclitaxel derivative	TU	G3
108	NSC610456	Camptothecin derivative	T1	G9
109	NSC610457	Camptothecin derivative	T1	G9
110	NSC610458	Camptothecin derivative	T1	G9
111	NSC610459	Camptothecin derivative	T1	G9
112	NSC618939	Camptothecin derivative	T1	G9
113	NSC623017	an. Antifol II	RF	G1
114	NSC629971	Camptothecin derivative	T1	G9
115	NSC633713	an. Antifol II	RF	G1
116	NSC643833	Camptothecin derivative	T1	G9

Table 2. Compounds in the dataset and the result of clustering. Abbreviations in MOA are as following. DNA alkylating agents: A2, alkylating at N-2 position of guanine; AC, alkyl transferase-dependent cross-linkers; A7, alkylating at N-7 position of guanine; AI, DNA intercalators. Anti-DNA agents: DI, incorporated; DP, polymerase inhibitors; DR, ribonuclease reductase inhibitors. Nucleotide synthesis inhibitors: RF, antifolates; RI, irreversible inhibitors; RO, anti other precursors; R, unknown locus of inhibition. Topoisomerase inhibitor: T1, topoisomerase I inhibitors; T2, topoisomerase II inhibitors. Tubulin-active antimitotic agents: TU

cluster group	map color	MOA	Accuracy (%)
G1	blue	RF	100
G2	yellow	T2	92
G3	gray	TU	100
G4	green	DP	100
G5	pink	TU	60
G6	cyan	AC, A7	94
G7	yellowish green	DI, DP, DR, RI, RO, R	82
G8	white	A2, A7	100
G9	red	T1, T2	75

Table 3. Summary of clustering compounds in this study. See Table 2 for the abbreviations for MOA. Accuracy was calculated as described in “2.9 sSOM analysis of Chemicals“

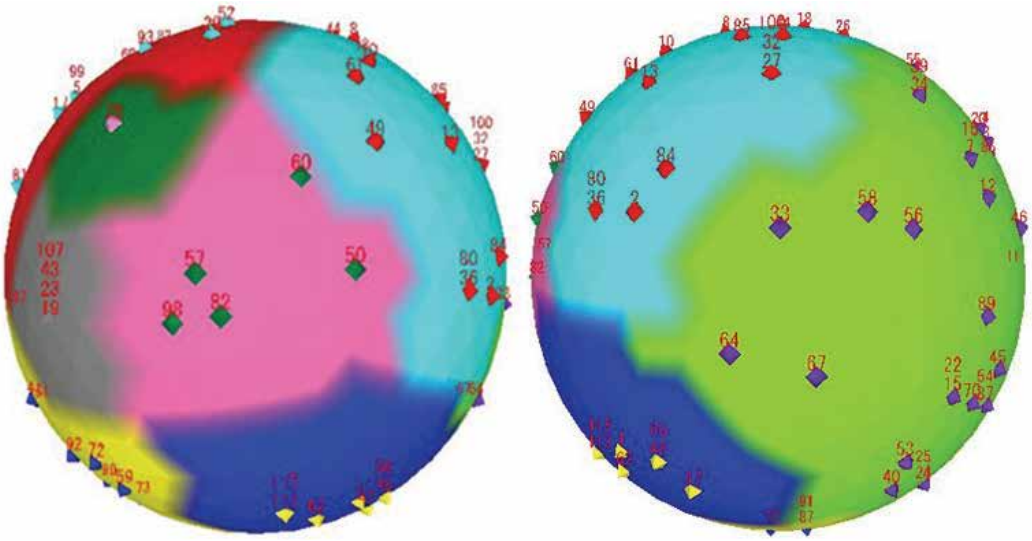


Fig. 5. Projections of clustered bioactive compounds by sSOM. Cluster colors: G1, blue; G2, yellow; G3, gray; G4, green; G5, pink; G6, cyan; G7, yellowish green; G8, white; G9, red. Numbers at the nodes indicate the ID of compounds. Two views on a single global map are shown from the opposite directions

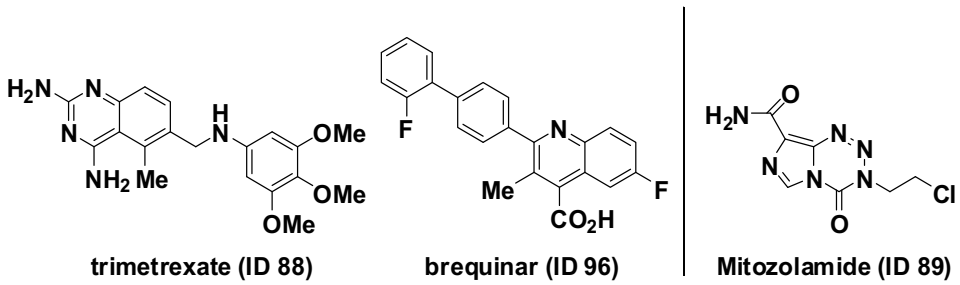


Fig. 6. Chemical structures of trimetrexate (ID 88), DUP 785 (ID 96), and mitozolamide (ID 89)

4. Conclusions

In order to characterize cells and tissues, gene expression profiling is one of the most popular procedures nowadays. Through these procedures, identification of cell surface markers specific to some cells or tissues is a key for diagnosing and molecular targeting. DNA microarray is a high-throughput technology believed to be a powerful tool to find genes differentially expressed in the cells or tissues. Although it can provide critically important and useful information even from one experiment, the amount of data is usually too large to be handled. Therefore, highly sophisticated software is expected to support to transform the multidimensional datasets into simple dimensions or glyphs. For example, visual cues such as shape and color, which make it comprehensive for researchers to

recognize and analyze the patterns hidden in the datasets. Here we successfully demonstrated cell surface marker analyses using our DNA microarray coupled with novel sSOM clustering procedure. The cell surface markers, which are common and specific to cancer derived cells, are proposed in this study and further assessment is now underway.
Here we have also examined sSOM for the classification of chemical compounds. sSOM successfully clustered 116 anti-cancer agents into 9 groups by their MOA using simple chemical descriptors as inputs. So we are now trying to apply this procedure to larger dataset for virtual screening.
Thus, we conclude sSOM is a powerful tool for data mining, knowledge discovery and visualization of multi-dimensional data.

5. References

Abou-Sharieha, S; Sugii, Y; Tuoya; Yu, D; Chen, L; Tokutaka, H & Seno, M. (2009) Identifi cation of TM9SF2 as a candidate of the cell surface marker common to breast carcinoma cells. *Clin. Oncol. Cancer Res.* 6(1), 1-9, ISSN: 1674-5361.

Barnes, N.L.; Khavari, S.; Boland, G.P.; Cramer, A.; Knox, W.F. & Bundred, N.J. (2005) Absence of HER4 expression predicts recurrence of ductal carcinoma in situ of the breast. *Clin. Cancer Res*. 11(6), 2163-2168, ISSN: 1078-0432.

Ben-Dor, A.; Shamir, R. & Yahkini, Z. (1999) Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4) 281-297, ISSN: 1557-8666.

Brüstle, M.; Beck, B.; Schindler, T.; King, W.; Mitchell, T & Clark T. (2002) Descriptors, physical properties, and drug-likeness. *J. Med. Chem.* 45(16), 3345-3355, ISSN: 0022-2623.

Fogel, M.; Friederichs, J.; Zeller,Y.; Husar, M.;, Smirnov, A.; Roitman, L.; Altevogt, P. & Sthoeger, Z.M. (1999) CD24 is a marker for human breast carcinoma. *Cancer Lett*. 143(1), 87-94, ISSN: 0304-3835.

Gasparini, G.; Gullick, W.J.; Maluta, S.; Dalla Palma, P.; Caffo, O.; Leonardi, E.; Boracchi, P.; Pozza, F.; Lemoine, N.R.; & Bevilacqua, P. (1994) C-erbB-3 and c-erbB-2 protein expression in node-negative breast carcinoma-an immunocytochemical study. *Eur J Cancer*, 30A(1), 16-22, ISSN: 0014-2964.

Gasteiger, J.; Teckentrup, A.; Terfloth, L. & Spycher, S. (2003) Neural networks as data mining tools in drug design. *J. Phys. Org. Chem.* 16 (4), 232-245, ISSN: 1099-1395.

Holbro, T.; Civenni, G.; Hynes, N.E. (2003) The ErbB receptors and their role in cancer progression. *Exp Cell Res.* 284(1), 99-110, ISSN: 0014-4827.

Kaiser, D.; Terfloth, L.; Kopp, S.; Schulz, J.; Laet, de R.; Chiba, P.; Ecker F. G. & Gasteiger J. (2007) Self-organizing maps for identification of new inhibitors of P-glycoprotein. *J. Med. Chem.* 50(7), 1698-1702, ISSN: 0022-2623.

Kaufman, L. & Rousseeuw, P.J. (1990) Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, ISBN: 978-0-471-73578-6, New York.

Keiser, M.J.; Setola, V.; Irwin, J.J.; Laggner, C.; Abbas, A.I.; Hufeisen, S.J.; Jensen, N.H.; Kuijer, M.B.; Matos, R.C.; Tran, T.B.; Whaley, R.; Glennon, R.A.; Hert, J.; Thomas, K.L.; Edwards, D.D.; Shoichet, B.K. & Roth, B.L. (2009) Predicting new molecular targets for known drugs. *Nature* 462(7270), 175-181, ISSN 0028-0836.

Lemoine, N.R.; Barnes, D.M.; Hollywood, D.P.; Hughes, C.M.;Smith, P.; Dublin, E.; Prigent, S.A.; Gullick, W.J.;& Hurst HC. (1992) Expression of the ERBB3 gene product in breast cancer. *Br J Cancer* 66(6), 1116-1121, ISSN 0007-0920.

Li J. & Gramatica P. (2010) Classification and virtual screening of androgen receptor antagonists. *J. Chem. Inf. Model.* 50(5), 861-874, ISSN: 1549-9596.

MacQueen, J. (1965) Some methods for classification and analysis of multivariate observations. In: *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281-297.

Melville, J. L.; Burke, E. K.; & Hirst, J. D. (2009) Machine learning in virtual screening. *Combinatorial Chemistry and High Throughput Screening*, 12(4), 332-343, ISSN 1386-2073.

Naidu, R.; Yadav, M.; Nair, S. & Kutty, M.K.(1998) Expression of c-erbB3 protein in primary breast carcinomas. *Br J Cancer.* 78(10), 1385-1390, ISSN 0007-0920.

Quinn, C.M.; Ostrowski, J.L.; Lane, S.A.; Loney, D.P.; Teasdale, J., Benson, F.A. (1994) C-erbB-3 protein expression in human breast cancer: comparison with other tumor variables and survival. *Histopathology*, 25(3), 247-252, ISSN 0309-0167.

R Development Core Team (2007) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.

Ray, P.S.; Wang, J.; Qu, Y.; Sim, M.S., Shamonki, J; Bagaria, S.P.; Ye, X.; Liu, B.; Elashoff, D.; Hoon, D.S.; Walter, M.A.; Martens, J.W.; Richardson, A.L.; Giuliano, A.E.& Cui, X. (2010) FOXC1 is a potential prognostic biomarker with functional significance in basal-like breast cancer. *Cancer Res.*, 70(10), 3870-3786, ISSN 0008-5472.

Reddy S. A.; Pati P. S.; Kumar P. P.; Pradeep N. H. & Sastry N. G. (2007) Virtual screening in drug discovery – a computational perspective. *Current Protein And Peptide Science*, 8(4), 329-351, ISSN 1389-2037.

Renner, S.; Hechenberger, M.; Noeske, T.; Böcker, A.; Jatzke, C.; Schmuker, M.; Parsons, G. C.; Wel, T. & Schneider G. (2007) Searching for Drug Scaffolds with 3D Pharmacophores and Neural Network Ensembles. *Angew. Chem. Int. Ed.* 46(28), 5336-5339, ISSN: 1521-3773.

Schabath, H.; Runz, S.; Joumaa, S. & Altevogt, P. (2006) CD24 affects CXCR4 function in pre-B lymphocytes and breast carcinoma cells. *J Cell Sci.*, 119(Pt2), 314-325, ISSN: 0021-9533.

Schneider, G. & Nettekoven, M. (2003) Ligand-Based Combinatorial Design of Selective Purinergic Receptor (A_{2A}) Antagonists Using Self-Organizing Maps. *J. Comb. Chem.* 5(3), 233-237, ISSN: 1520-4766.

Schneider, P. & Schneider, G. (2003) Collection of Bioactive Reference Compounds for Focsed Library Design. *QSAR Comb. Sci.* 22(7), 713-718, ISSN: 1868-1751.

Shiau, C.E.; Lwigale, P.Y.; Das, R.M.; Wilson, S.A.& Bronner-Fraser M. (2008) Robo2-Slit1 dependent cell-cell interactions mediate assembly of the trigeminal ganglion. *Nat. Neurosci.*, 11(3), 269-276, ISSN: 1097-6256.

Singh, A.P.; Senapati, S.; Ponnusamy, M.P.;, Jain, M.; Lele, S.M.; Davis, J.S.; Remmenga, S.& Batra SK. (2008) Clinical potential of mucins in diagnosis, prognosis, and therapy of ovarian cancer. *Lancet Oncol.* 9(11), 1076-1085, ISSN: 1470-2045.

Steinbeck, C.; Han, Y.; Kuhn, S.; Horlacher, O.; Luttmann, E. & Willighagen, E. (2003) The chemistry development kit (CDK): An open source Java library for chemo- and bioinformatics. *J. Chem. Inf. Comput. Sci.* 43(2), 493-500, ISSN: 1549-9596.

Travis, A.; Pinder, S.E.; Robertson, J.F.; Bell, J.A.; Wencyk, P.; Gullick, W.J.; Nicholson, R.I.; Poller, D.N.; Blamey, R.W.; Elston, C.W. & Ellis, I.O. (1996) C-erbB-3 in human breast carcinoma: expression and relation to prognosis and established prognostic indicators. *Br J Cancer* 74(2), 229-233, ISSN 0007-0920.

Tuoya; Hirayama, K; Nagaoka, T; Yu, D; Fukuda, T; Tada, H; Yamada, H. & Seno M. (2005) Identification of cell surface marker candidates on SV-T2 cells using DNA microarray on DLC-coated glass. *Biochem. Biophys. Res. Commun.* 334(1), 263-268, ISSN: 0006-291X.

Tuoya; Sugii, Y.; Satoh, H.; Yu, D.; Matsuura, Y.; Tokutaka, H. & Seno, M. (2008) Spherical self-organizing map as a helpful tool to identify category-specific cell surface markers. *Biochem Biophys Res Commun.* 376(2), 414-418, ISSN: 0006-291X.

van Osdol, W.W.; Myers, T.G.; Paull, K.D.; Kohn, K.W. & Weinstein, J.N. (1994) Use of the Kohonen self-organizing map to study the mechanisms of action of chemotherapeutic agents. *J. Natl Cancer Inst.* 86(24), 1853-1859, ISSN 0027-8874.

Wang, Y.; Li, Y.; Yang, S. & Yang, L. (2005) Classification of substrates and inhibitors of P-glycoprotein using unsupervised machine learning approach. *J. Chem. Inf. Model.* 45(3), 750-757, ISSN: 1549-9596.

21

Application of Self-Organizing Maps in Chemistry. The Case of Phenyl Cations

Daniele Dondi, Armando Buttafava and Angelo Albini
University of Pavia, Viale Taramelli 10-12, 27100 Pavia
Italy

1. Introduction

We have recently become interested in the application of Self-Organizing Maps (SOM) during a computational study on phenyl cations. As data accumulated, we realized that the analysis and interpretation of results, particularly when many variables were involved, could lead to a cognitive overload. In fact, it is for this reason that SOM found applications in chemistry in several problems, where the classification of large databases was not immediate, or the identification of the most characterizing properties of each class not obvious, since the a priori subdivision of the observed (complex) behaviour in more simple properties was not possible, as it is often the case.

The applications of SOM in chemistry are at present limited in number, but sufficient for indicating the potential of the method. The most important application is probably in the Quantitative Structure Activity Relationship (QSAR). The QSAR is a statistical method used in drug discovery where a correlation between biological activity (including desirable therapeutic effects and undesirable side effects) of chemicals (drugs/toxicants/environmental pollutants) with descriptors representative of molecular structure and/or properties is searched. Drug design has often the need to process enormous amounts of data, in which complex relationships have to be studied and modelled and is thus advantageously confronted by using SOM (Bienfait, 1994; Gramatica, 2007).

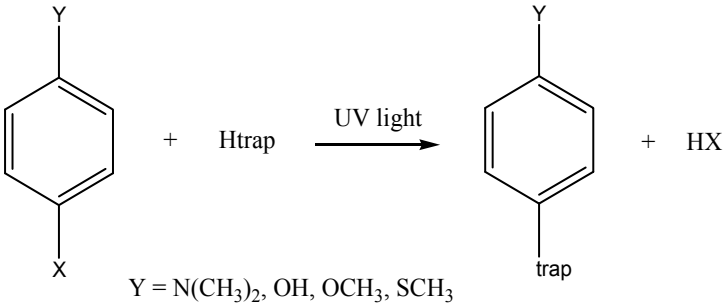
However, applications are really varied, including for example the analysis of complex mixtures such as raw oil spills (Fernàndez-Varela et al., 2010), the interpretation of spectra (Dow et al., 2004; Villman et al., 2008), studies of molecular conformation (Hyvönen et al., 2001), as well as the stucture of polymers (Llyod et al., 2008) or crystals (Willighagen et al., 2008), proteonomics (Herrero & Dopazo, 2002) and many others.

Closer to the topic of the present study, the SOM methods has been applied for recognizing the chemical properties of molecules, e.g. for assigning a degree of aromaticity (Alonso & Herradin, 2008), or more generally for predicting the chemical reactivity and its selectivity (Chen & Gasteiger, 1997; Noeske et al., 2006). As a matter of fact, phenyl cations had been all by unknown to chemists up to a decade ago, when it was discovered that electron-donating substituted phenyl halides, sulfonates and phosphates smoothly undergo heterolytic cleavage forming such intermediates in the triplet state, and that in this multiplicity these react efficiently with π, not with n, nucleophiles (Fagnoni & Albini, 2005). The synthetic potential of such intermediates seemed valuable, but the scope of both generation and

reaction required extensive work. Thus it was of interest to carry out computational studies and to apply the SOM method to the data case in order to recognize and predict the structural and chemical properties.
We used self-organizing maps, as developed by Kohonen (Kohonen, 2001; Wehrens et al., 2007). This is an unsupervised neural network that preserves on a two dimensional plane topological relations. We introduced as many as possible optimized geometric parameters and searched for structure similarities. Indeed a visually effective result was obtained through a Sammon map, which clearly separated triplet and singlet cations and helped to rationalize the different substituent effect on the two species.
In the following, the approach used will be illustrated and new examples will be discussed.

2. Description of the problem

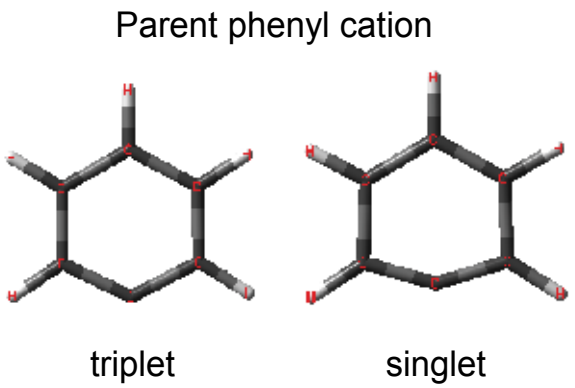
Density Functional Theory (DFT) is a computational approach that has shown to be able to determine accurately energy differences, geometries and spectroscopic data of chemical compounds. The method is particularly useful if applied to transient species, such as excited states or reaction intermediates, for which a direct observation could be difficult.
During the last years, in our laboratories a reaction of photoarylation was studied (see Scheme 1).



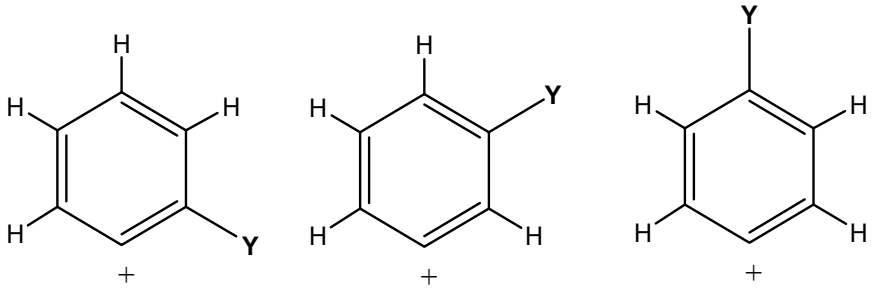
Scheme 1. Photoinduced arylation

This process is particularly interesting since such arylations can otherwise be obtained only under catalysis by transition metals, a highly versatile method that suffers however of some limitations, in particular related to the catalysts themselves, often delicate, expensive and polluting. Substituting light for heavy metal catalysis makes the reaction more compatible with the principles of green chemistry, i.e. a photoinduced reaction is more ecosustainable (Fagnoni & Albini, 2005).
We have demonstrated that the active chemical intermediate is the aryl cation, and more particularly the triplet spin state. Such an intermediate is highly reactive (it is a carbocation), but is nevertheless characterized by a high selectivity towards suitable traps. The other spin state, the singlet, is on the contrary completely unselective and reacts at diffusion controlled rate practically with everything. As a result, when generated in solution it typically reacts with the solvent.
The computational approach via DFT has been used often for cations and in the present case a first gratifying result was that the computed UV spectrum of the triplet cation was in

accordance with the experimentally determined spectrum of the intermediate, as obtained by flash photolysis (Manet et al., 2008).
From preliminary studies, it is known that the geometry of singlet and triplet phenyl cations differs significantly. In fact, the triplet phenyl cations have a more regular hexagonal geometry with respect to the singlet phenyl cation.



The introduction of substituents in such high-energy intermediates may induce important effects on structure and chemistry with respect to the parent cation. However, it is not obvious how this effect will operate. Singlet phenyl cations have a $\pi^6 n^0$ structure, that is the positive charge is localized at the divalent carbon and lies in the plane of the molecule. Therefore, mesomeric effects should not operate. On the other hand, triplets have a $\pi^5 n^1$, that is a diradicalic character, one radical site at the divalent carbon, one delocalized over the aromatic ring, which has thus a radical cation character. In this case, mesomeric effects may be significant. For simplicity, we considered monosubstituted derivatives, with the group in the three positions, ortho, meta and para, as indicated in the formulae below, where Y stands for an atom or a group of atoms.



3. Data analysis through SOM

The target of the investigation was determining which effect had substituents (and their position) on structure and energy of phenyl cations and how these difference were reflected in the chemical reactivity. Perhaps unexpectedly, in view of the considerations above, the singlet cations had a geometry heavily deformated, with strong dependence on the substituent, despite the fact that these species conserved the intact aromatic sextet. On the

contrary triplets, which had lost the sextet, maintained quite closely the planar hexagonal geometry. The rationalization was not obvious. In order to have a significant results, a sufficiently differentiated choice of substituents had to be considered, avoiding on the other hand to overcome a manageable number. We thus decided to consider eight substituents Y (in the three positions), chosen among those expected to induce large changes due to the electron-donating or –abstracting effect, and to compare their properties with those of the parent cation (Lazzaroni et al, 2008, 2010). The groups, ordered from electron-donating to electron-withrawing were: NH_2, SCH_3, OCH_3, CH_3, $Si(CH_3)_3$, H, CN and NO_2.
Even in this oversimplified space, these corresponded to a total of 8 (substituents) x 3 (positions) x 2 (spin states) = 48 + 1 structures that were calculated. It was immediately apparent that the comparison by hand of 49 structures in a 3D space was scarcely promising due to the large number of data available. On the other hand, limiting the choice to a few key geometrical parameters was no appealing alternative, due to the lack of objectivity in the choice of such parameters.
This thus appeared as a suitable case for making recourse to unsupervised learning, which can help to find correlations and similarities among large series of data, if a correct input is given.
In the data set, we avoided the use of parameters directly related to the substituent Y, because a trivial categorization factor would be otherwise inserted. Thus, the length of the C-Y bond was not included because it depended on the substituent's nature). For this reason, only parameters related to the carbon skeleton of the aryl cation ring were introduced. In order to describe exhaustively the structure of the six carbon atoms in the space, bond lengths, bond angles and dihedral angles of the phenyl ring were introduced.

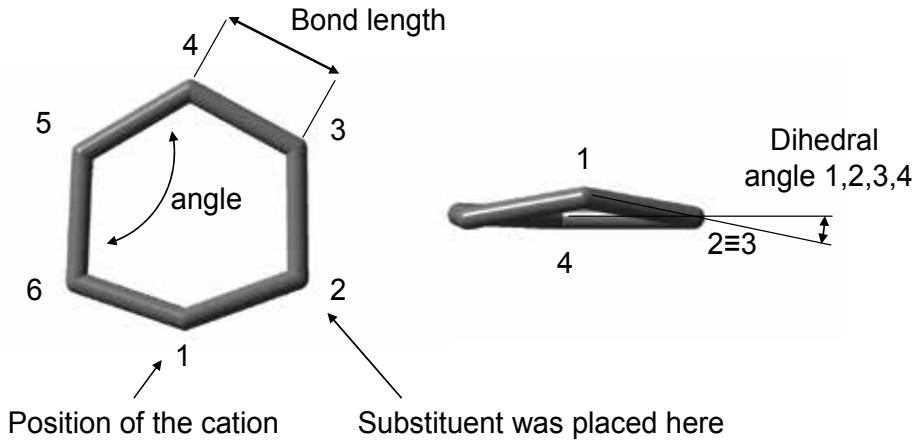


Fig. 1. Example of the geometrical data inserted as input

Furthermore, as the divalent carbon (C_1) underwent the most extensive geometrical modifications, all parameters were inserted starting from that carbon, and the shortest path to reach the substituent was adopted. A graphical representation of the choice of parameters and the numbering is given in Figure 1. This choice is essential for the correct application of the SOM.
The complete training set used is given in Table 1.

1.397	1.378	1.444	1.436	1.378	1.404	127.020	116.152	119.540	121.580	118.155	117.156	0.000	0.007	0.006	0.000	0.003	0.003	4-OCH_3^T
1.344	1.410	1.423	1.415	1.415	1.341	140.850	105.290	122.150	120.510	119.940	107.310	9.130	1.920	2.540	10.370	27.090	26.188	4-OCH_3^S
1.397	1.378	1.440	1.440	1.379	1.397	126.790	116.760	119.400	120.890	119.400	116.760	0.005	0.000	0.009	0.008	0.000	0.007	4-NH_2^T
1.360	1.400	1.430	1.430	1.400	1.360	133.780	108.790	120.630	119.780	120.660	108.750	13.100	0.360	0.360	13.100	34.740	34.740	4-NH_2^S
1.406	1.376	1.444	1.443	1.377	1.406	127.140	116.000	120.500	119.850	120.490	116.010	0.024	0.220	0.220	0.022	0.190	0.190	4-CH_3^T
1.327	1.433	1.403	1.403	1.433	1.326	146.720	104.120	123.090	118.850	123.090	104.110	0.317	0.117	0.113	0.323	1.322	1.319	4-CH_3^S
1.405	1.377	1.442	1.442	1.377	1.405	127.950	116.000	118.990	122.060	118.980	116.000	0.000	0.000	0.000	0.000	0.000	0.000	4-CN^T
1.328	1.429	1.407	1.407	1.429	1.328	147.560	104.150	121.640	120.860	121.640	104.150	0.011	0.003	0.000	0.007	0.029	0.030	4-CN^S
1.412	1.378	1.434	1.434	1.378	1.412	127.680	115.680	119.320	122.330	119.320	115.680	0.000	0.004	0.004	0.000	0.000	0.000	H^T
1.327	1.435	1.395	1.395	1.435	1.327	147.320	104.000	121.770	121.120	121.770	104.000	0.009	0.005	0.004	0.005	0.000	0.009	H^S
1.352	1.447	1.440	1.375	1.428	1.397	126.020	115.800	120.650	119.640	120.370	117.520	0.004	0.004	0.000	0.008	0.008	0.000	3-OCH_3^T
1.322	1.460	1.420	1.384	1.449	1.324	144.680	107.720	118.450	121.970	123.420	104.230	5.760	3.678	2.528	5.082	2.380	3.007	3-OCH_3^S
1.353	1.442	1.442	1.377	1.420	1.397	126.580	115.730	119.990	120.380	120.080	117.260	0.013	0.022	0.009	0.012	0.023	0.010	3-NH_2^T
1.324	1.468	1.437	1.374	1.455	1.324	141.400	108.270	118.480	121.950	120.400	107.790	6.835	3.674	7.857	13.846	1.487	2.153	3-NH_2^S
1.361	1.416	1.464	1.386	1.404	1.423	126.940	116.060	118.780	122.380	118.600	117.250	0.010	0.013	0.007	0.000	0.004	0.000	3-CH_3^T
1.324	1.447	1.402	1.393	1.434	1.326	147.600	105.700	118.890	122.350	122.610	102.830	0.006	0.006	0.010	0.012	0.021	0.018	3-CH_3^S
1.360	1.424	1.460	1.381	1.412	1.417	126.620	114.800	120.980	120.640	118.930	118.030	0.000	0.000	0.000	0.000	0.000	0.000	3-CN^T
1.326	1.446	1.405	1.392	1.437	1.326	147.160	105.480	119.950	121.110	123.230	103.050	0.014	0.009	0.006	0.012	0.007	0.006	3-CN^S
1.430	1.434	1.374	1.431	1.413	1.360	125.360	117.480	118.380	121.720	121.080	115.980	0.000	0.000	0.000	0.003	0.005	0.003	2-OCH_3^T
1.352	1.444	1.395	1.393	1.430	1.328	145.600	100.940	125.000	121.610	118.400	108.240	3.089	2.073	1.482	2.323	6.127	6.282	2-OCH_3^S
1.422	1.441	1.373	1.423	1.417	1.360	125.300	117.000	119.020	121.300	121.030	116.350	0.000	0.000	0.000	0.000	0.003	0.004	2-NH_2^T
1.380	1.446	1.377	1.412	1.429	1.336	136.630	109.140	118.870	121.970	122.490	106.030	9.451	1.721	3.944	13.353	28.715	27.424	2-NH_2^S
1.440	1.420	1.380	1.446	1.404	1.367	127.440	115.500	119.640	121.550	120.860	115.020	0.023	0.012	0.000	0.004	0.009	0.022	2-CH_3^T
1.326	1.450	1.392	1.397	1.426	1.333	148.870	100.360	124.390	120.980	119.960	105.440	0.007	0.003	0.000	0.004	0.002	0.019	2-CH_3^S
1.436	1.423	1.380	1.440	1.410	1.363	125.680	117.780	118.240	121.410	121.620	115.270	0.000	0.000	0.000	0.000	0.000	0.000	2-CN^T
1.343	1.444	1.391	1.395	1.438	1.322	147.150	103.310	122.250	121.290	121.530	104.470	0.006	0.004	0.003	0.009	0.011	0.006	2-CN^S
1.394	1.385	1.427	1.435	1.385	1.389	126.822	116.882	119.344	120.823	119.844	116.286	0.000	0.000	0.000	0.000	0.000	0.000	4-SCH_3^T
1.338	1.410	1.409	1.415	1.405	1.341	138.272	107.636	120.980	119.909	121.511	107.313	10.663	1.108	0.055	8.780	26.889	27.882	4-SCH_3^S
1.405	1.381	1.425	1.425	1.381	1.405	127.815	116.098	117.660	124.669	117.660	116.099	0.139	0.076	0.072	0.138	0.073	0.073	4-NO_2^T
1.330	1.427	1.395	1.395	1.427	1.330	148.485	103.695	120.350	120.425	120.350	103.695	0.084	0.049	0.049	0.080	0.034	0.050	4-NO_2^S
1.394	1.376	1.434	1.434	1.376	1.394	127.066	115.915	120.936	118.807	120.940	115.910	2.941	7.298	7.299	2.942	1.718	1.719	4-$Si(CH_3)_3^T$
1.315	1.444	1.390	1.393	1.438	1.318	145.545	104.524	123.598	118.084	123.422	104.828	0.003	0.000	0.000	0.000	0.000	0.003	4-$Si(CH_3)_3^S$
1.354	1.422	1.429	1.379	1.405	1.387	126.914	115.380	120.221	120.651	119.640	117.195	0.005	0.003	0.000	0.000	0.003	0.005	3-SCH_3^T
1.318	1.457	1.417	1.377	1.439	1.321	143.067	107.079	119.179	122.218	120.420	107.115	3.925	1.714	6.336	10.266	9.914	2.792	3-SCH_3^S
1.383	1.385	1.440	1.403	1.392	1.417	127.366	113.267	123.263	119.813	118.976	117.312	0.000	0.018	0.285	0.573	0.646	0.342	3-NO_2^T
1.323	1.425	1.391	1.399	1.428	1.334	146.584	103.116	124.085	119.624	121.662	104.930	0.017	0.008	0.013	0.018	0.007	0.010	3-NO_2^S
1.364	1.404	1.457	1.392	1.387	1.416	127.245	116.565	117.542	123.395	118.320	116.932	0.000	0.003	0.000	0.000	0.003	0.000	3-$Si(CH_3)_3^T$
1.310	1.448	1.393	1.392	1.416	1.331	148.014	100.910	123.280	122.595	117.920	107.281	0.132	0.103	0.081	0.102	0.260	0.266	3-$Si(CH_3)_3^S$
1.398	1.410	1.378	1.408	1.402	1.363	123.640	119.104	118.505	120.666	121.334	116.751	0.259	0.237	0.029	0.155	0.132	0.074	2-SCH_3^T
1.333	1.448	1.381	1.396	1.426	1.325	145.077	104.011	121.794	121.794	120.732	105.748	3.741	0.189	0.474	4.282	13.101	12.737	2-SCH_3^S
1.417	1.389	1.398	1.440	1.395	1.379	124.639	120.282	116.617	121.912	121.236	115.269	2.046	1.824	1.508	1.288	1.607	2.080	2-NO_2^T
1.403	1.411	1.381	1.440	1.389	1.371	128.776	114.430	120.001	121.667	120.542	114.585	0.164	0.103	0.000	0.027	0.053	0.150	2-$Si(CH_3)_3^T$
1.294	1.452	1.387	1.395	1.420	1.333	149.011	102.573	121.807	121.460	120.847	104.301	0.004	0.000	0.003	0.005	0.003	0.000	2-$Si(CH_3)_3^S$

Table 1. Complete training set used. The first group of six columns are bond lengths (units are Ångstrom), the second group are angles (degrees) and the third dihedral angles (degrees). The last column is the coded name of the phenyl cation (see text for a description)

For the SOM optimization, we started with 1000 cycles and with an initial learning rate parameter (alpha) of 0.05. The map was then refined with 10000 cycles starting with alpha = 0.02. A bubble neighbouring was used and the radius was 10 for the first step and 3 for the last step.

The software package utilized for the calculations of SOMs was SOMPAK (Kohonen, 1997), a program developed by the Helsinki University of Technology.

A first trial with a rectangular 2x2 grid was carried out in order to verify the correctness of the input data; indeed, the database was correctly divided into two classes, the first one containing only singlets and the second one only triplets.
A more advanced level of information could be obtained by increasing the size of the map. The program used allowed the creation of two different lattice arrangements of neurons: rectangular and hexagonal. After some experimenting, we found convenient to adopt the rectangular lattice.
The calculated rectangular 5x5 map containing all the phenyl cation intermediates is reported below (Table 2). The numbering in the table is referred to the substituent position, as indicated in Figure 2, while the spin state is specified by the capital letter superscripts S for singlets and T for triplets. The formula of the substituent Y is placed after the number and H is used for the unsubstituted (parent) phenyl cation.

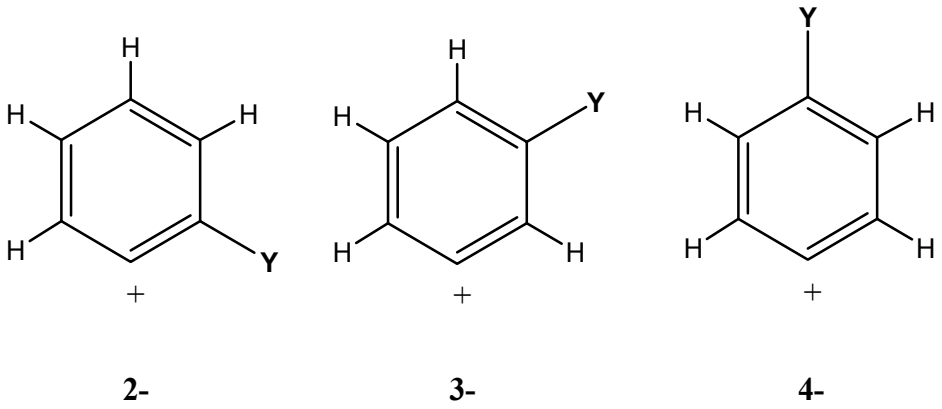


Fig. 2. Substituent numbering used in Table 2

The differences in geometry of singlet and triplet phenyl cations is well apparent, with the two classes lying in different positions in the table.
In this 5x5 map, 17 cathegories were represented, each one of them corresponding to an active neuron.
By analyzing the position of the entries in the table, some generalizations could be made; for example, when considering the singlets, substituents such as CN, NO_2, $Si(CH_3)_3$ and CH_3 were all grouped in the bottom left corner, close to the parent cation.
A chemical meaning can be immediately associated with this result. Thus, all of the substituents above are known to be electron-withdrawing or almost electron-neutral groups. The presence of an empty sp^2 orbital at C_1 in the place of a σ bond makes little difference for the intact π^6 system with which the substituents mainly interact, resulting in a moderate effect on the geometry of the phenyl cation.
On the other hand, cations bearing electron-donating substituents are found in the left part of the table, but their distribution is more spread across the table, with an effect that varies with the position in a different way for each substituent. The substituents with a strong mesomeric effect, such as amino, methoxy and thiomethoxy group cluster at the upper left corner. Clearly, the electron-rich π system does interact with the empty σ orbital, and both the nature and the position of ring substituents are important in determining the final geometric parameters.

As for the triplets, these are grouped in three big classes in the rightmost column. Each of them containing (mainly) cations bearing the substituent in the same position, regardless of their-donating or -withdrawing character. Apparently, with triplets the effect of the position of the substituent overcame the electronic effect in causing a deformation of the phenyl cation.

4-OCH_3^S 4-NH_2^S 2-NH_2^S 4-SCH_3^S				4-$Si(CH_3)_3^T$
	2-SCH_3^S	3-NH_2^S 3-SCH_3^S		3-NO_2^T
2-OCH_3^S	3-OCH_3^S		2-CH_3^T	4-CH_3^T 3-OCH_3^T 3-NH_2^T 3-CN^T 3-SCH_3^T
2-CH_3^S 3-$Si(CH_3)_3^S$		2-$Si(CH_3)_3^T$	4-OCH_3^T 4-CN^T H^T 3-CH_3^T 4-NO_2^T	4-NH_2^T 4-SCH_3^T
4-CN^S 2-CN^S H^S 4-NO_2^S 3-NO_2^S 2-$Si(CH_3)_3^S$	4-CH_3^S 3-CH_3^S 3-CN^S 4-$Si(CH_3)_3^S$		3-$Si(CH_3)_3^T$	2-OCH_3^T 2-NH_2^T 2-CN^T 2-SCH_3^T 2-NO_2^T

Table 2. 5x5 SOM map obtained for the complete set of calculated phenyl cations

Another useful information is given by the quantization of the error regarding each of the entries of the map. The quantization error is inversely proportional to the matching of each entry line (in our case each phenyl cation) within the assigned class.

Figure 3 shows the pictorial representation of the quantization errors for the 5x5 calculated map; each one of the grid intercepts has a corresponding element in Table 2. Circles with small areas are related to small quantization errors. As can be seen in the figure, a small number of neurons corresponds to classes having most of the best-matching entries. These neurons define a sharp-defined class, i.e. the entries located by this neuron are most similar among them. Considering those of the 'sharp' classes that contain more than one entry, it is found that these corresponds to the four classes evidenced in Table 2, one for the singlets with bearing no electron-donating substituent, three for the triplets with the substituent in each of the three position (that is the classes at positions 1;5, 4;4, 5;3 and 5;5).

Although the distribution in a table does give useful information about any alikeness, a more quantitative representation of class similarities in a calculated map is the projection of neurons through the Sammon's map (see Figure 4).

The label assigned to each point is that of the first entry in the class identified with that neuron. A look at Figure 4 makes it apparent that all of the triplets (squares) lie within a much smaller area with respect to that occupied by singlets (circles). Thus, even if the

number of classes belonging to triplets is similar to that of singlets, the former ones are subject to a smaller geometric variation.
Concerning singlets, the map shows how they are rather scattered on the Sammon's projection. The parent phenyl cation is located at coordinates -18, -10. Electron-donating substituents appear to cause a deformation that increases when passing from substituents in position 2 to the same in position 4.

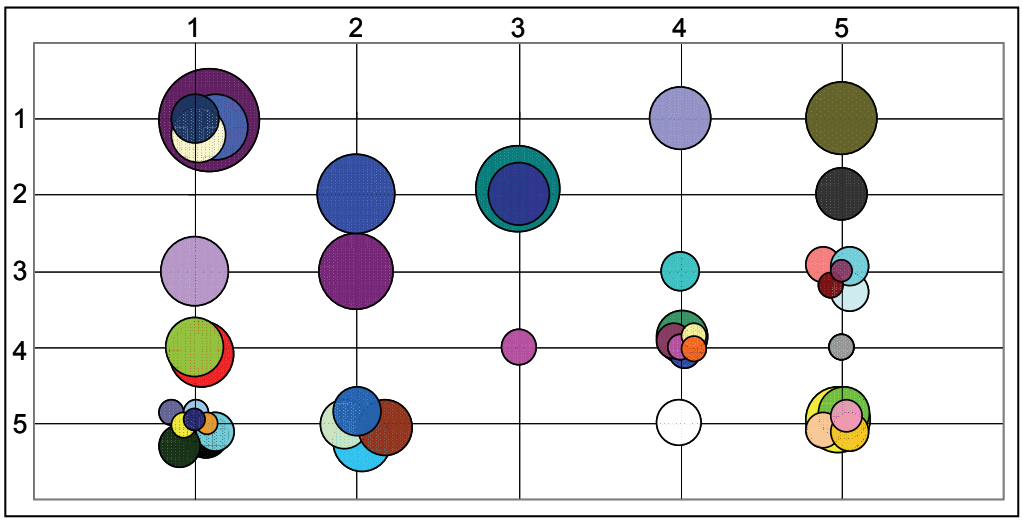


Fig. 3. Quantization error for the calculated 5x5 SOM. The area of the circle is proportional to the quantization error for that entry. Please note that overlap between circles is diminished because a random value was added to the position of the center

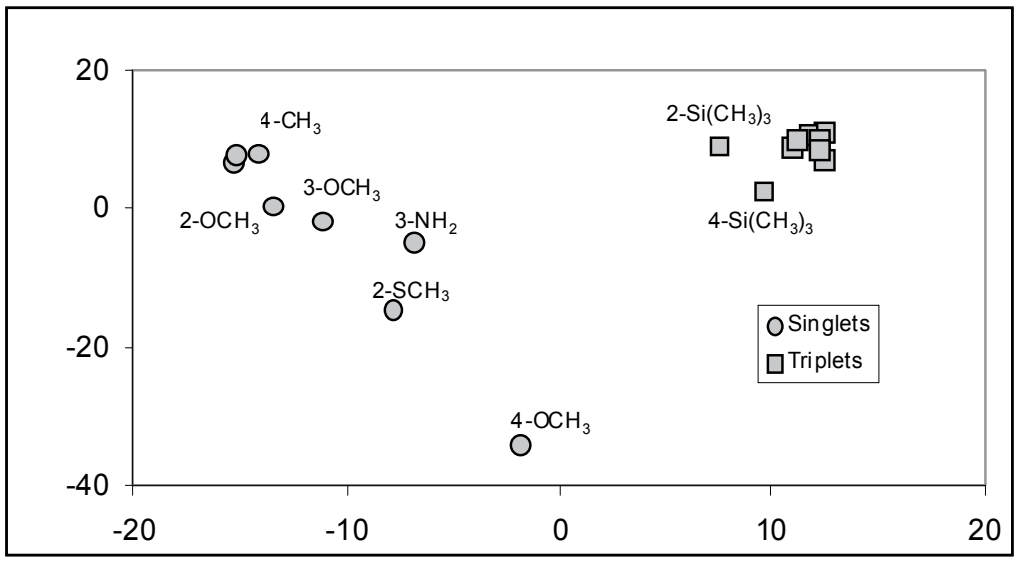


Fig. 4. Sammon's projection for the 5x5 map showed in table 1. Axes are in arbitral units

The above observation affords indications for assessing the effect of substituents on the geometry of phenyl cations. However, arriving at recognizing the key geometrical parameters for each class requires a different approach. This is possible by having recourse to the analysis of 'activation planes'. During the SOM optimization a plane of activation is created for each input variable. In our case we have three groups of six variables each, viz. bond lengths, bond angles and dihedral angles.

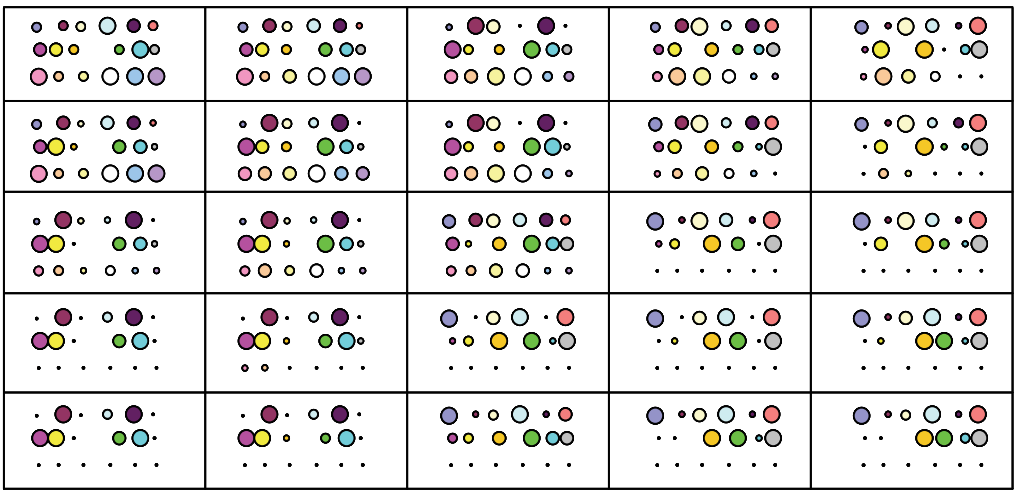


Fig. 5. Activation planes for each element of the grid

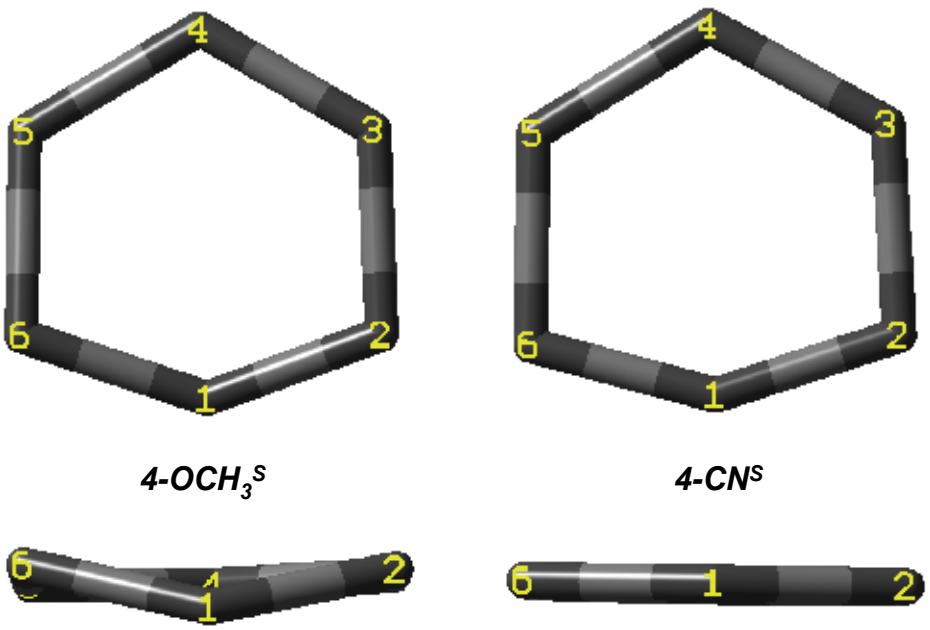


Fig. 6. Geometries of carbon rings for the phenyl cations 4-OCH_3^S and 4-CN^S, top and side view

In Figure 5 these values are shown as circles having an area proportional to the weight of the corresponding input. For every element of Table 2 three lines (bond length, angle, dihedral angle) containing six circles each are a representation of the importance of the input data. As it can be seen in the figure, even table elements (and thus neurons) having no class members have nonzero values. In this picture, is also apparent the meaning of the word neighbourhood in this context (and thus the importance of rectangular or hexagonal grid): neurons in close contact have similar activation weights.
Let us consider the leftmost column of grid shown in Figure 5. It is apparent that when proceeding from the top to the bottom of the figure, the classification algorithm changes from one characterized by a high weight of the dihedral angles to one characterized by a high weight for specific bond lengths (second and fifth bond) and bond angles (1,2,4,5).
Notice that only few of these characteristics are visible by observing the actual geometries (see Figure 6).
In fact, only the dihedral angle differences are immediately appreciated in the side view, while the variation of bond lengths are better visible by looking at the corresponding rows in Table 2. The largest effect among the out-of-plane variations is the lengthening of bonds 2 and 5 for the CN-substituted derivative. This effect could be reasonably assigned to the stabilization ot the radical site by the CN substituent. Correspondingly, the methoxy group stabilizes the charge by the resonance electron donation (see the formulae in Figure 7).

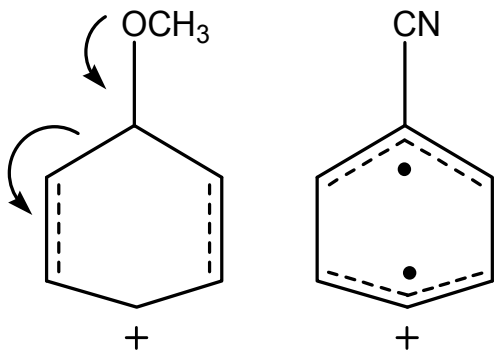
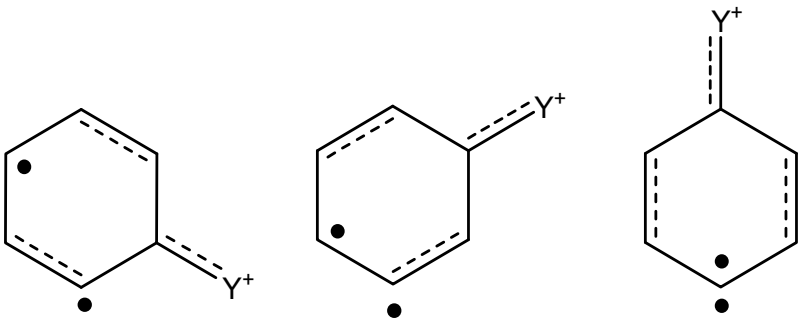


Fig. 7. Contribution of the electronic effect to the stabilization of singlet phenyl cations

In the same line, it may be asked which are the geometrical key parameter for the three distinct classes of triplets located at 4;4, 5;3 and 5;5. Figure 5 tell us that the dihedral angles do not contribute significantly (third row of six circles). In fact, all of these structures lie almost perfectly in a plane.
For these phenyl cations, the structures can be roughly represented as indicated in Figure 8, where the dashed double bond is slightly shorter with respect to other bonds. Since this effect, which is due to the electronic stabilization by the substituent, is quite small, almost no effect is visible observing the 3D structures (also in this case, a close scrutiny of the data in Table 2 reveals the effect, however).
Summing up, the various analyses carried out in this work allow a better understanding of the structure of phenyl cations and the effect of substituents on it. This is much more complex than that obtained by one- or two-parameters linear relationships (Lazzaroni et al., 2008, 2010) and not easily framed in the usual mesomeric/inductive effects. On the other hand, patterns are recognized and may have some predictive effect for further substituents

or for suggesting new chemistry (notice, as an example, the diradicalic nature of triplets, depending only from the position of the substituent, for which is not known at the moment any corresponding reaction.



2- **3-** **4-**

Fig. 8. Schematic structures for 2-, 3- and 4- substituted triplet phenyl cations

Finally, the important question of the relation between energy and geometry in these cations can be confronted. For this aim it is convenient to introduce a isodesmic chemical equation in which a hypothetical reaction is considered. One of these reactions has to be assumed as the reference zero, and in this case the unsubstituted phenyl cation was chosen. The isodesmic reaction considered is thus that shown in Figure 9.

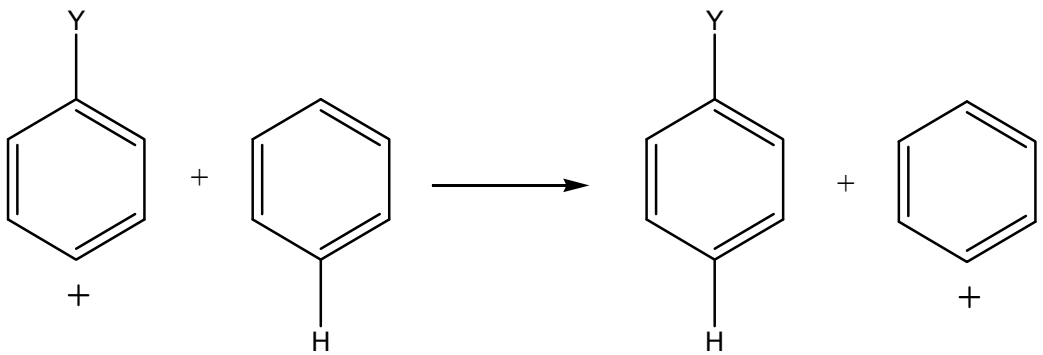


Fig. 9. Isodesmic reaction for the determination of the substituted phenyl cations stability

The calculation of the free energy (ΔG) of this reaction gives a direct information about the stabilization induced by substituent Y. If the free energy is negative for the reaction considered from the left side to the right side, as indicated in the equation the substituted phenyl cation is less stable than the parent cation. If, on the other hand, the reaction has a positive free energy value, the group Y stabilizes the phenyl cation.

If there is a relationship between the geometric classification and the stabilization of the phenyl cation, a pattern should appear by placing in the SOM table the calculated values of free energies. The result is shown in Figure 10, where, for each element of the table, a sphere was drawn having a radius directly proportional to the absolute value of the free energy.

The colour of the sphere indicate the sign of the free energy, blue and red for negative and positive values, respectively. When multiple data are located in the same cell, the sphere is shifted on the right.
As one can see, only a loose correlation between stabilization energy and geometric parameters can be perceived.

4. Conclusion

The geometry and the energy of the singlet and triplet state of a series of substituted phenyl cations have been investigated by DFT methods. The pattern of the effects with reference to the nature of the substituent and to its position has been recognized through the SOM method. The regular hexagon geometry of ring carbons in the triplet state is minimally affected by substituents. The singlet cation is more heavily deformed (cumulene character at the C2-C1-C6 moiety) and is affected to a large extent both by the position and by the nature of the substituent in particular with electron-donating substituents that cause a ring carbon to shift out of plane.

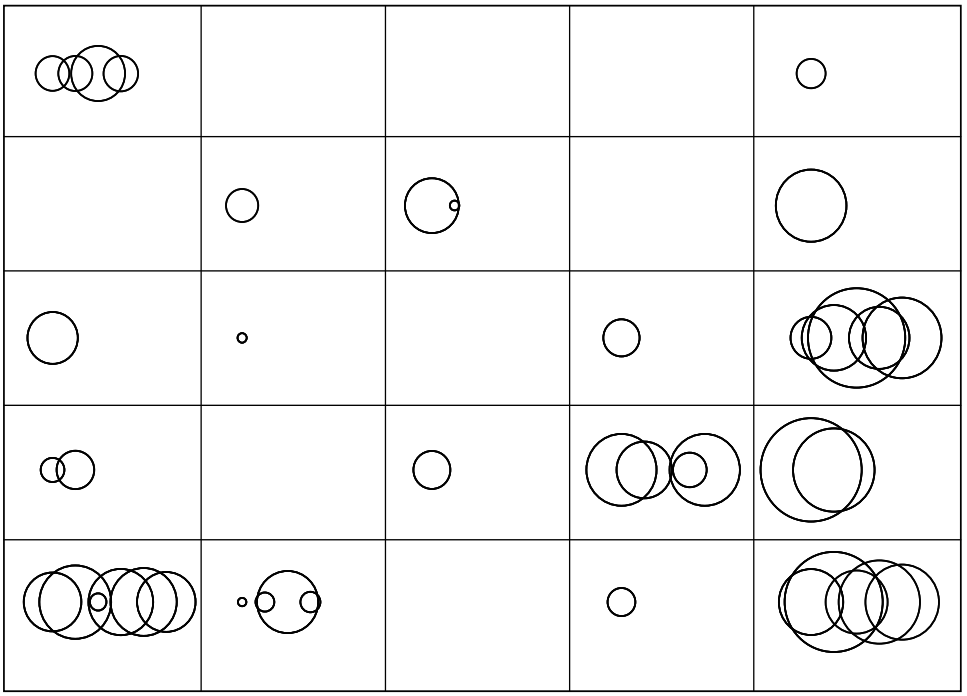


Fig. 10. Stabilization energies of phenyl cations corresponding to the classification in Table 2 (See text)

It was attempted to find a correlation between geometric structure and stabilization energy by the use of this procedure, but this turned out to be weak. We plan to take further advantage of the SOM method for the analysis of the structure and chemical properties of these novel intermediates, phenyl cations.
More generally, we feel that the SOM method should be more largely used for predicting chemical reactivity, both for novel intermediates, as is the case here, and for old ones, where

perhaps it may reveal that the substituent effect is treated in a simplicistic way when using a traditional approach.

5. References

Alonso, M. & Herradin, B. (2007). Neural Networks as a Tool To Classify Compounds According to Aromaticity Criteria, *Chem. Eur. J.*, 13, 3913 – 3923

Bienfait, B. (1994). Applications of High-Resolution Self-Organizing Maps to Retrosynthetic and QSAR Analysis, *J. Chem. Inf. Comp. Sci.*, 34, 890-898

Chen, L. & Gasteiger, J. (1997). Knowledge Discovery in Reaction Databases: Landscaping Organic Reactions by a Self-Organizing Neural Network, *J. Am. Chem. Soc.*, 119, 4033-4042

Dow, L. K.: Kalelkar, S. & Dow, E. R. (2004). Self-organizing maps for the analysis of NMR spectra, *Biosilico*, 2, 157-163

Fagnoni, M. & Albini, A. (2005). Arylation reactions: the photo S 1 path via phenyl cations as an alternative to metal catalysis, *Acc. Chem. Res.*, 38, 713-721.

Fernàndez-Varela, R.; Gòmez-Carracedo, M. P.; Ballabio, D.; Andrade, J. M.; Consonni, V. & Todeschini, R. (2010). Self Organizing Maps for Analysis of Polycyclic Aromatic Hydrocarbons 3-Way Data from Spilled Oils, *Anal. Chem.*, 82, 4264-4271

Gramatica, P. (2007). Principles of QSAR models validation: internal and external Principles of QSAR models validation: internal and external, *QSAR & Comb. Sci.*, 26, 694-701,

Herrero, J.; & Dopazo, J. (2002) Combining Hierarchical Clustering and Self-Organizing Maps for Exploratory Analysis of Gene Expression Patterns, *J. Proteome Res.*, 1, 467-470

Hyvönen, M. T.; Hiltunen, Y.; El-Deredy, W.; Ojala, T.; Vaara, J.; Kovanen, P. Y. & Ala-Korpela, M. (2001). Application of Self-Organizing Maps in Conformational Analysis of Lipids, *J. Am. Chem. Soc.*, 123, 810-816

Kohonen, T.; Hynninen, J.; Kangas, J. & Laaksonen, J. (1996). SOM_PAK: The Self-Organizing Map Program Package. Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, FIN-02150 Espoo, Finland.

Kohonen T. (2001). *Self Organizing Maps*, Springer-Verlag, Berlin

Lazzaroni, S.; Dondi, D.; Fagnoni, M. & Albini, A. (2008). Geometry and Energy of substituted phenyl cations, *J. Org. Chem.*, 73, 206-211.

Lazzaroni, S.; Dondi, D.; Fagnoni, M. & Albini, A. (2010). Selectivity in the reaction of triplet phenyl cations, *J. Org. Chem.*, 75, 315-323.

Manet, I.; Monti, S.; Grabner, G.; Protti, S.; Dondi, D.; Dichiarante, V. & Albini, A. (2008). Revealing phenylium, phenonium, vinylene phenonium and benzenium ions in solution, *Chem.Eur. J.*, 14. 1029-1039.

Lloyd, G. R.; Brereton, G. R. & Duncan, J. C. (2008). Self Organising Maps for distinguishing polymer groups using thermal response curves obtained by dynamic mechanical analysis, *Analyst*, 133, 1046–1059

Noeske, T.; Sasse, B. C.; Stark, H.; Parsons, C. G.; Weil, T. & Schneider, G. (2006). Predicting Compound Selectivity by Self-Organizing Maps: Cross-Activities ofMetabotropic Glutamate Receptor Antagonists, *ChemMedChem*, 1, 1066 – 1068

Villmann, T.; Schleif, F. M.; Kostrzewa, M.; Walch, A. & Hammer, B. (2008). Classification of mass-spectrometric data in clinical proteomics using learning vector quantization methods, *Brief. Bioinform.*, 9, 129-143

Wehrens, R. & Buydens, L. M. C. (2007) Self- and Super-organizing Maps in R: The kohonen Package, *J. Stat. Soft.*, 21, 1-15

Willighagen, E. L.: Wehrens, R.; Melssen, W.; de Gelder, R. & Buydens, L. M. C. (2007). Supervised Self-Organizing Maps in Crystal Property and Structure Prediction, *Cryst. Gro. Des.*, 7, 1738-1745

22

Myoelectric Knee Angle Estimation Algorithms for Control of Active Transfemoral Leg Prostheses

Alberto L. Delis[1,2], Joao L. A. Carvalho[1], Adson F. da Rocha[1], Francisco A. O. Nascimento[1] and Geovany A. Borges[1]
[1]*Department of Electrical Engineering, University of Brasilia, Brasilia-DF*
[2]*Medical Biophysics Center, University of Oriente, Santiago de Cuba*
[1]*Brazil*
[2]*Cuba*

1. Introduction

The electromyographic signal is the electrical manifestation of the neuromuscular activation associated with a contracting muscle. The surface electromyographic (SEMG) signal represents the current generated by ionic flow across the membrane of the muscle fibers that propagates through the intervening tissues to reach the detection surface of an electrode located over skin (De Luca (2006)). The SEMG signal provides a non-invasive tool for investigating the properties of skeletal muscles (Sommerich et al. (2000)). The main challenge in implementing controlled motion for prosthesis is correctly predicting the user's motion intention. SEMG signals have been used in an effective way in prosthesis control systems (Merletti & Parker (2004); Parker et al. (2006)). The SEMG signal is very convenient for prosthesis control, because it is intrinsically related to the user's intention (Hudgins et al. (1993)). A myoelectric control algorithm should be capable of learning the muscular activation patterns that are used in natural form for typical movements. It also needs robustness against variations in conditions during the operation, and the response time cannot create delays that are noticeable to the user (Fukuda et al. (2003)). Pattern recognition of the SEMG signal allows discriminating amongst the desired classes of limb motion and plays a key role in advanced control of powered prostheses for amputees and for individuals with congenital deficiency in the upper or lower limbs. The success of a myoelectric control scheme depends greatly on the classification accuracy.

Electronic knees can be designed for providing different levels of damping during swing, and for adjusting to different walking speeds, assuming they have the appropriate sensors and control algorithms for estimating the knee joint angle and the walking speed. With the appropriate control algorithm, it is possible to program the prosthesis to allow the knee to flex and extend while bearing a subject's weight (stance flexion). This feature of normal walking is not possible with conventional prostheses. Electronic knees use some form of computational intelligence to control the resistive torque about the knee. Several research groups have been involved in designing prototype knee controllers. Grimes et al. (1977) developed an echo control scheme for gait control, in which a modified knee trajectory from the sound leg is played back on the contralateral side. Popovic et al. (1995) presented a battery-powered active knee joint actuated by direct-current motors, together with a finite state knee controller

that utilizes a robust position tracking control algorithm for gait control. A small number of companies have also developed electronic knees for clinical use. For example, the Otto Bock C-leg (Kastner et al. (1999)) provides adjustable resistance for flexion and extension in swing through onboard intelligence and a special software package. Figure 1 presents the ongoing development of an active leg prosthesis prototype. The prosthesis has three degrees of freedom: one for the knee (sagittal plane), and two movements for the foot (sagittal and frontal plane). The three degrees of freedom are controlled by direct-current reduction motors. Onboard sensors allow estimating the foot orientation with respect to the ground (Ishihara et al. (2009)). This allows the foot to be oriented according to the terrain.

A pattern-recognition-based myoelectric control algorithm is typically composed of various main modules; a data segmentation stage handles the data before feature extraction, to improve precision and response time. A feature extraction stage pre-processes the data for reducing the amount of information to be analyzed. New variables (features) may be obtained by linear or non-linear transformation of the original data. The central component is the neural network classifier, which must be capable of learning relations between the input features and the desired control outputs. Significant advancements in pattern recognition methodology are in progress. A common approach is to extract parameters from the data, such as time-domain features (e.g., mean absolute value, waveform length, number of zero crossings) (Kelly et al. (1990); Hudgins et al. (1993)), spectral parameters (e.g., auto-regressive model) (Huang et al. (2005);Hargrove et al. (2008)), time-frequency coefficients (e.g., short-time Fourier transform) (Englehart et al. (2001)), and/or time-scale coefficients (e.g., discrete wavelet transform, wavelet packet decomposition) (Englehart et al. (2001); Chu et al. (2005); Wang et al. (2006)). Further data reduction may be achieved using a feature projection stage between pre-processing and classification (Englehart et al. (2001); Chu et al. (2005); Wang et al. (2006)). This approach eliminates redundant information, which speeds up the training process. It may also help mapping the data into small and well-separated clusters, by absorbing signal variations and noise present in the data's original vector-space.

The data processing and classification techniques discussed above have been successfully used for myoelectric control by several groups. Kelly et al. (1990) proposed an algorithm capable of discriminating between four motions of elbow and wrist joints from SEMG patterns measured from one pair of electrodes, using a Hopfield neural network for time-domain feature extraction, followed by a two-layer perceptron neural classifier. Englehart et al. (2001) designed an algorithm for dexterous and natural myoelectric control of powered upper limbs using a linear discriminant analysis (LDA) classifier, after principal component analysis (PCA) dimensionality reduction on a wavelet-based feature set. Chu et al. (2005) presented a pattern recognition algorithm for the control of a multifunction myoelectric hand, using the wavelet packet transform for feature extraction, a multilayer neural network classifier, and a linear-nonlinear feature projection composed of PCA and self-organizing feature maps, respectively. Huang et al. (2005) designed a classification scheme based on Gaussian mixture models for myoelectric control of upper limb prostheses, using feature extraction based on time-domain statistics, auto-regressive (AR) coefficients, and the root mean square of the signal. Wang et al. (2006) proposed an algorithm capable of discriminating between four types of hand and forearm movements, using wavelet packet feature extraction and PCA feature projection. Zhao et al. (2006) designed a control algorithm capable of recognizing six different hand motion patterns, using a multi-layer perceptron neural network classifier and feature extraction based on sample entropy, time-domain filtering, and wavelet transform coefficients. Hargrove et al. (2008) used a combination of time-domain features and AR coefficients with

a LDA classifier to determine the effect of electrode displacements on pattern classification accuracy, and to design a classifier training strategy to address this issue.

The application of multisensor data fusion has found widespread use in diverse areas (industry, commerce, local robot guidance for global military defense, etc.) (Luo et al. (2002)). Data fusion is the continuous process of implementing a model of the domain of interest, utilizing data of different natures (Gao & Harris (2002)). The purpose of data fusion is to produce an improved model or estimate of a system from a set of independent data sources. The use of range sensory data allows automatic extraction of information about the sensed environment under different operating conditions, and increases the performance, reliability, data rates and autonomy of the system (Luo (1996);Hall & Llinas (1997);Dasarathy (1997)). In many real-time applications, the desired model is the state vector of the dynamic process (Ferreira et al. (2005);Delis et al. (2009a)). The combination of the information from the sensors and subsequent estimation of the state should be done in a coherent manner, such that the uncertainty is reduced. The Kalman filter is a state estimator algorithm widely used for optimally estimating the unknown state of a linear dynamic system from Gaussian distributed noisy observations (Manyika & Durrant-Whyte (1994)). The algorithm uses a predefined model of the system to predict the state at the next time step (Diniz (1997)). The fusion of SEMG signals with other data is not common in the literature.

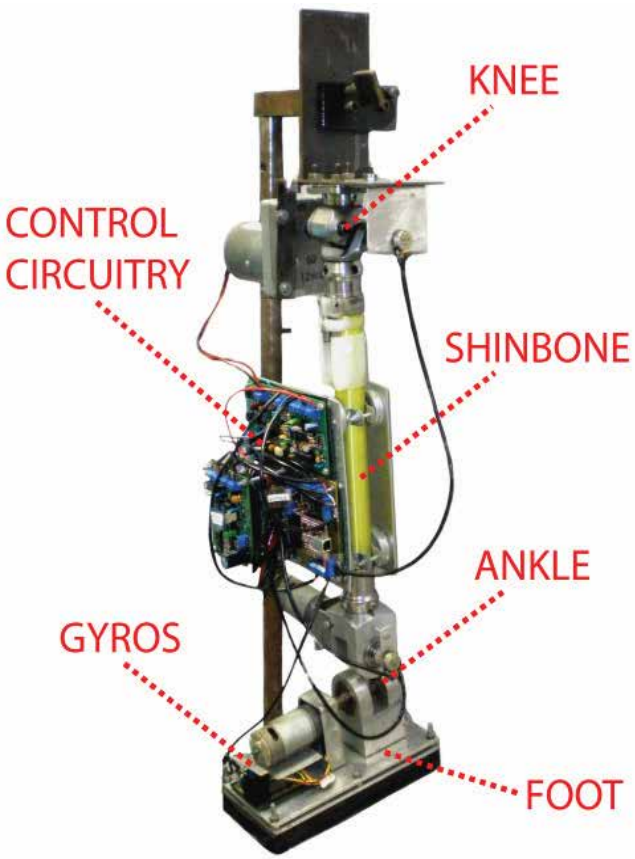


Fig. 1. Prototype of a leg prosthesis for transfemoral amputees.

Silva et al. (2003) applied data fusion of mechanomyography signals for the generation of binary control signals for an electrically powered prosthesis. The goal was to implement a practical mechanomyography-based detection system of muscle contractions for prosthesis control. Silicon-embedded microphone-accelerometer sensor pairs were used to record the mechanomyographic signals. A multisensor data fusion strategy for generation of binary control signals, based on the root-mean-square (RMS) values of the segmented signals, was trained and used as a detector (Silva et al. (2003)). Accuracies of 95% and 86% were achieved in the detection of contraction signals from the wrist extensors and flexors, respectively. Lopez et al. (2009) proposed two strategies for data fusion based on variance weighted average and decentralized Kalman filter, by means of an arrangement of redundant potentials, that is, by combining the SEMG signals. The muscle contraction amplitude was estimated and transformed to angular reference for the control of the robot joint. The algorithms demonstrated an efficient performance, and the joint never moved beyond its safety range (Lopez et al. (2009)).

Despite great success in decoding discrete movements such as individual finger flexion or extension, the matter of continuously predicting joint angles using SEMG signals is comparatively underdeveloped (Smith et al. (2008)). Increasing the number of SEMG channels that are acquired and processed may provide the user with higher accuracy in controlling the intensity of the contraction (Englehart et al. (2001)). However, as the number of inputs increases, the complexity of the network structure grows exponentially, which significantly increases the convergence time and the system response time. This chapter presents a feature extraction and pattern classification algorithm for estimating the intended knee joint angle from a two-channel SEMG signal, acquired using surface electrodes placed on the upper leg. This algorithm was designed for myoelectric control of an active transfemoral prosthesis (Cascão et al. (2005);Rodrigues et al. (2006)), as an improvement to the algorithm proposed by Ferreira et al. (2005). The proposed method improves the feature extraction stage by using a combination of spectral and temporal domain approaches – AR coefficients (Huang et al. (2005);Hargrove et al. (2008)) and signal amplitude histogram (Zardoshti-Kermani et al. (1995);Liu et al. (2007)), respectively – and by incorporating a feature projection stage, using a self-organizing map (SOM) (Kohonen (2001)). The incorporated Kohonen network reduces the dimensionality of the data at the input of the Levenberg–Marquardt (LM) neural classifier (Hagan & Menhaj (1994)), by mapping all the AR and histogram coefficients into a two-dimensional vector space (Chu et al. (2005)).

The accuracy of knee joint angle estimation algorithms based exclusively on pattern-recognition of SEMG signals may be greatly reduced by problems such as the required high level of amplification (due to the low level of the SEMG signals), motion of the sensor cables and/or noise caused by the power supplies (Merletti & Parker (2004)). These issues make myoelectric control rather sensitive. This motivates the use of other type of sensors on the prosthesis, which may potentially allow parameter adaptation during the use of the prosthesis by the patient. For example, micro-electromechanical gyroscopes and joint motion sensors may be used for measuring the angular velocity of the knee joint. The integration of these data can be used to obtain an estimate of the knee joint angle, which can be used to make small corrections of the neural network's coefficients in real-time. Fusion of the SEMG signals with proprioceptive sensor data could also improve the precision of the prosthesis control during movement and provide a more reliable myoelectric control (Oskoei & Hu (2007)).

This chapter presents various algorithms that use SEMG signals and proprioceptive sensors

for continuous estimation of the knee angle for control of active transfemoral prostheses. The next section presents the experimental protocol for SEMG signals and proprioceptive sensor data acquisition. Section 3 presents a knee angle estimation algorithm based exclusively on SEMG signal processing and analysis. Section 4 presents three algorithm variants based on data fusion of SEMG data and proprioceptive sensor (gyroscope) data. Section 5 proposes a performance comparison between the proposed algorithms. Results, discussions and conclusions are presented in sections 6, 7 and 8, respectively.

2. Experimental protocol and data acquisition

Myoelectric signal acquisition was performed using the microcontrolled bioinstrumentation system described by Delis et al. (2009b) and shown in Figure 2a. The system acquires two channels of amplified SEMG signals, the angular displacements signal and the data from the gyroscope sensors, using a 13-bit analog-to-digital converter, which is electrically isolated from the microcontroller and from the power supply using an optocoupler and a DC-DC converter. The sampling rate was 1043.45 Hz per channel. Analog filters are used to limit the SEMG signals to the 20–500 Hz frequency range (SENIAM (2008)). The microcontrolled system implements a digital real-time adaptive notch filter, which maintains a running estimate of the 60 Hz power line interference (Ahlstrom & Tompkins (1985)). The data is transferred to a personal computer through a serial interface. The experimental protocol was approved by the research ethics committee of the University of Brasilia (process no. 079/09, group III). Twelve able-bodied volunteers were studied and provided informed consent in accordance with institutional policy. Two pairs of 10-mm Ag/AgCl surface electrodes were placed in bipolar configuration over a pair of antagonist muscles (Figures 2b and 2c). These muscles correspond to the flexion and extension movements of the knee joint, respectively. The SEMG electrodes were attached to the skin over the muscle such that the longitudinal axes of the electrodes were parallel to the longitudinal axes of the muscle. The distance between the centers of the electrodes from each pair was 2–3 cm, according to the SENIAM protocol recommendations (SENIAM (2008)). The reference electrodes were placed over the lateralis and medialis epicondyle bones. An electrogoniometer was placed and strapped over the external side of the leg, and the gyroscope sensors were placed over the upper and lower legs, respectively (Figure 2d). The difference between the signals measured by the gyroscopes reflects the angular rate of the knee joint. Each of the twelve subjects was studied over the course of five days. Four 15-second measurements were performed on each day, with 5-minute rest periods between measurements. For each measurement, the subject was asked to walk in particular directions at a constant pace. Some variability in pace was observed between measurements. The first and third measurements from each day were used for training the algorithm's neural network, and the second and fourth measurements were used for testing the algorithm. Figure 3 presents simultaneously-acquired SEMG and proprioceptive signals from a representative subject. A total of 240 measurements were obtained, with half of them being used for algorithm training and the other half being used for algorithm testing.

3. Knee angle estimation based exclusively on SEMG data

The first proposed algorithm provides knee angle estimates based exclusively on information extracted from the electromyographic signals (Figure 4). The algorithm is composed of three main stages: (i) feature extraction, using a combination of spectral and temporal

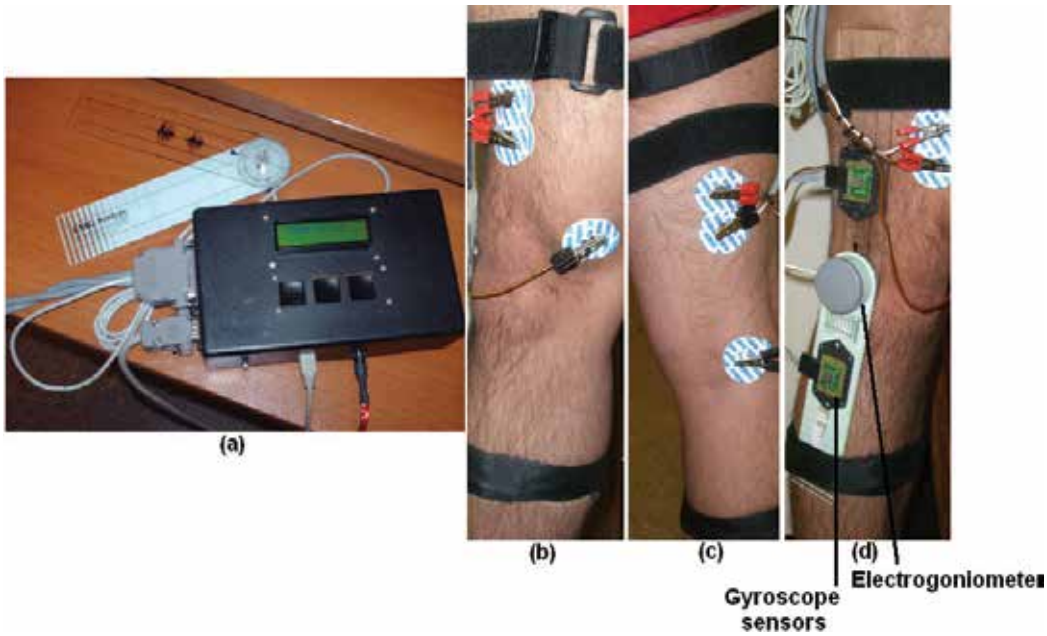


Fig. 2. Bioinstrumentation system (a) and placement of SEMG electrodes (b,c), electrogoniometer and gyroscope sensors (d).

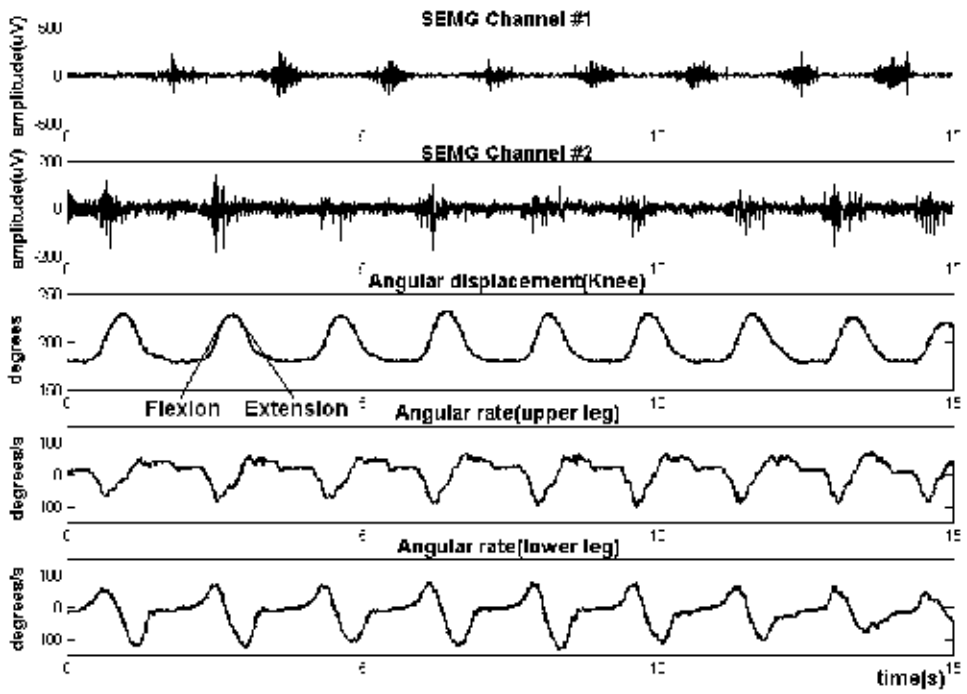


Fig. 3. Representative set of simultaneously-acquired SEMG signals, electrogoniometer angle and gyroscope measurements.

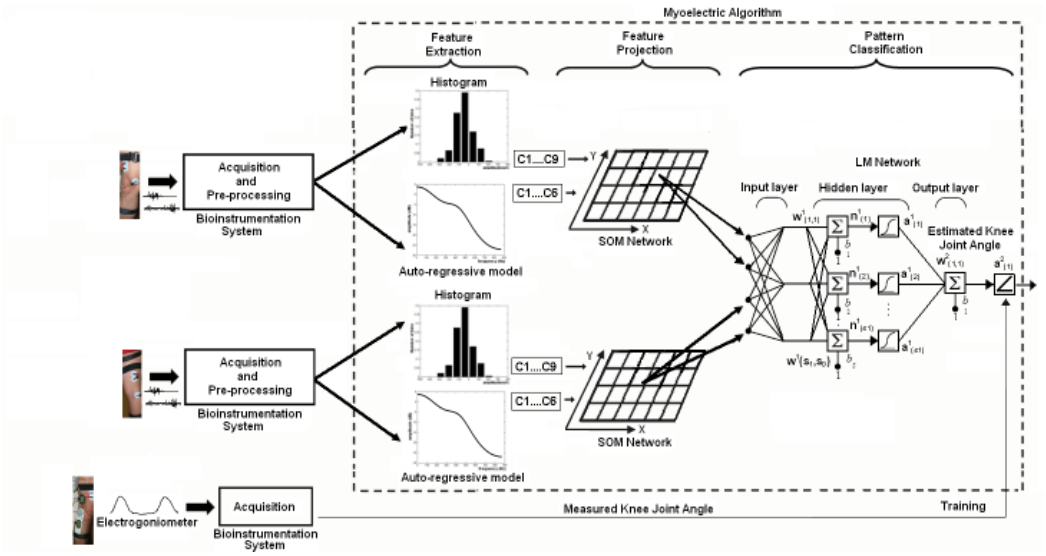


Fig. 4. Block diagram of the proposed knee joint angle estimation algorithm based exclusively on SEMG data.

domain approaches (AR coefficients and signal amplitude histogram, respectively); (ii) feature projection, using a self-organizing map; and (iii) pattern classification, using a Levenberg–Marquardt multi-layer perceptron neural network. Feature extraction and projection is performed independently for each SEMG channel. Data from the electrogoniometer is used as reference during network training, and is not used by the network during testing. Each of these stages is discussed in detail below, followed by a discussion on the approach for training the cascade networks.

3.1 Feature extraction

Presenting the myoelectric signal directly to a neural classifier is impractical, because of the dimensionality and random characteristics of the signal. The signal needs to be represented by a vector of reduced dimensionality, capable of representing the signal's information in a more compact fashion. Such vector is called a feature vector. In this work, the feature vector is composed of two sets of coefficients: the amplitude histogram bin counts, representing the time-domain characteristics of the SEMG signal, and the auto-regressive coefficients, representing the spectral content of the signal. The auto-regressive model is a convenient structure for model identification, in which the spectral envelope of the signal is modeled as an all-pole transfer function. The coefficients of this transfer function (the AR coefficients) contain information about the frequency content of the signal. In this work, the AR coefficients are used to compactly represent the spectral features of the SEMG signal (Huang et al. (2005); Hargrove et al. (2008)). The coefficients are calculated using the recursive least squares algorithm with a forgetting factor (Vaseghi (2000)). This gives more weight to the most recent samples at the moment of the iteration, which allows the algorithm to track temporal variations of the signal. The parameters are calculated recursively (Ljung (1987)) as presented

below:

$$\hat{\eta}_k = \hat{\eta}_{k-1} + L_k \left[y_k - \varphi_k^T \hat{\eta}_{k-1} \right], \quad (1)$$

$$P_k = \left[P_{k-1} - \frac{P_{k-1}\varphi_k \varphi_k^T P_{k-1}}{\lambda_k \varphi_k^T P_{k-1} \varphi_k} \right] \frac{1}{\lambda_k}, \quad (2)$$

$$L_k = \frac{P_{k-1}\varphi_k}{\lambda_k + \varphi_k^T P_{k-1} \varphi_k}, \quad (3)$$

where $\hat{\eta}_k$ are the vector coefficients that are estimated at discrete time k; φ_k is the regressors vector, P_{k-1} is the error covariance matrix and L_k is the gain vector of the filter. The forgetting factor λ_k controls the system response time. Based on literature (Huang et al. (2005);Ferreira et al. (2005)) and on an evaluation using the Akaike criterion (Ljung (1987)), we concluded that an AR order of four to six is sufficient for efficiently representing the SEMG signal. Thus, a sixth-order AR model was used, with a forgetting factor $\lambda_k = 0.995$, which is equivalent to 200 samples, or 192 ms. The coefficient estimated at instant k can be interpreted as a characteristic of the SEMG signal within the time interval specified by the forgetting factor, and it is a way of determining the angular displacement that the patient is trying to impose to the prosthesis (Ferreira et al. (2005)). The coefficients form a feature vector for the pattern classification process. This procedure is initialized with $P_0 = \mathbb{I}$ and $\hat{\eta}_0$ being a null vector.
The SEMG amplitude histogram is an extension of the zero crossing and the Willison amplitude measures (Zardoshti-Kermani et al. (1995)). The amplitude histogram provides a measure of the regularity in which the SEMG signal reaches each level of amplitude, associated with different histograms bins. Myoelectric signals reach relatively higher levels during the contraction period (compared to the base line amplitude), thus the amplitude histogram is capable of providing useful information about the state of a joint (Zardoshti-Kermani et al. (1995)). A histogram with nine symmetrically and uniformly distributed bins was used in this algorithm. The range of values was set based on the maximum and minimum SEMG amplitude levels measured on the training datasets. The window length was set to 200 samples (192 ms). Both the histogram window and the AR coefficients are updated for every new SEMG sample. This produces a more dense but semi-redundant stream of class decisions that could potentially be used to improve response time and accuracy (Englehart & Hudgins (2003)).

3.2 Feature projection

The feature extraction stage reduces the dimensionality of the data to fifteen (nine histogram bins, and six AR coefficients). The feature projection stage further reduces the dimensionality of the feature vector, by mapping it into a two-dimensional space using a self-organizing map. SOM neural networks (Kohonen (2001)) are trained using unsupervised learning, and are capable of arranging the input data into a discretized two-dimensional space (a map), which attempts to preserve the topological properties of the input space. The SOM is composed of nodes (or neurons). A position in the map space and a weight vector (of the same dimension as the input data vectors) are assigned to each node. The mapping algorithm consists in finding the node with the weight vector that is the closest to the input vector. The output of the SOM network is the two-dimensional coordinate of the winning node. To find the output neuron (winning node), the following steps are used, according to the learning rule of the Kohonen feature map (Haykin (1999)), applied to a SOM with N nodes trained with feature inputs x:

Step 1: Choose random values for the initial weight vectors $w_j(0)$.

Step 2: Find the winning neuron y_c at time step t (similarity matching), by using the minimum-distance Euclidean criterion:

$$y_c = \arg_{w_j(t)} \min \left\| x(t) - w_j(t) \right\|, j = 1,2,\ldots N \tag{4}$$

Step 3: Update the synaptic weight vectors of all neurons by using the following update rule:

$$w_j(t+1) = w_j(t) + \rho(t) h_{j,y_c}(t) \left[x(t) - w_j(t) \right] \tag{5}$$

where $\rho(t)$ is the learning rate, and $h_{j,y_c}(t)$ is the neighbor function centered around the winner y_c. $\rho(t)$ and $h_{j,y_c}(t)$ are changed dynamically during the learning stage, in order to obtain optimal results.

Step 4: Go back to Step 2 until no changes in the feature map are observed.

Each of the two SOM maps (one for each SEMG channel) is arranged in a topological net with 100 neurons in their interconnection structure (10×10 matrix). The dimension of the network was chosen empirically, based on experimentation. The initial learning rate was 0.9, and the time constants τ_1 and τ_2 were 1431 and 1000 iterations, respectively (Haykin (1999)). The neighborhood function initially contains all the neurons of the network, centered around the winning neuron, and with time it gradually decreases in size. Thus, the initial size of the neighborhood function is equal to the radius of the lattice (i.e., 5). At the output of the feature projection stage, the information in each of the SEMG channels is represented by only two coefficients, i.e., a 2D coordinate, resulting in a total of four coefficients at the input of the pattern classification stage. Different coordinate pairs represent different points of operation associated with the movement of the knee joint during a walk.

3.3 Pattern classification

The pattern classification stage is responsible for providing an estimate of the knee joint angle from the set of four SOM coefficients obtained from the feature projection stage. This is performed using a Levenberg–Marquardt multi-layer perceptron neural network (Hagan & Menhaj (1994)). There has been considerable research on methods to accelerate the convergence time of multi-layer feed-forward neural networks, such as methods that focus on standard numerical optimization techniques, including the conjugate gradient algorithm, quasi-Newton methods, and nonlinear least squares (Battiti (1992); Charalambous (1992)). The method used in this chapter is an application of a nonlinear least squares algorithm to the batch training of multi-layer perceptrons, called Levenberg–Marquardt algorithm. The LM algorithm is very efficient for training moderate-sized feed-forward neural networks (Hagan & Menhaj (1994)). Although the computational requirements of the LM algorithm become much higher after each iteration, this is fully compensated by its higher efficiency. This is especially true when high precision is required. Similarly to the quasi-Newton methods, the LM method was designed to approach second-order training speed without computing the Hessian matrix. The key step in the LM algorithm is the computation of the Jacobian matrix, which can be computed through standard backpropagation techniques (Hagan & Menhaj (1994)), which are much less complex than computing the Hessian matrix.

The LM network used in our algorithm has three layers in its structure, with four input nodes (output vectors of the SOM networks) in the first layer, six nodes in the second layer (associated with tangential functions), and one node in the output layer (associated with a linear function). This structure was chosen empirically, based on experiments aimed at

minimizing the mean squared error (MSE). The node in the output layer represents the estimated knee joint angle (Figure 4).
The cascade networks (SOM and LM) were trained independently for each set of 15-second two-channel SEMG test signals, using its correspondent set of training signals and electrogoniometer measurements. First, the histogram and AR coefficients associated with each sample of each of the two SEMG signals were calculated. Then, these coefficients were used in the SOM networks' unsupervised training process to configure the topological map structures and set the weight vector of each neuron. Then, the same feature vectors were used into the trained SOM maps, in order to generate two-dimensional vectors to be used for training the LM network. During LM network training, the outputs from the trained SOM network were used as inputs, and the corresponding angular displacement measurements from the electrogoniometer were used as the target outputs. The same initial weight values were used for all three network layers (zero for all neurons). The maximum number of iterations was set to 50, the MSE stop criterion was 10^{-10} n.u.^{2}, and the initial learning rate was 1.0. These values were empirically chosen, aiming at maximum reduction of the final MSE.

4. Knee angle estimation based on fusion of SEMG and proprioceptive sensor data

Angular rate information may be extracted from gyroscope sensor data using a Kalman filter. This approach was evaluated in three myoelectric algorithm variants. The angular rate information is used to correct the estimation of the intended knee joint angle by fusion with the SEMG features. The three algorithms are composed of a feature extraction stage, a pattern classification stage and variations of a data fusion stage.

4.1 Feature extraction

For this data fusion approach, the set of features is obtained from Cepstral coefficients extracted from SEMG signals. Cepstral analysis is used for frequency-domain SEMG signature discrimination. The cepstrum of a signal is defined as the inverse Fourier transform of the logarithm of the squared magnitude of the Fourier transform of a signal (Kang et al. (1995)). If all transfer function poles are inside the unit circle, the logarithmic transfer function can be represented as a Laurent expansion (Kang et al. (1995)). Hence, the following recursive relation may be used to calculate cepstral coefficients from AR coefficients:

$$\begin{aligned} c_1 &= -a_1 \\ c_i &= -a_i - \sum_{n=1}^{i-1} \left(1 - \frac{n}{i}\right) a_n c_{i-n}, \quad i = 2, ..., P. \end{aligned} \tag{6}$$

Using (6), the first P cepstral coefficients (c_i, $i = 1, ..., P$) can be obtained from the coefficients (a_k) of a Pth order AR model, estimated as in Section 3.1. Some works have reported that the AR-derived cepstrum feature has better performance than the unprocessed AR feature (Kang et al. (1995) ; Chiou et al. (2004)). Even though the cepstral coefficients are derived directly from the AR coefficients, they do not contain exactly the same information, because the recursive operation changes the distribution of the features nonlinearly (Kang et al. (1995)). In this work, the cepstral coefficients were obtained using a sixth-order AR model and (6).
A second approach for feature extraction is implemented using the entropy of the myoelectric signal, calculated and used as a time-domain feature vector (Ito et al. (2008)). We focus on the difference in entropy between the stationary SEMG signal in a relaxed state and

during motion. Assuming that electromyographic signals can be approximated by a normal distribution process with zero mean, the entropy of the distribution in a M-sample time window is computed as

$$H(\sigma_i) = \frac{1}{2}\log(2\pi\sigma_i^2), \tag{7}$$

$$\sigma_i^2 = \frac{1}{M-1}\sum_{m=0}^{M-1} x_i^2(k-m), \tag{8}$$

where σ_i^2 represents the signal variance estimated from the signal measured from each electrode and x_i is the SEMG signal from the i-th electrode (Ito et al. (2008)). For each SEMG channel, the calculated entropy is concatenated with the cepstral feature vector. This combination provides robustness in weak SEMG signals.

4.2 Pattern classification

The pattern classification stage is implemented using a LM neural network, just as described in Section 3.3.

4.3 Data fusion strategies

Three data fusion strategies for estimating the intended knee joint angle were evaluated: (i) data fusion implemented during pattern classification, which is performed on both SEMG features and estimated angular rate; or (ii, iii) data fusion performed after pattern classification, which is performed on the SEMG features only. These strategies are presented next.

4.3.1 First data fusion strategy

Figure 5 presents the block diagram for the proposed knee angle estimation algorithm based on the first data fusion strategy. The use of angular rate information from the gyroscopes improves angle estimation precision and reduces estimation artifacts. Feature extraction is performed using a Kalman filter. The goal of Kalman filters is the estimation of non stationary noisy signals, by minimizing the mean squared error, i.e., recursive least squares for stochastic models. The estimated signal is modeled using a state-space formulation, describing its dynamical behavior (Diniz (1997)), according to the following first-order linear stochastic model:

$$x(k) = x(k-1) + n(k) \tag{9}$$

$$y(k) = x(k) + v(k) \tag{10}$$

where $x(k)$ is the joint angular rate, $n(k)$ is the noise modeling the evolution of the joint angular velocity between two sampling intervals, $y(k)$ is the measured angular rate, obtained from subtracting the angular rate values measured on the upper and lower legs, respectively; and $v(k)$ is the measurement noise. It is assumed that $n(k)$ and $v(k)$ are zero mean, uncorrelated Gaussian distributions, with variances q^2 and r^2, respectively. When applying the Kalman filter to this model, the prediction process for each iteration cycle is expressed according to

$$\hat{x}(k|k-1) = \hat{x}(k-1) \tag{11}$$

$$P(k|k-1) = P(k-1) + q^2 \tag{12}$$

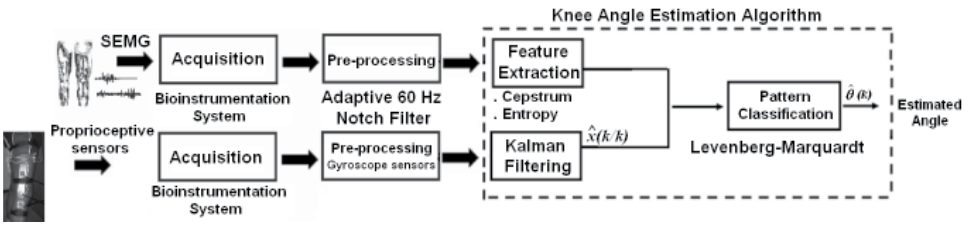


Fig. 5. Block diagram of the proposed knee angle estimation algorithm based on the first fusion strategy.

where $P(k|k-1)$ is the predicted error covariance matrix. The algorithm is initialized as $q^2 = 4$, $r^2 = 10$, $\hat{x}(0) = 0$ and $P(0) = 0.01$. These predictions are corrected, using the angular rate measure provided by the gyroscopes, $y(k)$, as follows:

$$G(k) = \frac{P(k|k-1)}{P(k|k-1)+r^2} \quad (13)$$

$$\hat{x}(k) = \hat{x}(k|k-1) + G(k)\left(y(k) - \hat{x}(k|k-1)\right) \quad (14)$$

$$P(k) = \left(\mathbb{I} - G(k)\right)P(k|k-1) \quad (15)$$

where $G(k)$ is the Kalman filter gain, and $\hat{x}(k)$ is an optimal estimate of $x(k)$ in the least-squares sense. It can be shown that, for this specific problem, this filter is equivalent to a unity-gain, low-pass, first-order filter with time-varying cut-off frequency. This cut-off frequency is computed considering noise variances q^2 and r^2, as well as the error variance associated with $\hat{x}(k)$ (Diniz (1997)). The value of $\hat{x}(k)$ is an optimal estimate of the mean of the knee joint angular rate at sampling step k. Thus, at each time instant k, the optimally filtered angular rate estimate $\hat{x}(k)$, along with the SEMG cepstral and entropy coefficients are used as inputs to the neural classifier (Figure 5).

4.3.2 Second data fusion strategy

The second data fusion strategy is based on information fusion in the correction process of a Kalman filter. This may reduce the perturbations that are generated on the angle estimation process from the neural network. This data fusion strategy is presented in Figure 6. In this strategy, the feature vectors obtained from feature extraction are used as inputs to the LM neural network. The estimated knee joint angle is modeled using a state-space formulation, describing its dynamical behavior (Diniz (1997)), according to the following linear stochastic model:

$$x(k) = x(k-1) + Tu(k) + n(k) \quad (16)$$

$$y(k) = x(k) + v(k) \quad (17)$$

where $x(k)$ now represents the knee angle, $u(k)$ is the measured angular rate acquired with a sampling period T, obtained from subtracting the angular rate values measured on the upper and lower legs, respectively. $n(k)$ is noise modeling the evolution of the knee joint angle between two sampling intervals. $y(k)$ is the measured knee joint angle obtained from the LM neural network output, and $v(k)$ is the associated measurement noise. It is assumed that $n(k)$ and $v(k)$ are zero mean, uncorrelated Gaussian distributions, with variances q^2 and r^2, respectively. When applying the Kalman filter to this model, the prediction process for each

iteration cycle is expressed according to

$$\hat{x}(k|k-1) = \hat{x}(k-1) + Tu(k) \quad (18)$$

$$P(k|k-1) = P(k-1) + T^2\sigma_u^2(k) + q^2 \quad (19)$$

where $\sigma_u^2(k) = 25\ \text{deg}^2\ /\ \text{sec}^2$ is the variance of the measured angular rate information $u(k)$. q^2, r^2, $\hat{x}(0)$ and $P(0)$ have the same values as in the previous strategy, and the correction process is expressed by the same equations, (13)-(15). However, the value $\hat{x}(k)$ is an optimal estimate of the knee joint angle from the fusion process at each time instant k.

4.3.3 Third data fusion strategy

A third variant is a modification of the previous strategy. This variant introduces a compatibility test based on the Mahalanobis distance (De Maesschalck et al. (2000)). The Mahalanobis distance is a useful way of determining similarity of sample sets, as it is not dependent on the scale of the measurements. The Mahalanobis distance is computed between the prediction and correction process of the Kalman filter (Figure 7). The objective is to detect possible artifacts that come from the estimated angle at the LM neural network output, on each time step of the data fusion process.

When the Kalman filter is applied to the linear stochastic model described by equations (16) and (17), the prediction process for each iteration cycle is described by equations (18) and (19). The Mahalanobis distance is calculated between the estimated knee angle $y(k)$ from the LM neural network and the predicted knee angle $\hat{x}(k|k-1)$, based on the following equations:

$$d^2(k) = \frac{(y(k) - \hat{x}(k|k-1))^2}{P(k|k-1) + r^2}. \quad (20)$$

It can be shown that $d^2(k)$ is χ_1^2 distributed. Thus, $y(k)$ and $\hat{x}(k|k-1)$ are said to be statistically compatible if $d^2(k) \leq 3.81$, according to the 95% confidence threshold obtained from the chi-square table. In such a case, $y(k)$ is used to correct $\hat{x}(k|k-1)$ using equations (13)-(15). If $d^2(k) > 3.81$, the filter uses the predicted values as estimates: $\hat{x}(k) = \hat{x}(k|k-1)$ and $P(k) = P(k|k-1)$, protecting the estimation process from possible angle estimation artifacts at the neural network, originated from SEMG signals.

4.4 Parameter setting for the myoelectric algorithms based on data fusion

Network training and testing were performed in Matlab (The MathWorks, Inc., Natick, MA, USA). For each SEMG channel, the proposed algorithms were implemented using 200 sample (192 ms) sliding windows for the feature extraction process (cepstral analysis and entropy).

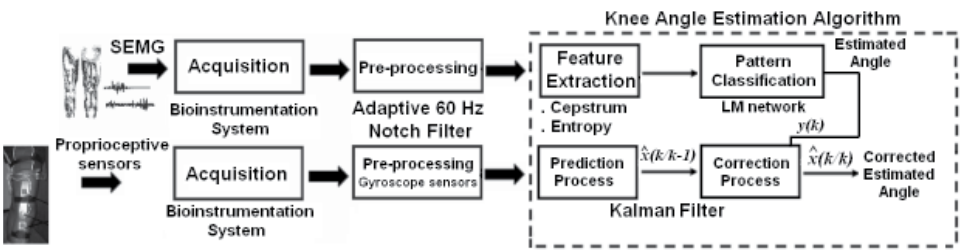


Fig. 6. Block diagram of the proposed knee angle estimation algorithm based on the second fusion strategy.

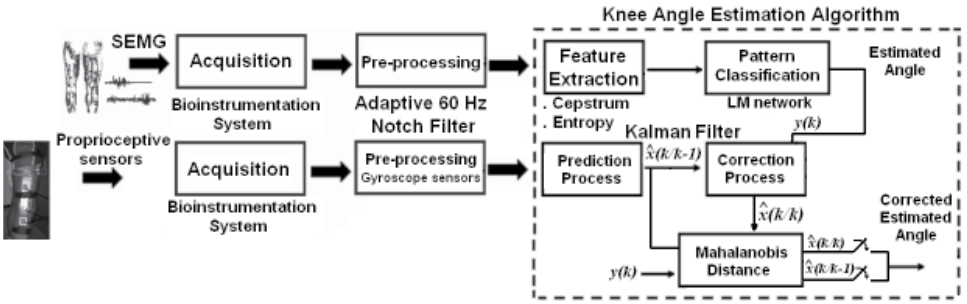


Fig. 7. Block diagram of the proposed knee angle estimation algorithm based on the third fusion strategy.

For each new pair of gyroscope sensor samples, estimates of updated Kalman filter angular rate (first proposal) and knee joint angle (second and third proposals) were calculated. This results in a 15-coefficient feature vector per sample interval (six cepstral coefficients and one entropy coefficient per SEMG channel, plus one angular rate coefficient) for the first proposal. For the second and third proposals, we obtained a 14-coefficient feature vector (6 cepstral coefficients and 1 entropy coefficient, per SEMG channel). In the three algorithms, the information is transferred to a LM neural network, with 15 (first proposal) or 14 (second and third proposals) nodes in the input layer, 6 nodes in the hidden layer, and 1 node in the output layer. The output node represents the estimated knee joint angle. The network architecture and size was empirically chosen, aiming at maximum reduction of the final MSE. The same initial weight values were used for all three network layers (zero for all neurons). The maximum number of iterations was set to 50, the MSE stop criterion was 10^{-10} n.u^2. and the initial learning rate was 1.0. The true displacement angle measured with the electrogoniometer was used as training reference.

5. Performance comparison between the proposed algorithms

For performance evaluation, the myoelectric algorithms are quantitatively compared using statistics metrics based on: (i) the error-to-signal percentage, (ii) the correlation coefficient and (iii) statistics of error events, including the number of error events, the maximum error event duration and the maximum error amplitude (Delis et al. (2009a)). The statistics were calculated for each set of SEMG signals, and the average and standard deviation of those parameters were calculated for each subject. For consistency, the same training process and test sets were used with the myoelectric algorithms based solely on SEMG signals and with those based on data fusion. The same sliding window length (192 ms) and step (1 sample) and the same AR order and forgetting factor configuration were used for all evaluated methods. 120 sets of SEMG, electrogoniometer and gyroscope data which were not used for training were used for comparing the methods. The performance of each algorithm was evaluated by comparing the knee angle estimated from the SEMG signals with the angular displacement values measured with the electrogoniometer.

A threshold was applied to the time-series to detect the error events (Delis et al. (2009a)). This threshold was empirically set to 10°. Each series of consecutive errors found to be above the threshold was considered an error event.

The Mahalanobis distance was calculated for each metric as a means of assessing the statistical difference between the proposed method based solely on the SEMG signals and the proposed

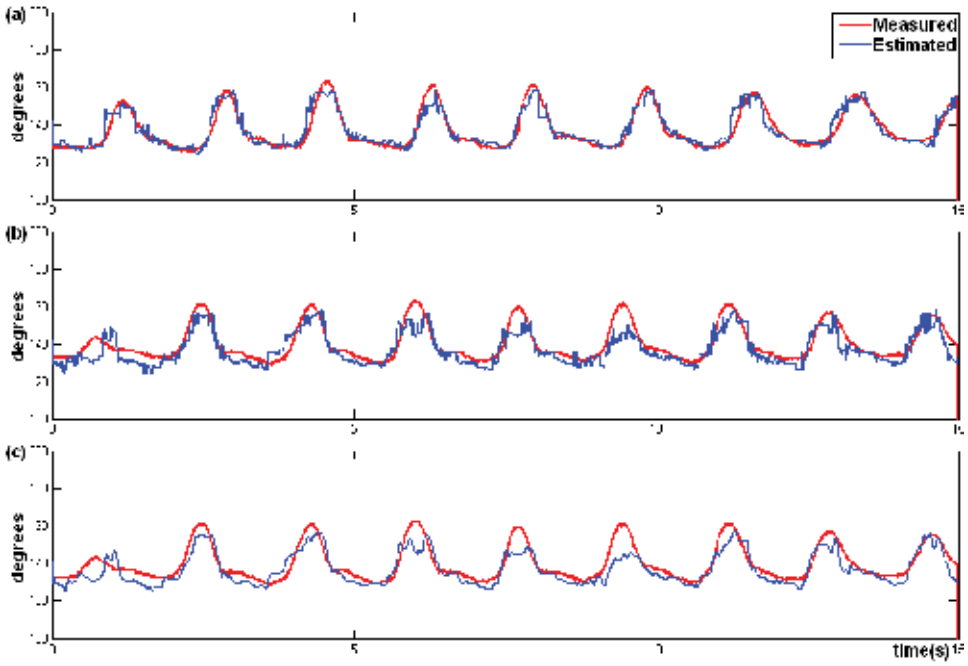


Fig. 8. Knee angle estimation results for the algorithm based exclusively on SEMG data, (compared with the electrogoniometer measurements) for two sets of signals from the same subject: (a) training results; (b) test results; (c) filtered test results.

methods based on data fusion (Delis et al. (2009a);Duda & Hart (2000)). For $N = 12$ datasets, the Mahalanobis distance between a same metric computed using two techniques, represented by m_1 and m_2, is given by

$$\begin{aligned} d^2_{m_1 m_2} &= \sum_{n=1}^{N} \frac{(m_1(n) - m_2(n))^2}{\sigma_1^2 + \sigma_2^2} \\ \sigma_i^2 &= \frac{1}{N-1} \sum_{n=1}^{N} \left(m_1(n) - \frac{1}{N} \sum_{n=1}^{N} m_i(n) \right)^2, \quad i = 1,2 \end{aligned} \tag{21}$$

where n means the n-th dataset. The metrics m_1 and m_2 are considered to be statistically similar (with 95% confidence) if $d^2_{m_1 m_2} \leq 21.03$, which is equivalent to $d_{m_1 m_2} \leq 4.58$.

6. Results

6.1 Testing process

Figure 8 presents two time-series of estimated knee joint angle from a subject, obtained during the training and testing processes, respectively, using the algorithm based exclusively on SEMG data. In the test results, a 50-tap (48 ms) moving average filter was used for reducing the estimation noise and the variance (Figure 8c). Such filtering removes jitter in the output signal, which could cause undesirable and unintentional motion of the prosthesis. The results were satisfactory, with a reduction of impulsive noise and maintaining the slope change in the estimated angle.

Figure 9 shows three time series for estimated knee angle by the proposed myoelectric algorithms based on data fusion. It can be noted that the three algorithm variants provide good tracking of the knee angle with respect to the measured angle, in spite of the occurrence of discrete artifacts. Such artifacts may be imperceptible for myoelectric control, because of the prosthesis' mechanical inertia.

6.2 Comparison between the proposed algorithms

Figure 10 presents a qualitative comparison between the myoelectric algorithms based exclusively on SEMG data and those based on data fusion. Measured and estimated angle displacements from a subject are shown for (a) the first proposal based solely on SEMG signals; (b) the first variant based on data fusion; (c) the second variant based on data fusion; and (d) the third variant based on data fusion. The absolute difference between measured and estimated angles is also shown.

Figure 11 presents a similar qualitative comparison, but in the presence of motion artifacts. The straps holding the electrode cables were intentionally left loose during this experiment, which caused motion artifacts in the SEMG signal. Measured and estimated angle displacements from a subject are shown for (a) the algorithm based exclusively on SEMG signals; (b) the first variant based on data fusion; (c) the second variant based on data fusion; and (d) the third variant based on data fusion. The absolute difference between measured and estimated angles is also shown. In spite of the occurrence of false positives during the knee joint angle estimation process with the proposed algorithms, the level of degradation of the estimate is lower with the algorithms based on data fusion. The second and third variant presents errors peaks that could be imperceptible for the movement of the leg prosthesis,

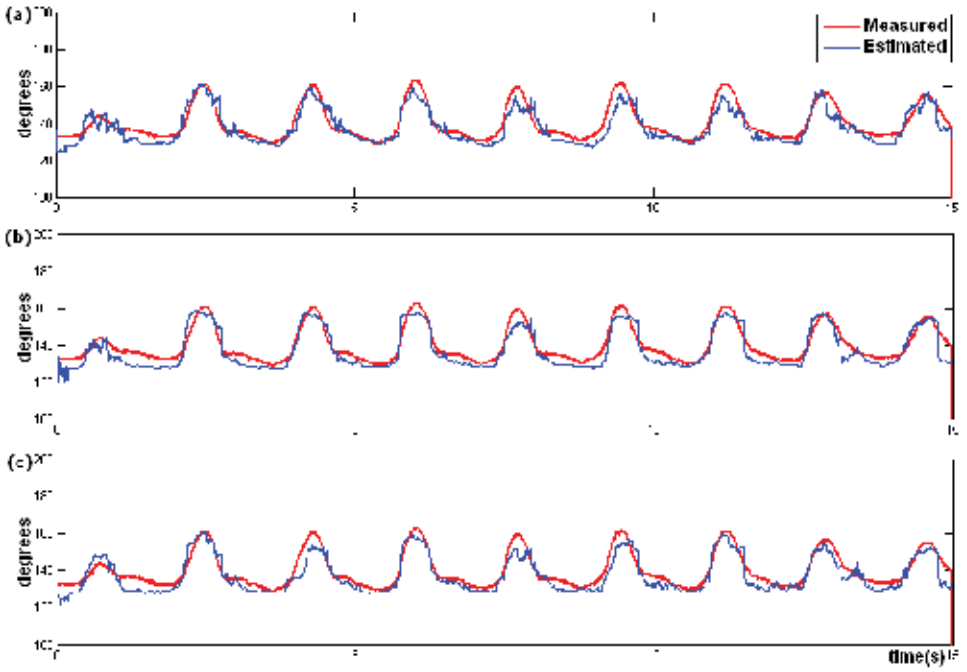


Fig. 9. Measured and estimated knee joint angle for the three proposed algorithms based on data fusion: (a) first variant; (b) second variant; (c) third variant.

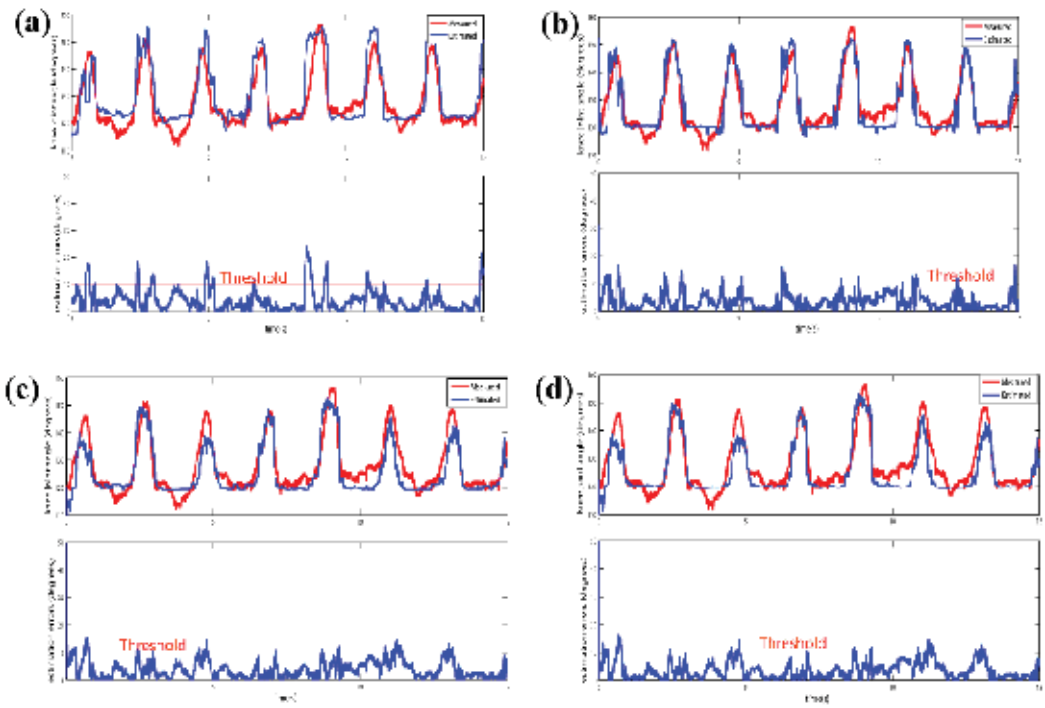


Fig. 10. Qualitative comparison between the proposed myoelectric algorithms. Measured and estimated displacement angle from a representative experiment and their absolute differences (estimation error) are shown for the following algorithms: (a) first proposal based solely on SEMG signals; (b) first variant based on data fusion; (c) second variant based on data fusion; and (d) third variant based on data fusion.

depending on their duration. The best results in the presence of motion artifacts were obtained with the second and third data fusion variants, in which the fusion process is implemented between the SEMG signals and the gyroscopes sensors on the correction process by Kalman filtering.

Another implemented test was the evaluation of the robustness against power line 60-Hz interference. A 60-Hz signal with an amplitude of 0.1 mV was added to each SEMG channel. This amplitude value was chosen for this test because this was the maximum 60-Hz interference level registered during the experiments. Figure 12 presents a qualitative comparison between the estimated and measured angles. Measured and estimated angle displacements from a subject are shown for (a) the first algorithm based exclusively on SEMG signals; (b) the first variant based on data fusion; (c) the second variant based on data fusion and (d) the third variant based on data fusion. The absolute difference between measured and estimated angles is also shown. It is observed that, in spite of the presence of discrete false positives, the estimated knee joint angle for the myoelectric algorithms is reasonably similar to the measured angle.

Table 1 presents the computed Mahalanobis distance – see equation (21) – between each method based on data fusion and the algorithm proposal based solely on SEMG signals. According to the threshold $d_{m_1 m_2} \leq 4.58$, only the first data fusion variant strategy presented statistically different results for the maximum error event amplitude metric. That means

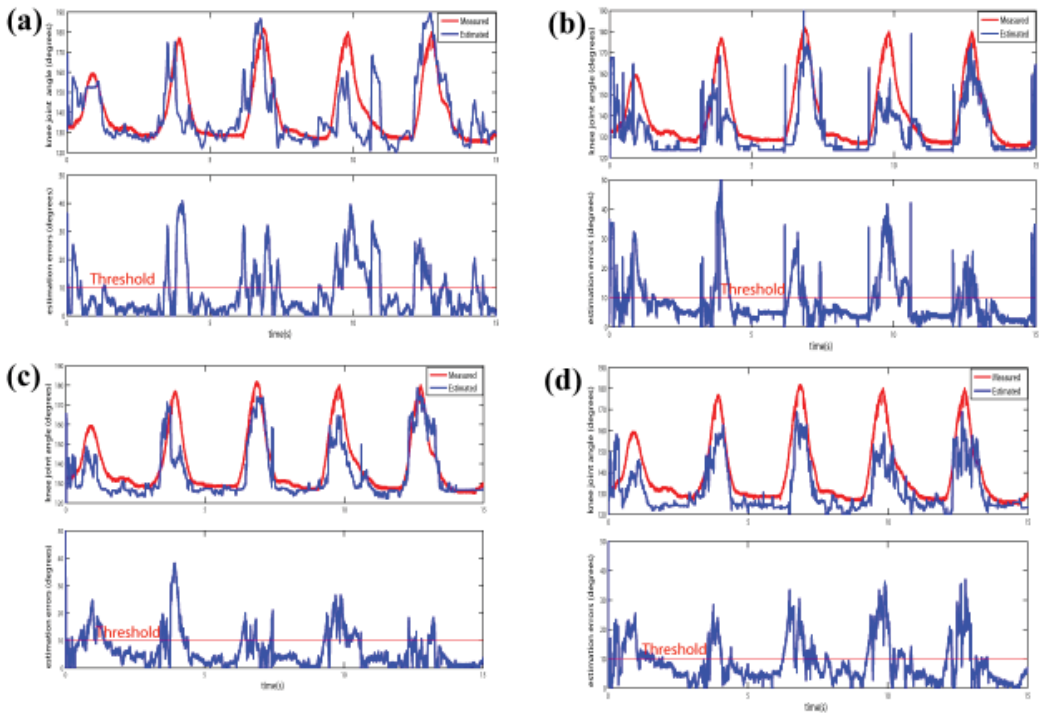


Fig. 11. Qualitative comparison between the proposed myoelectric algorithms. Measured and estimated angle displacements from measurements with movement artifacts and their absolute difference (estimation error), are shown for the following algorithms: (a) first proposal based exclusively on SEMG signals; (b) first variant based on data fusion; (c) second variant based on data fusion; (d) third variant based on data fusion.

that the second and third data fusion strategies present results which are similar in mean. However, the robustness of the third strategy with respect to SEMG artifacts is superior, as seen in the previous figures.

7. Discussion

The proposed myoelectric algorithms provide dimensionality reduction that makes possible the connection of a larger number of SEMG sensors without affecting the performance of the Levenberg–Marquardt multi-layer perceptron neural network. The algorithms based

Metric	First strategy	Second strategy	Third strategy
Error-to-signal percentage	1.61	1.00	1.53
Correlation coefficient	1.59	1.37	1.37
Number of error events	2.67	1.33	1.21
Maximum error event amplitude	17.30	1.27	1.07
Maximum error event duration	0.90	0.87	0.85

Table 1. Mahalanobis distance $d_{m_1 m_2}$ between each of the algorithm variants based on data fusion and the myoelectric algorithm based exclusively on SEMG signals.

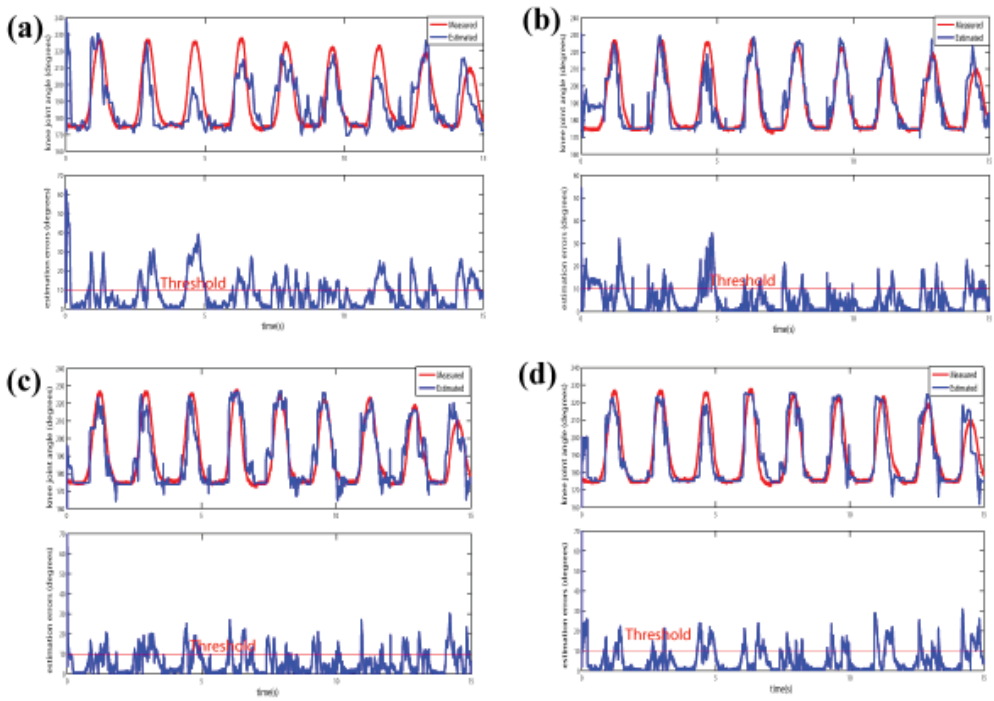


Fig. 12. Qualitative comparison between the proposed myoelectric algorithms. Measured and estimated angle displacements from measurements with added 60 Hz interference and their absolute difference (estimation error) are shown for the following algorithms: (a) algorithm based solely on SEMG signals; (b) first variant based on data fusion; (c) second variant based on data fusion; (d) third variant based on data fusion.

on data fusion makes possible the integration of different types of sensors, besides SEMG signals, using a Kalman filter. The access to additional information by the myoelectric algorithms during knee angle estimation improves precision and robustness for the prosthesis myoelectric control.

The maximum error amplitudes measured with the proposed methods are considerably reduced; however, they are still large (Figure 10, 11 and 12). Nevertheless, this may not be a significant issue, as short duration error events are unnoticeable to the leg prosthesis, due to the system's mechanical inertia. These short duration error peaks may be caused due to noise in the feature space, and/or by an insufficient number of neurons in the SOM network and in the LM network's hidden layer. This problem may be addressed by increasing the number of neurons, by increasing the number of SEMG signals, and/or adding other variables associated with leg proprioception (e.g., accelerometers). These approaches would result in increased computational network complexity and convergence time. Alternatively, error peaks may be avoided by increasing the forgetting factor of the recursive least squares AR algorithm and the window length of the histogram. However, this approach would increase the response time of the prosthesis. The accuracy of the proposed method in the presence of transient data may be improved using time-frequency and time-scale feature projection (e.g., wavelets, short-time Fourier transform) (Englehart et al. (2001)). However these approaches are more computationally intense than the combination of AR coefficients

with an amplitude histogram, as proposed in this work, and would also affect the networks' complexity. Furthermore, time-domain and AR features have been shown to outperform time-frequency features for stationary or slowly changing data, and to provide equivalent results for steady-state SEMG signals (Huang et al. (2005)).

The comparison of the first variant of the algorithm based on data fusion with the algorithm based exclusively on SEMG data showed a significant reduction on the maximum amplitude of error event. This emphasizes the fusion in the LM neural network of the information from the feature extraction process (cepstral coefficients and SEMG entropy) with the low-pass filtered angular rate information obtained from the Kalman filter. This fusion removes the noise on the estimated knee joint angle. It is expected that an increase in the amount of information supplied to the myoelectric algorithms (e.g., number of input channels) in the estimate process of the knee joint angle may result in improved precision for the control of the leg prosthesis.

The presence of artifacts due to movement of the electrode cables and 60-Hz interference during knee angle estimation may be interpreted by the leg prosthesis as false positives, depending of their duration (Figure 11b). The second and third variants based on data fusion, which use an optimal estimate of the knee joint angle obtained on the fusion process with the angular rate information at each time instant, are more robust than the myoelectric algorithms based solely on SEMG signals (Figures 11 and 12). The addition of other variables associated with leg proprioception (e.g., gyroscope sensors) may improve the precision and reduce artifacts in knee angle estimation, without significantly increasing the computational complexity of the myoelectric algorithm. However, the implementation of these algorithms involves an additional degree of complexity for obtaining the cepstral coefficients from the AR coefficients, in comparison with the first myoelectric algorithm proposal.

The first proposal is preferred for on-line implementation when the number of sensors is large and computational power is limited. However, considering the robustness aspect in the presence of movement artifacts, the second proposal based on data fusion is recommended.

Although the computational complexity of the Levenberg–Marquardt algorithm increases after each iteration during the training process, this is compensated by a gain in efficiency and a reduction in the network's convergence time. Hagan & Menhaj (1994) present comparisons between the Levenberg–Marquardt algorithm and modifications implemented in the back-propagation neural network, based in the conjugate gradient and variable learning rate. The results show failure in the convergence time for the evaluated modifications, while the same tests converged with acceptable results with the Levenberg–Marquardt algorithm. The results indicate that the LM algorithm is very efficient when it is trained with hundreds of neurons in their interconnection structure.

8. Conclusions

This chapter introduced a myoelectric algorithm based solely on SEMG data and three variants of myoelectric algorithms based on data fusion with the purpose of improving the knee joint angle estimation. The first proposal improves the algorithm originally presented by Ferreira et al. (2005), by adding a feature projection stage (a SOM network), and by incrementing the feature extraction stage with a signal amplitude histogram. Feature extraction now combines time-domain (histogram) and frequency-domain (AR coefficients) features. Pattern classification is still performed using a Levenberg–Marquardt multi-layer perceptron neural network, but this is now more efficient due to the dimensionality reduction provided by the SOM network.

The second proposal was based on three algorithm variants, which implement data fusion using Kalman filters. Through a prediction-correction formulation process, this provides an optimal estimate of the estimated knee angle, which is obtained by fusion of the information from gyroscope sensors using a Kalman filter. The myoelectric algorithms strategies present a feature extraction process based on cepstral coefficients and the entropy of the myoelectric signals (mixture of coefficients in frequency and time domains, respectively). A Levenberg–Marquardt multi-layer perceptron neural network is used for pattern classification. It was demonstrated that the fusion of SEMG signals with proprioceptive sensors reduces artifacts in the estimated joint angles.
The three algorithm variants based on data fusion present equivalent results when compared with the myoelectric algorithm based exclusively on SEMG signals, however, their performance is better in the presence of signal artifacts.
The concepts used in these algorithms may be useful in the development of a control algorithm for active leg prostheses, in which signals from many different sensors may be fused and used in the conception of a movement predictive model. We have demonstrated that it is possible to continuously decode knee position from SEMG signals collected from a generalized electrode placement in an able-bodied subject.

9. Acknowledgment

This work was partially supported by Brazilian Ministry of Education (MEC/CAPES), Brazilian Ministry for Science and Technology (MCT/CNPq), and Research and Graduate Council of the University of Brasilia.

10. References

Ahlstrom, M. L. & Tompkins, W. J. (1985). Digital filters for real-time ECG signal processing using microprocessor, IEEE Trans. Biomed. Eng., Vol. 32, pp. 708-713.

Battiti, R. (1992). First and second order methods for learning: between steepest decent and Newton's method, Neural Comput., Vol. 4, pp. 141-66.

Cascão, Jr. C. A.; Ferreira, R. U.; Beckmann, E. D.; Borges, G. A.; Ishihara, J. Y. & da Rocha, A. F. (2005). Estudo e desenvolvimento de uma prótese ativa de perna comandada por sinais eletromiográficos, Proceedings VII Simpósio Brasileiro de Automação Inteligente / II IEEE Latin-American Robotics Symp.

Charalambous, C. (1992). Conjugate gradient algorithm for efficient training of artificial neural networks, IEE Proc. G, Vol. 139, pp. 301-10.

Chiou, Y.; Luh, J.; Chen, S.; Lai, J. & Kuo, T. (2004). The comparison of electromyographic pattern classifications with active and passive electrodes, Medical Engineering and Physics, Vol. 26, No. 7, pp. 605-610.

Chu, J. U.; Moon, I.; Kim, S. K. & Mun, M. S. (2005). Control of multifunction myoelectric hand using a real time EMG pattern recognition, Proceedings IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 3957-62.

Dasarathy, B. V. (1997). Sensor fusion potential exploitation-innovative architectures and illustrative applications, Proceedings IEEE, pp. 24-38.

De Maesschalck, R.; Jouan-Rimbaud, D. & Massart, D. L. (2000). The Mahalanobis distance, Chemometrics and Intelligent Laboratory Systems, Vol. 50, pp. 1-18.

De Luca, C. J. (2006). Electromyography. Encyclopedia of Medical Devices and Instruments, John Wiley Publisher, pp. 98-109.

Delis, A. L.; Carvalho, J. L. A.; da Rocha, A. F.; Ferreira, R. U.; Rodrigues, S. S. & Borges, G. A. (2009a). Estimation of the knee joint angle from surface electromyographic signals for active control of leg prostheses, Physiological Measurements,Vol. 30, pp. 931-946.

Delis, A. L.; Carvalho, J. L. A.; da Rocha, A. F.; Nascimento, F. A. O. & Borges, G. A. (2009b), Development of a Myoelectric Controller Based on Knee Angle Estimation, Proceedings International Conference on Biomedical Electronics Devices, pp. 97-103.

Diniz, P. S. R. (1997). Adaptive Filtering Algorithms and Practical Implementation, Kluwer Academic Publishers.

Duda, R. O.; Hart, P. E. and Stork, D. G. (2000). Pattern Classification, John Wiley, 2nd edn, New York.

Englehart, K.; Hudgins, B. & Parker, P. A. (2001). A wavelet-based continuous classification scheme for multifunction myoelectric control, IEEE Trans. Biomed. Eng., Vol. 48, pp. 302-10.

Englehart, K. & Hudgins, B. (2003). A robust, real time control scheme for multifunction myoelectric control, IEEE Trans. Biomed. Eng., Vol. 50, pp. 848-54.

Ferreira, R. U.; da Rocha, A. F.; Cascão, C. A. Jr.; Borges, G. A.; Nascimento, F. A. O. & Veneziano, W. H. (2005), Reconhecimento de padrões de sinais de EMG para controle de prótese de perna, Proceedings XI Congresso Brasileiro de Biomecânica.

Fukuda, O.; Tsuji, T.; Kaneko, M. & Otsuka, A. (2003). A human-assisting manipulator teleoperated by EMG signals and arm motions, IEEE Trans. Rob. Autom., Vol. 19, pp. 210-22.

Gao, J. B. & Harris, C. J. (2002). Some remarks on Kalman filters for the multisensor fusion. Information Fusion, Vol. 3, pp. 191-201.

Grimes, D. L.; Flowers, W. C. & Donath, M. (1977). Feasibility of an active control scheme for above knee prostheses, Journal Biomechanical Engineering, Vol. 99, pp. 215-221.

Hagan, M. T. & Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm, IEEE Trans. Neural Networks, Vol. 5, pp. 989-93.

Hall, D. L. & Llinas, J. (1997). An introduction to multisensor data fusion, Proceedings IEEE, pp. 6-23.

Hargrove, L., Englehart, K. & Hudgins, B. (2008). A training strategy to reduce classification degradation due to electrode displacements in pattern recognition based myoelectric control, Biomed. Signal Process. Control, Vol. 3, pp. 175-80.

Haykin, S. (1999). Neural Networks: A Comprehensive Foundation, Prentice Hall 2nd edn, New Jersey.

Huang, Y.; Englehart, K.; Hudgins, B. & Chan, A. D. C. (2005). A Gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses, IEEE Trans. Biomed. Eng, Vol. 52, pp. 1801-11.

Hudgins, B.; Parker, P. A. & Scott, R. N. (1993). A new strategy for multifunction myoelectric control, IEEE Trans. Biomedical Eng., Vol. 40, No. 1, pp. 82-94.

Ito, K.; Tsukamoto, M. & Kondo, T. (2008). Discrimination of intended movements based on nonstationary EMG for a prosthetic hand control, Proceedings the ISCCSP, pp. 14-19.

Ishihara, J.Y.; Terra, M.H.; Borges, G.A.; Scandaroli, G.G.; Inoue, R.S.; Grassi Jr., V. (2009). Applications of Robust Descriptor Kalman Filter in Robotics. In: Victor M. Moreno ; Alberto Pigazo. (Org.). Kalman Filter: Recent Advances and Applications. Vienna: I-Tech Education and Publishing KG, pp. 507-534.

Kang, W. J.; Shiu, J. R.; Cheng, C. K.; Lai, J. S. & Tsao, H. W. (1995). The Application of Cepstral Coefficients and Maximum Likelihood Method in EMG Pattern Recognition, IEEE

Transactions on Biomedical Engineering, Vol. 42, pp. 777-785.
Kastner, J.; Nimmervoll, R. & Wagner, I. P. (1999). What are the benefits of the Otto Bock C-leg? A comparative gait analysis of C-leg, 3R45 and 3R80, Journal Medical Orthopedic, Vol. 119, pp. 131-137.
Kelly, M. F.; Parker, P. A. & Scott, R. N. (1990). The application of neural networks to myoelectric signal analysis: a preliminary study, IEEE Trans. Biomed. Eng., Vol. 37, pp. 221-30.
Kohonen, T. (2001). Self-Organizing Maps, New York: Springer, 3rd edn.
Liu, Y. H.; Huang, H. P. & Weng, Ch. H. (2007). Recognition of electromyographic signals using cascade kernel learning machine, IEEE/ASME Trans. Mechatron, Vol. 12, pp. 253-64.
Ljung, L. (1987). System Identification: Theory for the User, Prentice Hall, New Jersey.
Lopez, N. M.; di Sciascio, F.; Soria, C. M. & Valentinuzzi, M. E. (2009). Robust EMG sensing system based on data fusion for myoelectric control of a robotic arm, Biomedical Engineering Online, Vol. 8, No. 5.
Luo, R. C. (1996). Sensor technologies and microsensor issues for mechatronics system, IEEE/ASME Trans. Mechatron, Vol. 1, pp. 39-49.
Luo, R. C.; Yih, Ch. & Su, K. L. (2002). Multisensor Fusion and Integration: Approaches, Applications, and Future Research Directions, IEEE Sensor, Vol. 2, pp. 107-119.
Manyika, J. & Durrant-Whyte, H. (1994). Data Fusion and Sensor Management:A Descentralized Information – Theoretic Approach, Ellis Horwood, London.
Merletti, R. & Parker, P. (2004). Electromyography: Engineering and Noninvasive Applications, John Wiley & Sons - IEEE Press, ISBN 978-0-471-67580-8, Hoboken, NJ, USA.
Oskoei, M. A. & Hu, H. (2007). Myoelectric Control System - A survey. Biomedical Signal Processing and Control, Vol. 2, pp. 275-294.
Parker, P.; Englehart, K. & Hudgins, B. (2006). Myoelectric signal processing for control of powered limb prostheses, Journal of Electromyography and Kinesiology, Vol. 16, pp. 541-548.
Popovic, D.; Oguztorelli, M. N. & Stein, R. B. (1995). Optimal control for an above-knee prosthesis with two degrees of freedom, Journal of Biomechanics, Vol. 28, pp. 89-98.
Rodrigues, S. S.; Ferreira, R. U.; Marques Junior, M. F.; Beckmann, E. D.; Santos, G. F.; Borges, G. A.; Ishihara, J. Y. & da Rocha, A. F. (2006). Estudo e desenvolvimento de uma prótese robótica de perna comandada por sinais eletromiográficos. Proceedings IV Congresso Ibero-Americano Sobre Tecnologias de Apoio a Portadores de Deficiência.
SENIAM (2008). Surface Electromyography for Noninvasive Assessment of muscle, [http://www.seniam.org].
Silva, J.; Chau, T. & Goldenberg, A. (2003). MMG-Based Multisensor Data Fusion for Prosthesis Control. Proceedings IEEE CMBS, pp. 2909-2912.
Smith, R. J.; Tenore, F.; Huberdeau, D.; Etiene-Cummings, R. & Thakor, N. V. (2008). Continuous decoding of finger position from surface EMG signals for the control of powered prostheses, Proceedings of the 2008 IEEE/EMBC 30th Annual International Conference Engineering in Medicine and Biology Society, pp. 2393-2396.
Sommerich, C. M.; Joines, S. M., Hermans, V. & Moon, S. D. (2000). Use of surface electromyography to estimate neck muscle activity, J. Electromyography Kinesiol., No. 6, pp. 377-98.
Vaseghi, S. V. (2000). Advanced Digital Signal Processing and Noise Reduction, John Wiley

Publisher, 2nd edn, New York.
Wang, G.; Wang, Z.; Chen, W. & Zhuang, J. (2006). Classification of surface EMG signals using optimal wavelet packet method based on Davies-Bouldin criterion, Med. Biol. Eng. Comput., Vol. 44, pp. 865-72.
Zardoshti-Kermani, M.; Wheeler, B. C.; Badie, K. & Hashemi, R. M. (1995). EMG feature evaluation for movement control of upper extremity prosthesis, IEEE Trans. Rehabil. Eng., Vol. 3, pp. 324-33.
Zhao, J.; Xie, Z.; Jiang, L.; Cai, H.; Lio, H. & Hirzinger, G. (2006). EMG control for a five-fingered interactuated prosthetic hand based on wavelet transform and sample entropy, Proceedings IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 3215-20.

23

A Self Organizing Map Based Postural Transition Detection System

Wattanapong Kurdthongmee
Walailak University
Thailand

1. Introduction

A high percentage of elderly people has been continually increased in every part of the world. This results from the successes of both the advancements of medical knowledge and the high coverage of medical services which altogether make people healthier and live longer. One of the consequences of advancing years is greater frailty due to declining health and mobility, leading to increased risk of injury and fatality due to accidents, especially in the home. It can be said that falls are the most common type of home accidents among elderly people and are a major threat to their health and independence (Najafi, 2002). Thirty-two percent of a sample of community dwelling persons 75 years and older fell at least once a year. Among them, 24% sustained serious injuries (Tinetti, 1988). In addition, falling can dramatically change elderly people's self-confidence and motivation, affecting their ability to function independently and result in decreased activity, isolation and further functional decline. In addition, the cost of caring for elderly people after a fall is substantial, although estimates vary. Some suggest that fall-related injuries account for 6% of all medical expenditures in the USA (Stevens, 2006). Considering the growing proportion of old people (over 75) in the populations of industrial countries, falls will be one of the major problems of this important part of the population (Askham, 1990). In 2050, 16.4% of the world population and 27.6% of the European population are projected to be 65 years and above, and in 14 countries, including nine European countries, more than 10% of the total population will be 80 years or older. Most cases of falls sustained by elderly people appear to result from the cumulative effect of multiple specific disabilities. The normal changes of aging, like poor eyesight or poor hearing, can make elderly people more likely to fall. Illnesses and physical conditions can affect their strength and balance. Poor lighting or throw rugs in their home can make them more likely to trip or slip. The side effects of some medicines can upset elderly people's balance and make them fall. Medicines for depression, sleep problems and high blood pressure often cause falls. Some medicines for diabetes and heart conditions can also make them unsteady on their feet.
As a way to observe the activities of elderly people in order to prevent and/or alert when they fall, an EKG (Electrocardiogram) sensor, a pulse sensor and a blood pressure sensor are also employed. It is expected that if some measuring parameters are changed, they might result in a severe fall and unconsciousness of elderly people. To function correctly, these sensors must be firmly attached to the body of wearers by a well-trained medical staff. The sensors may cause uncomfortable for wearers and make them lack of self-confidence to live by their own. In addition, from the technical point of view, these sensors consume a high percentage of battery power if they are operated continuously or at the high sampling frequency. The aim

of our research as a whole is to minimize the fall risk effecting parameters which should be sampled from the movement of wearers. This also leads to minimizing the number of sensors to implement the monitoring and alert system.
Evaluating the risk of falling is important because it enables the provision of adapted assistance and of taking preventive measures with subjects deemed at risk of falling. The risk of falling has generally been evaluated by using questionnaires with their associated problems of subjectivity and limited accuracy in recall (Cummings, 1988). Risk of falls can also be evaluated by clinical and functional assessment including posture and gait, independence in daily life, cognition, and vision (Tinetti, 1986). However, no simple objective method is available. A method of evaluating the characteristics of postural transition (PT) and their correlation with falling risk in elderly people is described in (Tinetti, 1988; Riley, 2008).
This chapter is aimed to present the details of an adaptive postural transition detection system which operates by use of a wearer's knee movement. The system is developed in order to make a decision if a wearer has the transition types of either sit-to-stand (Si-St) or stand-to-sit (St-Si) which are believed to be a fall-risk in elderly people. The output from the system can be used to trigger an additional sensor to operate in order to make further measurement and judgement. Within the system, the Kohonen self organizing map (KSOM or SOM) neuron network was employed to perform an adaptive analyzer and classification functions which made the system applicable to wearers with different ages and motion conditions. In the next section, the literature survey of related research work on the topic of PT detection is described.

2. Literature survey

The PT detection by use of a single 3-axis accelerometer attached to the chest is presented in (Barralon, 2005). To automate the identification processes for sitting and standing postures and pave the way to implement an automatic fall detection system, (Nyan, 2006) reports the developed system for measuring the time of Si-St and St-Si transitions and their duration. In the developed system, a miniature gyroscope attached to the chest and a portable data logger placed on the waist are employed. The comparison between two groups of elderly people subjects (with high and low fall-risk) shows that the computed parameters are significantly correlated with the fall risk as determined by the record of falls during the previous year. Moreover, the parameters are correlated to the balance and gait disorders, visual disorders, and cognitive and depressive disorders. A 2-axis accelerometer is used for motion detection of body movement in (Yang, 2007). A dedicated body movement algorithm embedded in the microcontroller is developed to actively recognize three still postures (sitting, standing, and lying), four postural transitions (sit-stand transitions and lie-sit transitions) and locomotion (walking) in a home environment. The proposed algorithm relies on using the slope mapping technique. The experimental result shows that the detection correctnesses of Si-St and St-Si are 92.2 and 95.6%, respectively. In (Bidargaddi, 2007), a wavelet-based algorithm for detecting and calculating the durations of Si-St and St-Si transitions is developed and reported. The algorithm processes the signal vector magnitude of the measured acceleration signal. Later, (Najafi, 2008) proposes an approach to utilize the pattern of the vertical accelerometer to detect a PT, recognize the interval of PT and finally classify the transition types. The classification process is based on the time-frequency analysis. This is performed by considering that a gravitational acceleration has a lower frequency component as compared with a translational (linear) acceleration, and that a velocity does not have a DC component.
Accelerometers and magnetometers are used to study the activity of person (Fleury, 2009). The signal from these sensors are analyzed by a wavelet-based pattern recognition algorithm

in order to detect the postural transitions. This is fairly similar approach to the one previously presented in (Bidargaddi, 2007). Results of an experiment are also given to show a mean classification rate of 70% for this approach. In contrast to the previously proposed approaches, a Hidden Markov Model (HMM) processing framework for stochastic identification of body postures and physical contexts is proposed in (Quwaider, 2009). The key idea relies on collecting and wireless transmitting data from multi-modal sensor attached to a human subject's body segments. The data is processed by using HMM in order to identify the subject's instantaneous physical context. It is claimed that the conducted experiments show that a technique can achieve high detection match rates for all posture transitions. Unfortunately, no comparative result is presented in the publication. Recently, it is studied in (Dijkstra, 2010) whether gait and postures can accurately be detected with a single small body-fixed device. The study is focused on a group of patients with mild to moderate Parkinson's disease (PD). It is concluded that the triaxial monitor system is a practical and valuable tool for objective, continuous evaluation of walking and posture transitions in patients with mild to moderate PD. Unfortunately, it is concluded that detection of sitting and standing requires further fine-tuning.
From our point of view, there are several drawbacks of the previously proposed system. That is to say the differences in the collected data among different persons, or even within the same person but different time, which are very common are not taken into account. The proposed postural transition detection systems are all in a class of a pre-programmed system with the threshold for warning or alert resulted from limited samples. This is in contrast to our proposed system that relies on utilizing SOM to make it adaptable to a particular wearer. The movement nature of the wearer is used for system training and, in turn, used to report the transition types of the wearer. The details of our proposed system are given in the next sections.

3. Method

In this section, an introduction to SOM which is our main tool is given. Then, it is followed by the details of our proposed system which consists of the wireless sensor network based system and the developed computer software to communicate with the wireless sensor network.

3.1 A brief introduction to SOM

In general, SOM is one of the most prominent artificial neural network models adhering to the unsupervised learning paradigm (Kohonen, 1990). It has been employed to solve problems in a wide variety of application domains. For the applications in engineering domain, it was elaborately surveyed and reported in (Kohonen, 2002). Generally speaking, the SOM model consists of a number of neural processing elements (PE). Each of the PE, i, is assigned an n-dimensional weight vector m_i where n is the dimension of an input data. During the learning stage, the iteration t starts with the selection of one input data $p(t)$. $p(t)$ is presented to SOM and each PE determines its activation by means of the distance between $p(t)$ and its own weight vector. The PE with the lowest activation is referred to as the winner, m_c, or the best matching unit (BMU) at the learning iteration t, i.e.:

$$m_c(t) = min_i \parallel p(t) - m_i(t) \parallel . \tag{1}$$

The euclidean distance (ED) is one of the most popular way to measure the distance between $p(t)$ and a PE's weight vector $m_i(t)$. It is defined by the following equation:

$$d(p(t), m_i(t)) = \sqrt{(p(t)_1 - m_i(t)_1)^2 + (p(t)_2 - m_i(t)_2)^2 + ... + (p(t)_n - m_i(t)_n)^2} \quad (2)$$

Finally, the weight vector of the winner PE as well as the weight vectors of selected PEs in the vicinity of the winner are adapted. This adaptation is implemented as a gradual reduction of the component-wise difference between the input data and weight vector of the PE, i.e.:

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot h_{ci}(t) \cdot [p(t) - m_i(t)]. \quad (3)$$

Geometrically speaking, the weight vector of PEs of the adapted units are moved a bit towards the input data. The amount of weight vector movement is guided by a learning rate, α, decreasing with time. The number of PEs that are affected by this adaptation is determined by a neighborhood function, h_{ci} which also decreases with time. This movement makes the distance between these PEs decrease and, thus, the weight vector of the PEs become more similar to the input data. The respective PE is more likely to be a winner at future presentations of this input data. The consequence of adapting not only the winner alone but also a number of PEs in the neighborhood of the winner leads to a spatial clustering of similar input patterns in neighboring parts of the SOM. Thus, similarities between input data that are presented in the n-dimensional input space are mirrored within the two-dimensional output space of SOM or SOM map. The learning stage is terminated after the final SOM map is labelled with some known conditions.
The classification stage is very similar to the learning stage with some exceptions. That is to say there is no need to perform adaptation to the winner PE and its neighbours of the SOM map with respect to the input data. Instead, the label of the winner PE corresponding to the input data is returned and used for further interpretation; i.e. if the input data is mapped to the PE whose label is either Si-St or St-Si transitions.

3.2 The wireless sensor network based system

To be successfully employable to the main target of the proposed detection system which are elderly people, several criteria must be considered (Rajendran, 2008). One of the most important criteria is that the system must be the least obtrusive. In addition, the system should be as small as possible to ease attachment to the body of elderly people and to avoid eye catching by others. It must require the least cable connection. Also, the power consumption of the overall system must also be kept minimum in oder to lengthen the battery operating time. To fulfill these requirements, the wireless sensor network (WSN) based system was selected from our initial design stage. During the course of design and experimentation, a WSN based data acquisition system was developed to be the least obtrusive manner to gather knee movement data from elderly people. During the learning stage, the gathered data is used for training SOM and activity types labeling to its resulting map on a semi-automatic basis. Later during the normal operation mode (classification stage), the sampled data is automatically gathered and mapped to the appropriate PE. The label of the mapped PE which is either Si-St or St-Si transition is then returned. The overall roles of the developed WSN are shown in Fig. 1.
The developed WSN system was based on the MSP430 family of microcontrollers from Texas Instruments Inc (Texas Instruments Inc., 2008). The eZ430-RF2500 development tool, both hardware and software, was selected as it consists of all required components to accelerate

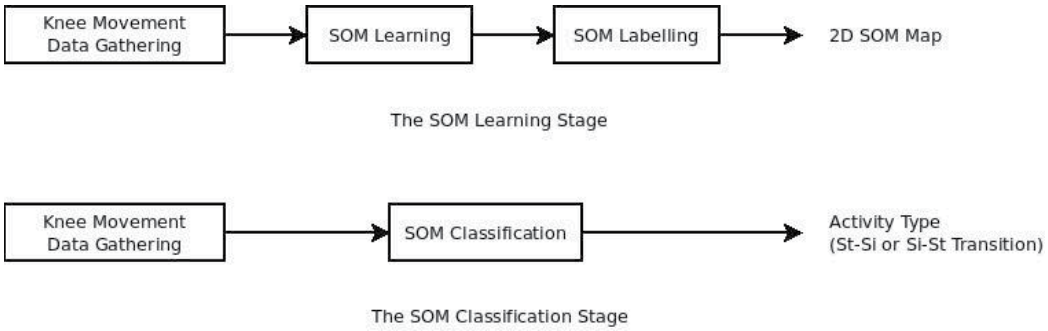


Fig. 1. The overall roles of proposed system.

the WSN design and implementation process. Fig. 2 illustrates the proposed system during the SOM learning and classification stages. The hardware part of the tool integrates the MSP430F2204 microcontroller (which combines 16-MIPS performance with a 200-ksps 10-bit Analog-to-Digital Converter (ADC)) with the CC2500 multi-channel RF transceiver chip. The overall size of the system is about half of the thumb size as clearly be shown in the top-left part of Fig. 3(a). A pair of eZ430-RF2500 were used with two different versions of software to operate as an end device and an access point modes. The main role of an end device during the learning stage is to sample the knee movement data via an external sensor (will be detailed shortly), pack the sampled data into a communication protocol, and transfer the protocol to an access point via a wireless data communication. An access point which is connected to a personal computer is responsible for receiving incoming protocol from one or many end devices and transforming and transferring the packet to a personal computer via a USB port. The upper left and right parts in Fig. 3(a) illustrates an eZ430-RF2500 module and an access point (an eZ430-RF2500 module with an RS-232 to USB protocol converter module), respectively. The middle part of the figure is the prototype of an end device with an external sensor. Fig. 3(b) illustrates the attachment of an end device to our volunteer's knee by use of a Tubigrip.

In contrast to the previously proposed PT detection systems which rely on using an

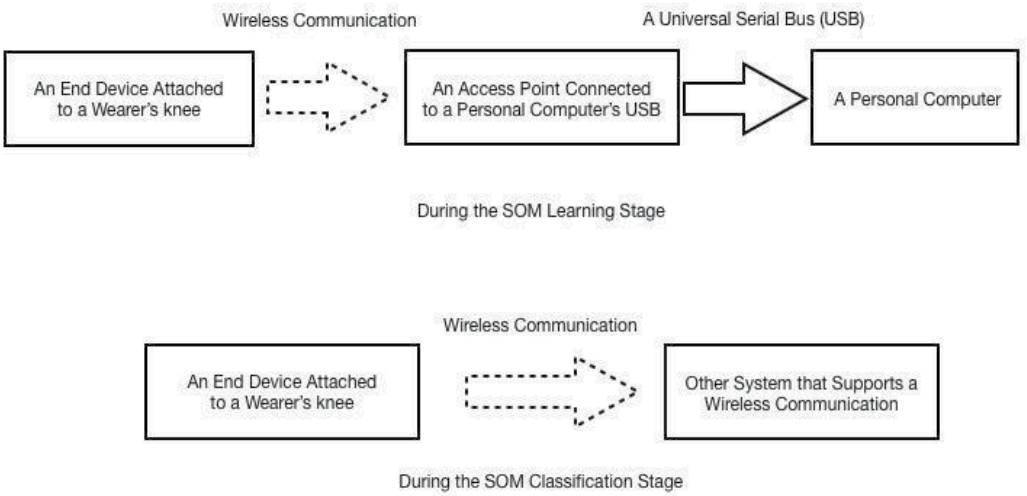


Fig. 2. The architecture of the proposed WSN system.

(a)

(b)

Fig. 3. (a) An eZ430-RF2500 module (top-left), an access point (top-right), the prototype of an end device with an external sensor (middle), (b) The attachment of the end device with our volunteer's knee by use of a Tubigrip

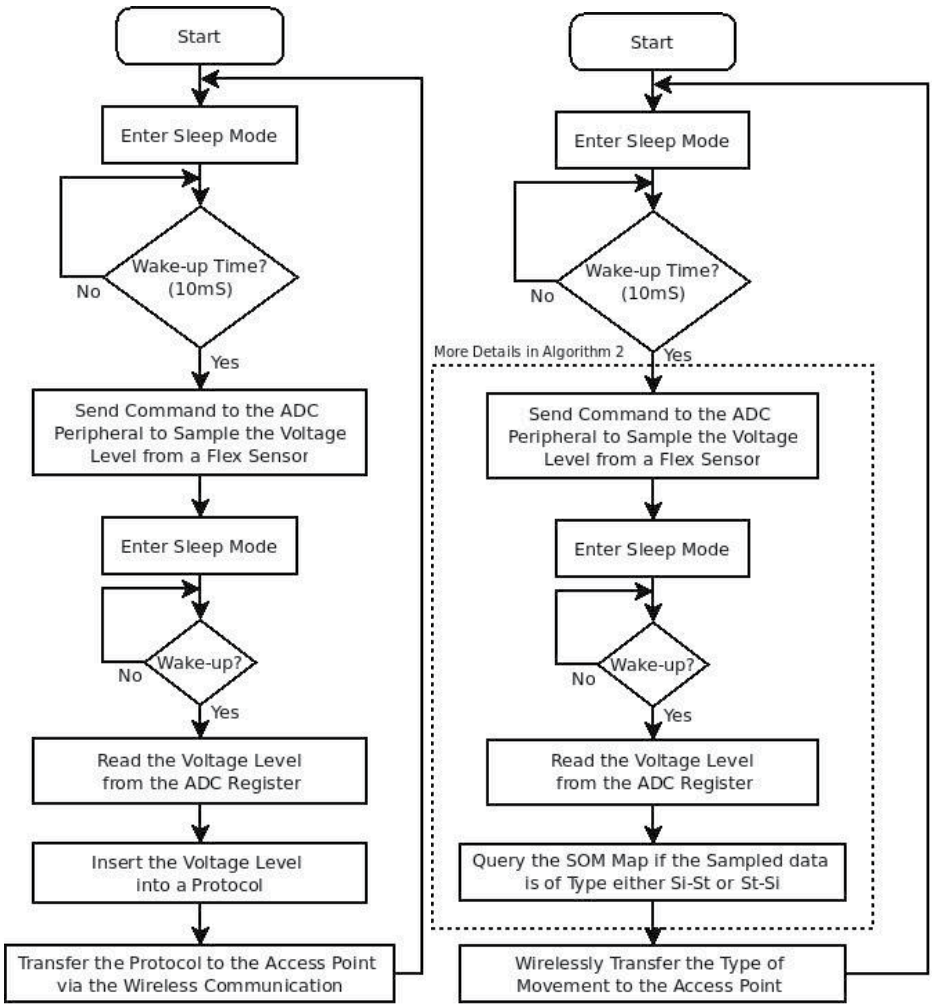


Fig. 4. Flowcharts of (a) an end device operations during the data gathering/transmitting stage, (b) an end device operations during the normal operation stage

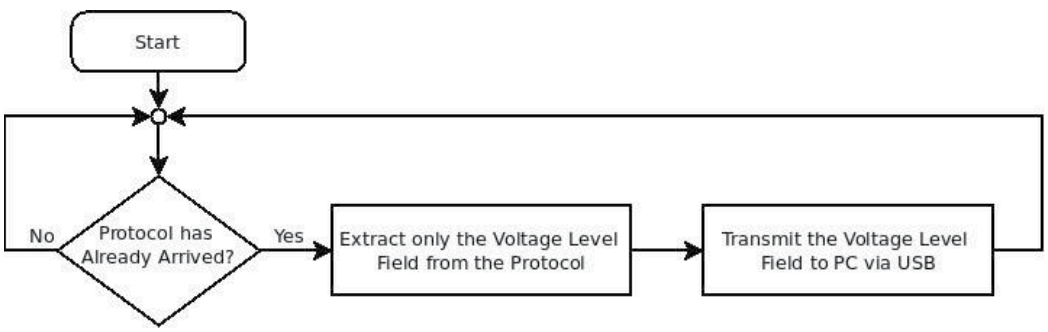


Fig. 5. Flowchart of an access point.

Fig. 6. The flex sensor: the sensor used to measure a knee movement variable in our proposed system. Image from: www.sparkfun.com

accelerometer or other active sensors (i.e. tilt sensors and/or gyroscopes), our system makes use of a flex sensor. The sensor is a tiny bendable strip, with the dimension of about 3/8″ wide by 5″ long, which is shown in Fig. 6. It is a passive resistive device that can be used to detect bending or flexing. Previously, the sensor has been found used in many applications which are: to detect collision on mobile robots and to construct virtual reality (VR) gloves and VR suits. In addition, some physics applications and experiments have also exploited this sensor to measure/detect bending. The resistance of the sensor decreases in proportion to the amount it is bent in either direction. As it is a passive device, it consumes no electric current during operation. This makes it very attractive for an application that requires a long battery operating time like this application. In our proposed system, only a simple voltage divider circuit is required to interface between the MSP430F2204 microcontroller and the flex sensor. The voltage divider consists of a single extra resister to connect between the positive terminal of a power supply and one terminal of the flex sensor. The other terminal of the sensor is connected to the ground of the power supply. The connection point between the resister and the sensor is served as an input to an analog-to-digital pin of the MSP430F2204 microcontroller.

It is worth noting at this point that there are two different roles of an end device which are clearly illustrated in Fig. 1. That is to say during SOM learning stage, it serves as a system to gather knee movement data and wirelessly transmit to an access point. The transferred data is used for training SOM on a personal computer in order to produce the most appropriate map; i.e. the map whose quantization error is the minimum. This is operated on an offline and semi-automatic basis and on a personal computer. In the normal operation stage, the role of the end device is changed to sample the knee movement and query the map to make a final decision if the movement is of type St-Si or Si-St transition. This means that there is no need to use an access point anymore. The resulting decision is, in turn, wirelessly sent to caretakers or other system in order to alert caretakers to pay more attention to the wearer or to activate the other system to further measure additional movement parameters. Alternatively, the activation can be used to initiate the video system to capture motion patterns of the

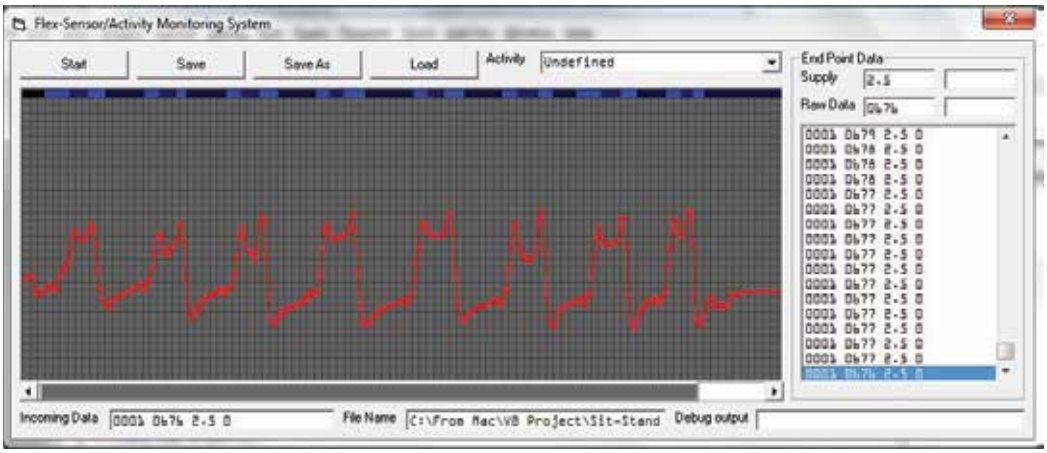


Fig. 7. The screenshot of the application running on a personal computer.

wearer. There are wide variety of applications of the resulting decision from our proposed PT detection system. The flowcharts showing the operational steps of an end device in these two stages are illustrated in Fig. 4. In addition, the flowchart of an access point previously detailed is also presented in the same figure. It is noticeable that an end device spends most of its operational time in waiting for wake-up loop; i.e. sleep mode of operation. It is our intention to design an end device to behave in this way as it lengthens the battery life cycle. This comes from the fact that the microcontroller of an end device consumes the least power in the sleep mode.
In the next section, we describe the details of a personal computer based application software responsible for making communication with an access point, providing assistance for activity types labeling and invoking the backend SOM toolbox to perform the learning stage.

3.3 The developed computer software

During a SOM learning stage, running on a personal computer is a Microsoft Windows based application to communicate with an access point via a universal serial bus (USB). The application was developed in-house by use of the Microsoft Visual Basic 6.0. The screenshot of the application is illustrated in Fig. 7. The signals currently shown on the screen of the application are captured while the wearer of the end device was repeatedly performing Si-St and St-Si transitions (with short periods of sitting and standing). Along with showing the plot of the incoming signals, the application also shows the time series of the incoming raw data. These are partially shown in the listbox on the right side of the application window.
Followings are the roles of the application software:

- Retrieve the data packets from an access point, extract the knee movement signals and store in a personal computer on a real-time basis,
- Provide a user interface tool to assist an application user to locate and mark the beginning and end of each type of knee movement to the signals,
- Preprocess the signals and perform the SOM learning stage in order to come up with the final SOM map.

At this point, it is expected that it became clear to readers for the first role of the application. Let's give more details for the second role of the application which results in generation

the thin horizontal colour bar embedded at the topmost of the signal display screen. It is observable from Fig. 7 that there are three different colours in the colour bar for an individual Si-St and St-Si transitions which are light blue, black and dark blue. These colours represent three different activities which are: Si-St and St-Si transition, respectively. The application was developed to provide a mechanism to respond to the mouse click event from an application user. Typically, an application user clicks the mouse in order to place vertical lines at different location on the signal display screen. The vertical lines are used to cluster signals resulting from the same activity types together and separate the signals resulting from different activity types. Having finished placing the vertical lines, the next step is to assign the activity type to all the signals between each pair of the vertical lines. It is noted that the above steps are required to perform manually. This could inevitably result in incorrect labeling signal and/or misplacing activity separation lines even with very careful placement of the vertical lines. However, we developed an additional algorithm to make correction to these (will be presented in Section 4) and have found from the experiment that this does not degrade the overall outcome PT detection results.
Since during the experiment a single end device was used, given the incoming raw data represented by the following set:

$$S = \{K_0, K_1, K_2, K_3, ..., K_{n-1}\}. \tag{4}$$

The second and third roles of the application resulted in producing the following set of data to be ready for the SOM learning stage:

$$\{K, \delta K, a\} = \{(K_0, K_1 - K_0, a_0), ..., (K_i, K_{i+1} - K_i, a_i), ..., (K_{n-2}, K_{n-1} - K_{n-2}, a_{n-1})\}, \tag{5}$$

whose member consists of the signal value K_i, the slope between the consecutive pair of signals K_{i+1} and K_i and the activity type a_i. The activity types are classified into jogging, sitting, standing, walking, walking downstair, walking upstair, sit-to-stand and stand-to-sit and numbered between 1 to 8, respectively. The sample period of the end device was fixed to be 10 mS. It can be seen that only the second field of data was calculated by the application. The third data is resulted from the second role of the application described earlier. The first and second fields of data are then normalized to have a mean of zero and a standard deviation (SD) of one to be ready for the SOM learning stage. As a backend tool to perform the SOM learning and labeling stages, the application relies on using a SOMPAK toolkit (Kohonen, 1996).
In the next section, we present the routine to be performed during the SOM learning stage, the details of the algorithm to increase the correctness of the PT classification and the experimental results.

4. Experimental results

In order to perform the SOM learning stage with the knee movement data captured by the system detailed in the previous section, the procedures to capture data was developed. Followings are the activities that need performing by a wearer during the data capturing stage:

- Perform a normal walk,
- Perform a normal jogging,
- Perform a normal walking up and down stairs,
- Perform a sit to stand and sit transition

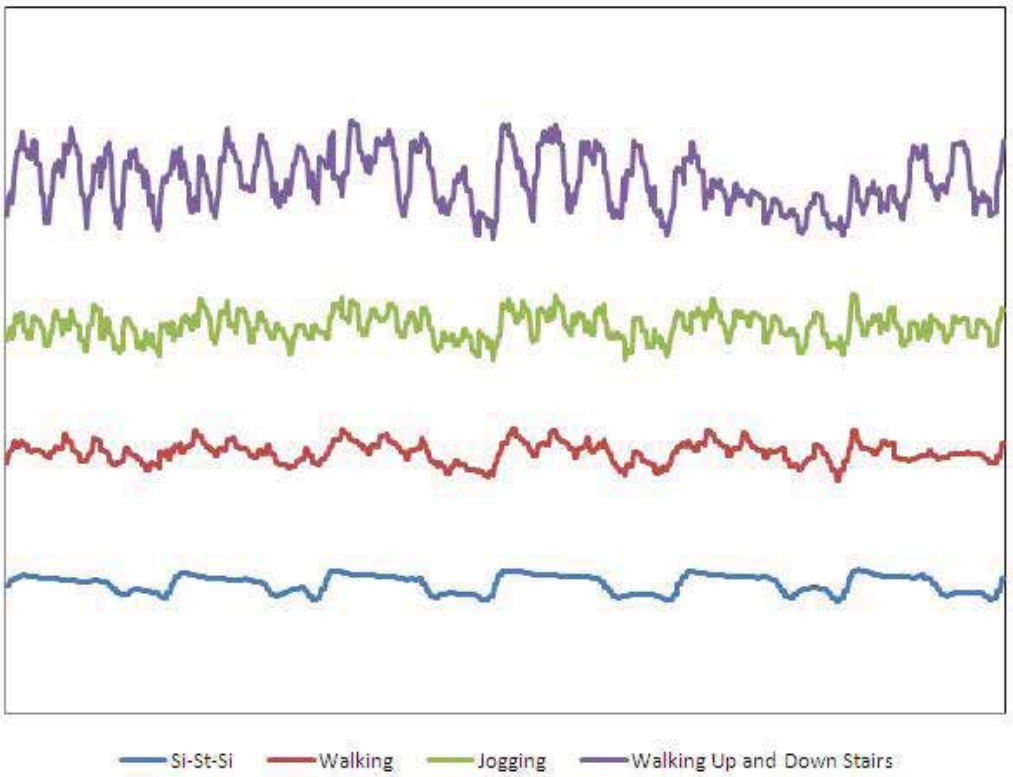


Fig. 8. The plot of the knee movement data from all captured activities in the same graph.

Each activity is equally captured and wirelessly transferred from the end device to a computer via the access point for a period of 10 seconds. This is equivalent to the number of 1,000 sampled data per activity as the sample period is 10 mS. It is noted that while performing these procedures, a wearer needs to be closely monitored by our staff. This is done in order to prevent the wearer to perform incorrect activities apart from the activity to be captured. This reduces capturing unwanted activities which may result in increasing the incorrect activity labeling. Having finished data capturing stage, the captured data are manually labelled utilizing the features of the computer application and ready for use in the SOM learning stage. Fig. 8 illustrates the plots of all activities with the aim to show that different activities result in dissimilar raw data pattern. It is noted that the signal levels should not be considered in this case.

With respect to our experimentations, the SOM learning stage which gave rise to the best quality map, measured by a quantization error (QE), consisted of two phases with the neural processing elements of 14×20. In the first phase, the learning coefficient $\lambda(t)$ was set to decrease from 1.0 to 0 in 100,000 iterations, while the radius of the neighbourhoods was reduced from 15 at the beginning to 1 at the end of the learning stage. In the second phase, $\lambda(t)$ was set to decrease from 0.125 to 0 in 10,000 steps, while the radius of the neighborhood was reduced from 3 to 1. The map were labelled with only 2 knee movement types which are: Si-St and St-Si transitions whose number is 7 and 8, respectively. Fig. 9 depicts (a) the resulting SOM map, (b) the SOM map projection on the normalized knee movement

data only, and (c) the SOM map projection on the normalized delta of knee movement data only. These maps were reproduced by use of a Neural Networks Tool (Nenet) from koti.mbnet.fi/phodju/nenet/Nenet/General.html.
It is observable from the resulting map that some PEs were labelled with both knee movement types. This can be explained that Si-St and St-Si transitions have shared some common knee movement parameters. This could also be resulted from incorrectly manual labeling of activity types during using the application software by our staff. In order to avoid making difficulties to the labeling stage, we developed an addition algorithm, Algorithm 1, to take into account the popularity of being matched for all PEs. That is to say, in stead of keeping only the activity type label, each PE needs to have an additional register set, a popular register set or PS in the algorithm, to keep track of the number of times it is matched by activity types during the SOM labeling stage. For example, at the end of learning stage after applying this addition algorithm, it is found that the registers of the PE at position (1, 14) of Fig. 9 have the following values: $[0,0,0,0,0,0,0,2670,1330]$. This means that this PE is more popular to the activity type 7 (Si-St transition) compared to the activity type 8 (St-Si transition). During the classification stage, if a knee movement parameter, K, and its delta, δK, counterpart is mapped to this PE, it is reasonable to conclude that the knee movement is of type Si-St transition.

input : SOM: the already trained SOM map, $\{K,\delta K\}$: Data set for training
output: The labelled SOM map with embedded popular set; PS, in all PEs

foreach *member indices* (i,j) *of SOM* **do**
  $PS_(i,j) \leftarrow [0,0,0,0,0,0,0,0,0]$
end

foreach *member m of* $\{K,\delta K\}$ **do**
  $a \leftarrow$ *the activity type of* $\{K,\delta K\}_m$;
  $(r,c) \leftarrow$ *indices of the BMU with respect to* $\{K,\delta K\}_m$;
  $PS_{(r,c)}[a] \leftarrow PS_{(r,c)}[a] + 1$;
end

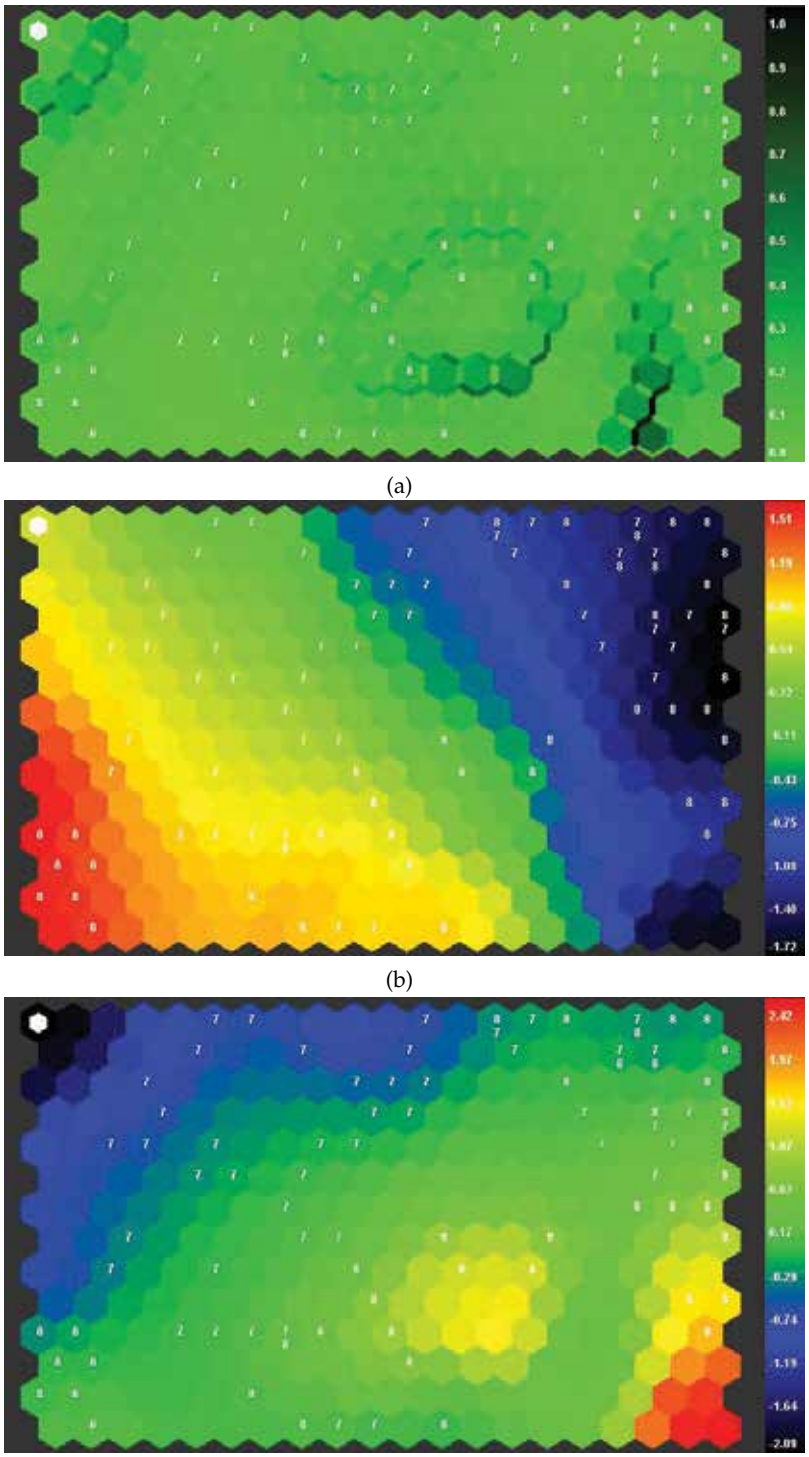
Algorithm 1: An algorithm for SOM map labeling with taking into account the popularity of being matched of all PEs.

input : The labelled SOM map with embedded PS; a set of 10 samples of $\{K,\delta K\}$
output: C: Classification result

$C = 0$;
for $i \leftarrow 1$ **to** 10 **do**
  $(r,c) \leftarrow$ *indices of the BMU with respect to* $\{K,\delta K\}_i$;
  $a \leftarrow arg(max(PS_{(r,c)}))$;
  $C \leftarrow C + a$;
end
$C = C/10$

Algorithm 2: An algorithm for activity classification with making used of the popularity of being matched and calculating the average activity from 10 consecutive samples.

Following the previous additional algorithm for classifying a single sample, the classification error is reduced significantly. However, this does not satisfy our requirement. We then



(a)

(b)

(c)

Fig. 9. (a) The result SOM map after labeling stage which is displayed by use of a Nenet-tool.

Activity Type	Percent Correctness
Si-St	100
St-Si	93.3

Table 1. The percent correctness of the verification results from testing the map with unknown activity data retrieved from the end device

further take an additional feature to the classification algorithm. This is simply done by taking the average of the classification results instead of relying on returning the activity type immediately after querying the SOM map with only a single knee movement parameter. The algorithm is detailed in Algorithm 2. We have found that by taking the average of the classification results from 10 consecutive samples of knee movement parameter (equivalent to 100 mS), it gives rise to the best result. Taking more samples results in reducing the detection sensitivity of the system without increasing the classification correctness.
After integrating all proposed algorithms to the end device, we then verify whether the resulting map was capable of finding the correct movement type with respect to an unknown activity retrieved from the end device. The verification results are shown in Table 1.
It can be seen that the verification results give rise to very high degree of correctness. We, therefore, ported the system to implement as a stand-alone system relied on using a knee movement data. The operations of the system has already been illustrated in Fig. 4(b).

5. Conclusion

In this chapter, we proposed another utilization of SOM as an approach to classify postural transitions from knee movement data. The details and experimental results of a SOM-based stand-alone postural transition detection system were also presented. The stand-alone system is embedded with a low-power consumption microcontroller with a multi-channel RF transceiver chip. The sensor uses for gathering knee movement data is a flex sensor whose type is a passive component. The overall system is in the category of wireless sensor network (WSN) with two different operation stages: a SOM learning and a classification stage. During the SOM learning stage, the end device subsystem samples and wirelessly transfers a knee movement data to the access point subsystem. The access point subsystem then encodes the incoming protocol and transmits only the movement data to a personal computer. The data, which are consisted of a movement data and the delta between a consecutive pair of data, are used for training SOM on an off-line and semi-automatic basis. We experimentally proved that SOM could be successfully employed in this application domain. The results, the final SOM map, show that similar movement parameters are clustered together. Also, SOM gives rise to more than 93.3 percents of correctness for making decision of stand-to-sit and sit-to-stand transition types from a knee movement parameter. With positive experimental results, we modified the software of the end device subsystem to perform the SOM classification stage. The final stand-alone system can be utilized to make the decision and alert to caretakers or another fall-risk detection system.

6. Acknowledgements

This work was supported by Thailand Toray Science Foundation and Thailand's National Electronics and Computer Technology Center (NECTEC) under the project: a study of fall-risk affecting motion/movement parameters and develop a neuron network-based adaptive

fall-risk analyzer. The author would like to thank all anonymous reviewers for their comments in previous versions of this paper.

7. References

Askham, J.; Glucksman, E.; Owens, P.; Swift, C.; Tinker, A. & Yu, G. (1990), Home and leisure accident research: A review of research on falls among elderly people, In: *Age Concern Institute of Gerontology*, King's College, London, UK.

Barralon, P.; Noury, N.; and Vuillerme, N., Classification of Daily Physical Activities from a Single Kinematic Sensor, Proc of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, Shanghai, China, 2005.

Bidargaddi, N.; Klingbeil, L.; Sarela, A.; Boyle, J.; Cheung, V.; Yelland, C.; Karunanithi, M. & Gray, L., (2007), Wavelet based approach for posture transition estimation using a waist worn accelerometer, In: *Conf Proc IEEE Eng Med Biol Soc.*, 2007, pp. 1884 - 1887.

Cummings, S. R.; Nevitt, M. C. & Kidd, S. (1988), Forgetting falls: The limited accuracy of recall of falls in the elderly, In: *J. Amer. Geriatr. Society*, Vol. 36, pp. 613 - 616.

Dijkstra, B.; Kamsma, Y. P.; Zijlstra, W. (2010), Detection of gait and postures using a miniaturized triaxial accelerometer-based system: accuracy in patients with mild to moderate parkinsons disease, In: *Archives of Physical Medicine and Rehabilitation*, Vol. 91, No. 8, pp. 1272-1277.

Fleury, A.; Noury, N.; and Vacher, M., A wavelet-based pattern recognition algorithm to classify postural transitions in humans, Proc of 17th European Signal Processing Conference (EUSIPCO 2009), 2009, Glasgow, Scotland.

Hwang, J. Y.; Kang, J.M.; Jang, Y.W. & Kim, H. C. (2004), Development of novel Algorithm and Real-time Monitoring Ambulatory System Using Bluetooth Module for Fall Detection in the Elderly, In: *Proc. of the 26th Annual Int. Conf. of the IEEE EMBS*, San Francisco, CA, USA.

Jovanov, E; Milenkovic , A; Otto, C; de Groen, P.; Johnson, B.; Warren, S. & Taibi, G. (2005), A WBAN System for Ambulatory Monitoring of Physical Activity and Health Status: Applications and Challenges, In: *Proc. of the 27th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, Shanghai, China.

Joutsiniemi, S. L.; Kaski S. & Larsen, T. A. (1995), Self-Organizing Map in Recognition of Topographic Patterns of EEG Spectra, In: *IEEE Trans. on Biomedical Engineering*, Vol. 42, No. 11, pp. 1062 - 1068.

Kohonen, T. (1990), The Self-Organizing Map, In: *Proc. of IEEE*, Vol. 78, No. 9, 1990, pp. 1464 - 1480.

Kohonen, T; Hynninen, J.; Kangas, J. & Laaksonen, J. (1996), In: *SOMPAK: The Self-Organizing Map Program Package*, Available from www.cis.hut.fi/research/papers (accessed on 13/03/2009).

Kohonen, T; Oja, E.; Simula, O.; Visa, A. & Kangas, J. (2002), Engineering applications of the self-organizing map, In: *Proc. of IEEE*, Vol. 84, No. 10, 2002, pp. 1358 - 1384.

Najafi, B.; Aminian, K.; Loew, F.; Blanc, Y. & Robert, P. A. (2002), Measurement of Stand - Sit and Sit - Stand Transitions Using a Miniature Gyroscope and Its Application in Fall Risk Evaluation in the Elderly, In: *IEEE Trans. on Biomedical Engineering*, Vol. 49, No. 8, pp. 843 - 851.

Najafi, B.; Wrobel, J. & Armstrong D. G. (2008), A novel ambulatory device for continuous 24-h monitoring of physical activity in daily life, In: *Proceedings of American Society of Biomechanics 2008*, Ann-Arbor, Michigan.

Nyan, M. N.; TAY, Francis E. H.; Manimaran, M. & Seah K. H. W. (2006), Garment-based detection of falls and activities of daily living using 3-axis MEMS accelerometer, In: *Journal of Physics: Conference Series*, Vol. 34, 2006, pp. 1059 - 1067.

Oja, M.; Kaski, S. & Kohonen, T. (2002), Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001 Addendum, In: *Neural Computing Surveys*, 3, 1 - 156.

Patent (2008), Patents: 5823845, 7141026, 6165143, 6095991, 6059576, 5919149, Available from http://www.patentstorm.us (accessed on 16/06/2008).

Quwaider, M.;Plummer Jr., A.; Rao, J.; Taghizadeh, M. & Biswas, S. (2009), Real-time posture detection using body area sensor networks, In: *The 13th IEEE International Symposium on Wearable Computers*, Linz, Austria.

Rajendran, P.; Corcoran, A; Kinosian, B. & Alwan, M. (2008), Falls, Fall Prevention and Fall Detection Technology, In: *Eldercare Technology for Clinical Practitioners*, Alwan, M; Felder, R. (Eds.), Springer, ISBN: 978-1-58829-898-0, pp. 187 - 202.

Riley, P. O; Paylo, K. W. & Kerrigan, D. C. (2008) Mobility and Gait Assessment Technologies, In: *Eldercare Technology for Clinical Practitioners*, Alwan, M; Felder, R. (Eds.), Springer, ISBN: 978-1-58829-898-0, pp. 33 - 51.

Stevens, J. A.; Corso, P. S.; Finkelstein, E. A.; Miller, T. R. (2006), The costs of fatal and nonfatal falls among older adults, In: *Injury Prevention*, Vol. 12, No. 5, pp. 290 295.

Texas Instruments Inc., MSP430x2xx Family: User's Guide, Online Datasheet Available from: http://focus.ti.com/lit/ug/slau144e/slau144e.pdf (accessed on 06/08/2010).

Tinetti, M. E.; Williams, T. F. & Mayewski, R. (1986), Fall risk index for elderly patients based on number of chronic disabilities, In: *Amer. J. Med.*, Vol. 80, pp. 429 - 434.

Tinetti, M. E.; Speechley, M. & Ginter, S. F. (1988), Risk factors for falls among elderly persons living in the community, In: *N. Eng. J. Med.*, Vol. 319, pp. 1701- 1707.

Yang, C. C. & Hsu, Y. L. (2007, Developing a wearable system for real-time physical activity monitoring in a home environment, In *Proc. of International Association Of Science And Technology For Development*, ISBN: 978-0-88986-668-3, Quebec, Canada, pp. 45 - 50.

Zachrison, A. & Sethson, M. (2006), Detection of System Changes for a Pneumatic Cylinder Using Self-Organizing Maps, In: *Proceedings of the 2006 IEEE Conf. on Computer Aided Control Systems Design*, Munich, Germany, pp. 2647 - 2652.

24

Apparent Age Estimation System Based on Age Perception

Hironobu Fukai, Hironori Takimoto, Yasue Mitsukura, and Minoru Fukumi
Ritsumeikan University
Okayama Prefectural University
Tokyo University of Agriculture and Technology
The University of Tokushima
Japan

1. Introduction

Recently in the world, age estimation is widely studied. Especially, when we use face images, subject's resistance and psychological loads of the age estimation method are smaller than the method by using any other method by using the non-face image. Because faces are always opened to the society compared with other bodily features. Thus, the age estimation method by face image and appearance information is actively researched. By the way, sight information that faces, hairstyle, clothes, and appearance appears the feature of the age and the gender. Especially, face information is thought to be the most much feature information of the age. Because, human see the face and estimate the age when they communicate with somebody else. Therefore, many of the age estimation method by face image is studied [1-10].

In the study on a physical shape change of the face from the age, todd show that contour of skull are approximated by cardioid transform (Todd, Mark). On the other hand, age estimation by computer are performed. Kanno shows that the man was identified by the neural network for four ages (12 years, 15years, 18 years, and 22 years). Y.H.Kwon is reported that the theory has only been implemented to classify input images into one of three age-groups: babies, young adults, and senior adults. The computations are based on cranio-facial development theory and skin wrinkle analysis. Burt studied the age perception that uses the averaged face from 25 to 60 years. Especially, he used texture and shape. Ueki are reported that the age-group classification by the dimension compression.

However, there are some problems in these techniques. It has only the characteristic of roughly classifying the person. Thus, it was assumed that the reliability of the recognition accuracy was a doubt. Because there were possibilities to be classified into a different class in 19 and 20 years old. However, 20 and 29 years old were classified into the same class. In this case, the age is very far from 20 and 29 years old. Therefore, it is necessary to be classified into a different category. Furthermore, it is necessary to classify the same category 19 years old and 20 years old. Actually, it is very difficult to classify as mentioned above. This difference is important problem in the age estimation. Furthermore, in the method of extracting a lot of characteristics, it is difficult to extract these accurately. Then, in this paper,

we propose age estimation every age by using the overall feature of the face. There are few studies for age generation but there is no study to investigate the every age for person. Moreover, feature data for age estimation is the normalized based on the both eyes. Furthermore, we want to analyze what features of the face decide the apparent age. Therefore, we pay attention to the mechanism of the human perception. The main motivation, in this study is to obtain the feature for age and apparent age. The second motivation is to classify the every age like human skill automatically by the proposed method.

First of all, the apparent age based on person's subjectivity in the face image is given. Next, the apparent age is estimated by the computer simulation. The estimation method used supervised self organized map (SOM). It delimits from 15 to 64 years old, and we create an age estimation map of one dimension. Thus, the continuous estimation is possible. The problem by clustering at the age can be solved by estimating in one of every age. Moreover, the feature data that becomes important for the age estimation is extracted by the genetic algorithm (GA). The individual feature that the person has and the feature at the age are extracted by using genetic algorithm for the image that does fast Fourier transform (FFT). To compare all face images as much as possible on the same condition, we propose an original normalized technique of the face image. The rotation angle is corrected to the inclination of the face in normalization. An area necessary for recognition is extracted and to compare it on the same scale, a centre part of the face is cut out. Thus, the background part that is unnecessary information can be reduced. Furthermore, information thought to be related to age excluding face information like the hairstyle and clothes, etc. can be reduced. Finally, it is thought that the age estimation limited to face information can be done.

In order to verify the effectiveness of the proposed method, we show the computer simulation based on the real data.

Size	640×480 [pix.]
	24 bit color
Gender	150 images for each
Age	30 images per 5 years old
Emotion	neutral

Table 1. The detail of the face image database

Fig. 1. Example of original images

2. Preprocessing

To compare faces, we need face normalization and apparent age database creation.

2.1 Face image database

The face database is provided from Human and Object Interaction Processing (HOIP) organization in JAPAN [11]. These subjects are the people with a wide age group that doesn't sport a pair of glasses. The background and proof were made the same condition for all subjects. Subject was directed to make the lens of the camera see, and it took a picture with that look of natural (Table 1). Fig.1 shows the example of the original image. 252 people who gave the preprocessing beforehand are used as subject. In this paper, the face database has obtained the use permission from corporation SOFTOPIA JAPAN. It is prohibited to copy, to use, and to distribute without the authorization of the right holder.

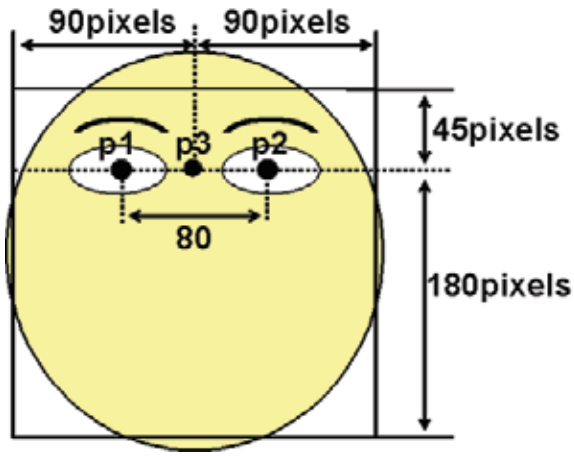


Fig. 2. The outline of the normalization method

2.2 Normalization

It is necessary to normalize the face image to the age estimation. The normalization method of face image in the paper is shown. The face image is normalized based on both eyes. The reason for having used the eye for normalization of face image is as follows. The first, eyes are having been easy to perform the normalization about a rotation and a size, compared with other features of face. Next, many researches of extracting the region of an eye are proposed [12], [13]. Therefore, to use eye for normalization of face image is efficient.

First of all, an original image (640×480 pixels, 24bit color) is changed into 8bit into the gray scale image, and median filter is performed in order to remove the noise.Next, center positions of both eye s are extracted. Then, the line segment joining both eyes is rotated so that it matches the horizontal line. Furthermore, the distance between both eyes is made 80 pixels by scale change (Fig.2). Moreover, in order to diminish influence of hair and clothes, an image is cut out. That is, letting the midpoint of the segment joining both eyes be a standard point, the region spreads by 180 pixels (by 90 pixels in right and left direction, respectively) in horizontal direction, and by 225pixels (by 45pixels in the upper part and by 180 pixels in the lower part) in the vertical direction. Finally, the image in a center part of the

face like Fig.3 (180 × 225pixels) is obtained by this processing. Furthermore, in this paper, we make low-resolution images like Fig.4 (12 × 15pixels).

Fig. 3. Normalization of the face images

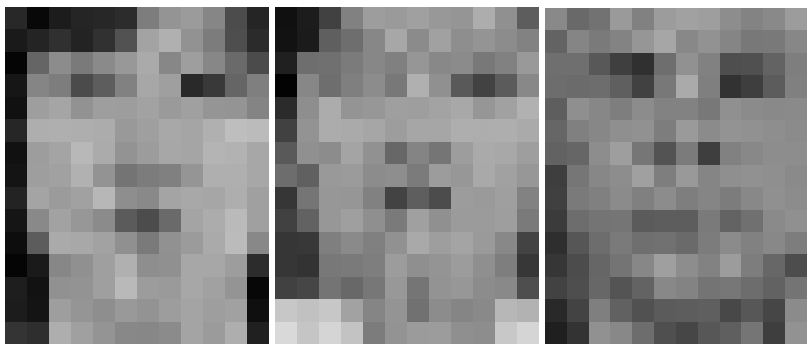


Fig. 4. Low-resolution images

2.3 Apparent age database

In this paper, the apparent age was given by doing the questionnaire survey to 38 subjects. It was thought that the objectivity of the apparent age to which the person evaluated the subjectivity went up by giving the questionnaire survey a lot of subjects. Moreover, the error margin of age estimation for each age can be reduced by having elected various generations and gender questionnaire subjects.

As the questionnaire method, the subject sees the face image after normalization, and the apparent age is given. The face image prepares the one arranged at random, and the subject estimate the age from the edge by intuition sequentially. Fig. 5 represents a actual questionnaire. When the age that obviously outlier was given compared with the age that many other subjects gave the apparent age, it excluded it from the object of the apparent age as a outlier. The apparent age was assumed to be a mean value of the age that the subject had given except outlier. In proposed method, this age is adopted for the teacher data as apparent age.

2.4 Feature data extraction

In this paper, we think that the shape of the face and information on wrinkle are important for the age characteristic. Then, the feature at the age is extracted by the FFT (Fig.6). In this paper, spectrum data was used for a feature data. However, unnecessary data for age estimation are included in the all spectrum. Then we select the feature data for the age estimation by using the GA (see 3.1).

Fig. 5. Actual questionnaire

3. Age estimation system and feature data selection

Feature data by spectrum after the FFT are left a lot of feature data. However, this data has not important data for age estimation. Then, it is necessary to select the feature data. Therefore, we use the GA for feature selection in the proposed method. An age estimation system that used little feature data was developed to the combination of an age estimation system by the supervised SOM. The feature data selection by the GA and an age estimation system by the supervised SOM are described in as following processing.

3.1 Feature data extraction

Age feature are appear in skin texture. Then, the feature data is extracted by Fourier transform.

Genetic algorithm (GA) is an algorithm that technologically imitates the process of the evolution of the living thing. This is widely used as one of the optimization techniques. In this paper, the recognition error margin of the age estimation system by the supervised SOM was used in the fitness function of the GA. Then, the combination of the frequency band regions where the recognition error margin as decreases as possible is selected by the GA.

In this paper, the following fitness function was used. *error* is the age error margin by the supervised SOM, and $error_{\max}$ is the maximum age error margin. The maximum age error margin was determined by the experience to 20.

$$Fitness = 1 - \frac{error}{error_{\max}}$$

Using this fitness function, the feature data to obtain high recognition accuracy can be selected. Moreover, Table 2 shows the parameter of genetic algorithm used by proposed method.

The parameter of the genetic algorithm was decided in consideration of the calculation cost. In this paper, we used a roulette rule about the individual selection. Moreover, we used a uniform crossover method, and we used a elite strategy.

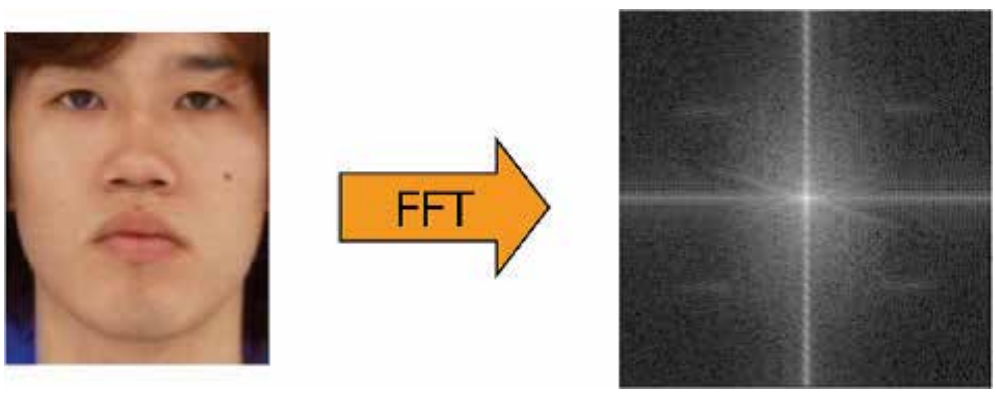


Fig. 6. Feature data extraction by FFT

Individuals	20
Generations	1000
Crossover method	Uniform crossover
Crossover rate	0.8
Mutation rate	0.02
Chromosomes	Decide by the GA

Table 2. Parameter of the GA

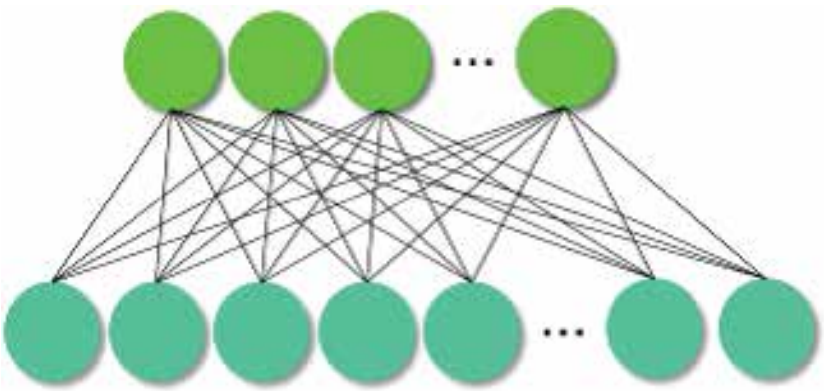


Fig. 7. Supervised SOM

Learning number	10000
Category	50
Input number	Decide by the GA

Table 3. Parameter of the supervised SOM

3.2 Feature data selection

Frequency feature has not only age feature. To extract age feature, feature data selection is necessary. Genetic algorithm (GA) is used for feature data selection.

The supervised self organized map (SOM) is one of the neural net works which have two layers. The network is a combination of an input layer and a rival layer of the processing unit. Furthermore, it is trained in study with the teacher. In this paper, the one dimension supervised SOM is designed to the age estimation (Fig.7). The age is continuously identified by the supervised SOM. The feature data for the age estimation was obtained from genetic algorithm, and is trained by supervised SOM. In this proposed method, the average of the error margin of the teacher data and the output result is assumed to be an age estimation error margin. Moreover, the parameter in the supervised SOM is shown in Table 3. These parameters were obtained from empirically as well as the parameter of the GA. To identify it from 15 to 64 years old continuously at intervals of one year old, the number of neurons of rival layers was assumed to be 50 pieces.

3.3 Age estimation

To estimate continuous age, we use the self organizing map (SOM). Moreover, proposed method is supervised SOM that is one of neural net works which have two layers.

4. Computer simulations

In order to show the effectiveness of the proposed method, we show the simulation examples.
In this paper, only 50 men estimated the age.
Furthermore, we examined the low-resolution images.
Here, the recognition accuracy was calculated by the cross-validation method.
Table 1 shows the age error margin when we use the extracted feature data by the genetic algorithm.
Moreover, age error margin by human's age perception ability is shown at same time.
From this result, the age estimation by few data is thought to be high recognition accuracy.
Moreover, the age error margin closes to human beings.

All feature	9.96 years old
Selected feature	6.8 years old
Human beings	4.94 years old

Table 1. Age error margin

Then, Table 5 shows number of feature data, and the Fig.8 shows the extracted feature point. The data chosen from the symmetry of data becomes below the half. The inside data is low frequency, and the outside data is high frequency. Therefore, both frequencies were chosen to be feature data. In addition, unnecessary data are reduced by the GA about 83%. Moreover, Fig.9 shows the result of inversed FFT by using extracted data. This image shows the possibility that can be estimate the age by blur image. In conclusion, it is thought that the human's atmosphere and shape of face is important in human's age perception, and it can be said that the age estimation by using low resolution image.
Next, we surveyed the weight of supervised SOM after learning. Fig.10 shows that the weight of the supervised SOM. X-axis indicates that the rival layer number, and that means the age. Small number suggest that the young people, and large number indicate that the old person. In addition, Y-axis indicate that the value of weight, and Z-axis suggest that the node number. It is considered to be this figure has that some rules. First, the weight value was decreased in the sixth node numbers by increased age. Furthermore, the weight value was increased in the 12th node numbers by increased age. Similarly, the weight value was increased in the 20th node numbers. From this result, it is considered that human's feature changes continuously along with increased age. Indeed, we surveyed the actual spectrum data. Fig.11 indicates three men's spectrum data (apparent age: 21 years, 39 years, and 55 years). In the sixth node numbers, the largest weight value was a 21 years man, and next large weight value was a 39 years man. Furthermore, in the 12th and 20th node numbers, the largest weight value was a 55 years man and next large weight value was 39 years man. These results were the same as the weight of supervised SOM. Therefore, it can be said that the hypothesis that human's feature changes continuously along with increased age is correct even by actual data.

However, some subjects that have large age error margin exist. In this case, they have difference spectrum data compared with weight of supervised SOM. The reason is considered that the normalized image have unnecessary information for the age estimation, for example, clothes, hair, and so on.

	12 ×15 pixels
Use all data	9.96
Use extracted data	6.8
Human's age perception	4.94

Table 4. Age error margin

	12 ×15 pixels
Use all data	256
Use extracted data	44

Table 5. Number of feature data

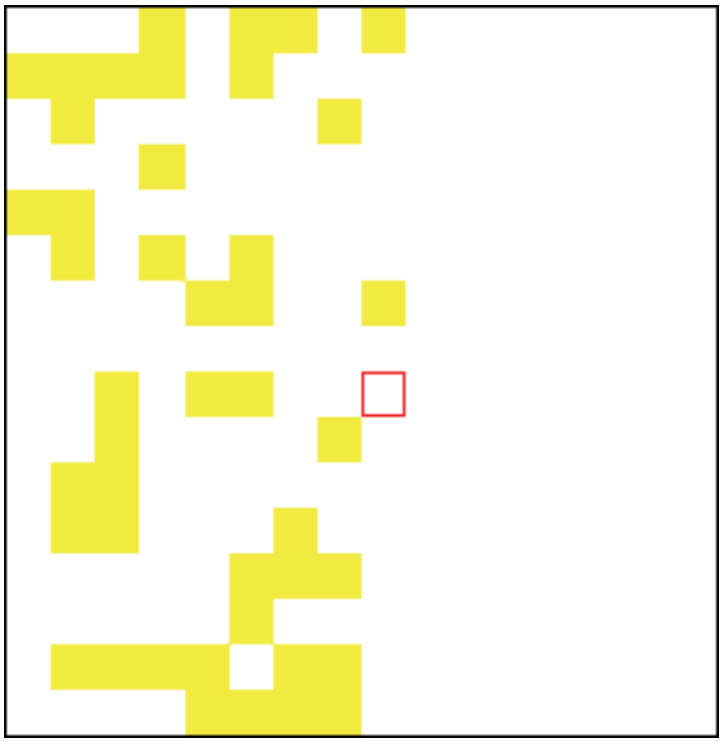


Fig. 8. Extracted feature data

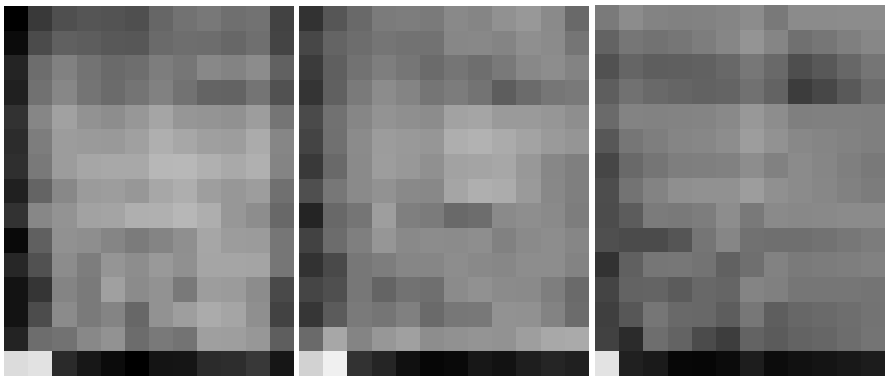


Fig. 9. The result of inversed FFT

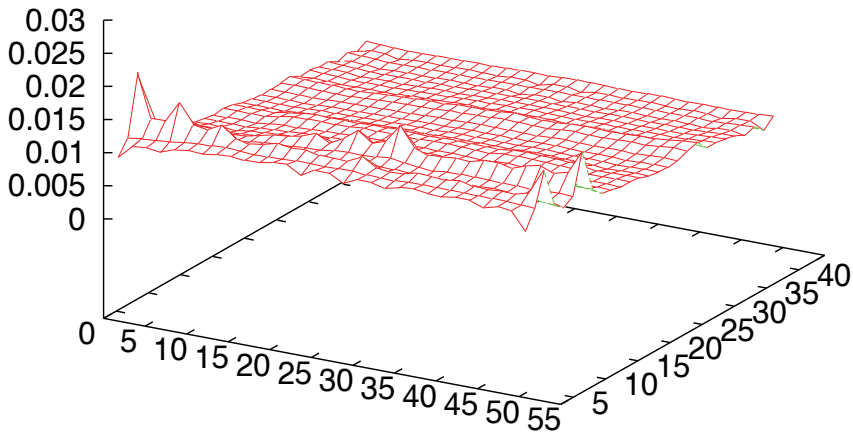


Fig. 10. Weight of the supervised SOM

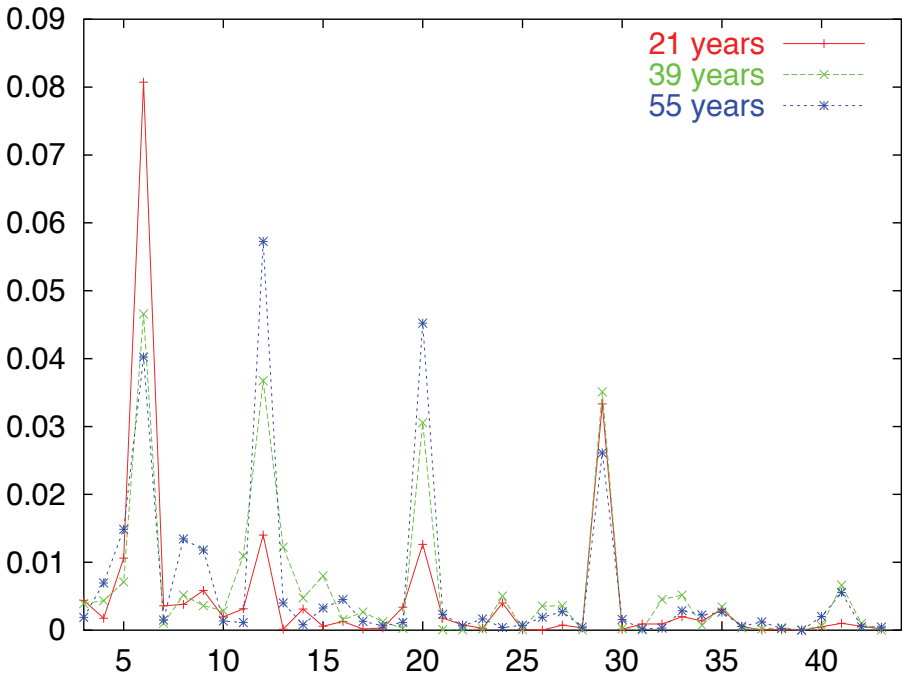


Fig. 11. Spectrum data

5. Conclusions

In this paper, we proposed a new age estimation system by the GA and the supervised SOM.
Then, the age is estimated continuously by the supervised SOM.
The GA is used for the feature selection based on the human sense.
To verify the effectiveness of the proposed method, we show the simulation examples.
From this result, unnecessary data are drastically reduced by the GA and recognition accuracy that closes to human beings.
Furthermore, the recognition accuracy by the low-resolution images has higher accuracy than the normalized images.
Indeed, it is difficult for the computer to get the same perception as human.
However, in this paper, not to mention complete, we unlimitedly achieved.
It is thought that this approached the research for the computer to have man's sense again by one.
Finally, this research is a one to make the computer judge the sense that man judges externals (human's perceptions).
It is thought it is possible in the near future though this is difficult.
From these results, we can get the relatively good results from the view point of the few data recognition and were able to obtain a result almost similar to human's perception.
Therefore, it was confirmed that the proposed method works well.
In the future works, we analyze the cause of the misidentification knowledge that exists now, and we improve the recognition accuracy of an age estimation system.

Moreover, we search for optimal image size because of the high frequency has been extracted in the low resolution image.
Furthermore, we thought the hypothesis that human's feature changes continuously along with increased age.
Therefore, we'll propose the statistical estimation method.
In addition, we'll increase the number of subject, and we'll survey the women subjects.

6. References

J. T. Todd, L. S. Mark, R. E. Shaw, and J. B. Pittenger, "The perception of human growth", *Scientific American Perception*, vol.242, PP.106-114, 1980.

L.S. Mark, J.B. Pittenger, H.Hines, C. Carello, R.E. shaw, and J.T. Todd, "Wrinkling and head shape as coordinated sources of age-level information", *Perception & Psydhophysics*, vol.27, pp.117-124, 1980.

Tsuneo Kanno, Masakazu Akiba, Yasuaki Teramachi, Hiroshi Nagahashi, Takeshi Agui, "Classification of age Group Based on Facial Images of Young Males by Using Neural Networks", *IEICE Trans. Inf. & Syst*, Vol.E84-D, No.8, pp.1094-1101, August 2001.

Y. H. Kwon and N. D. V. Lobo, "Age classification from facial images", *CVPR'94*, pp.762-767, Seattle, US, June 1994.

D.M. Burt and D.I. perrett, "Preception of age in adult daudasian male faces: computer graphic manipulation of shape and colour information", *Perception*, Vol.259, No.1355, pp.137-143, January 1995.

K. Ueki, T. Hayashida, and T.kobayashi, "Subspace-based age-group classification using facial images under various lighting condhitions", *Proc. of IEEE Intl. Conf. on Automatic Face and Gesture Recognition*, PP.43-48, 2006.

Takayuki Fujiwara, Hiroyasu Koshimizu, "Age and Gender Estimations by Modeling Statistical Relationship among Faces", *KES2003*, LNAI2774, pp.870-876, 2003.

Noriko Nagata, Seiji Inokuchi, "Subjective Age Obtained from Facial Images -How Old We Feel Compared to Others", *KES2003*, LNAI2774, pp.877-881, 2003.

P.A. George and G.J. Hole, "Factors influencing the accuracy of age estimates of unfamiliar faces", *Perception*, vol.24, no.1, pp.1095-1073,Feb. 1995.

M. K. Yamaguchi, T. kato, and S. Akamatsu, "Relationship between physical traits and subjective impressions of the face - Age and sex information", *IEICE Trans.*, vol. J79-A, no.2, pp.279-287, Feb.1996.

http://www.hoip.softpia.pref.gifu.jp

S. Kawato, N. Tetsutani, "Circle-Frequency Filter and its Application", *Proc. Int. Workshop on Advanced Image Technology*, pp.217-222, 2000.

T. Kawaguchi, D. Hikada, and M. Rizon, "Detection of the eyes from human faces by hough transform and separability filter", *Proc. of ICIP 2000*, pp.49-52, 2000.

Part 7

Use of SOM in the Mechanical and Manufacturing Engineering

25

Parametric and Robust Optimization Study of a Vibration Absorber with a Generalized Cubic, Quadratic and Non Integer Nonlinearities of Damping and Stiffness

M.–Lamjed Bouazizi[1] and S. Ghanmi[1] and R. Nasri[2]
[1]*Preparatory Engineering Institute (IPEIN), University of 7 November, 8000 M'rezgua –Nabeul,*
[2]*Research Unity of Materials Engineering, National Engineering School of Tunis, University of Tunis El Manar, BP37,1002 Le Belvedere Tunis, Tunisia*

1. Introduction

In order to reduce the vibrations in the revolving machines and the structures in general, the dynamic absorbers are often used in different mechanical and civil applications (Crankshaft, rotor of the wings of a helicopter, beams, etc).
The absorber is a mechanical oscillator called auxiliary system and it's added to the main vibrating structure, in order to carry out a transfer of energy from the main system to the auxiliary system. This latter can either be added on the main system, or directly envisaged in design.
A linear model is often insufficient to describe correctly the dynamical behaviour of a system, and then it's natural to introduce non linear structure models. To solve these non linear problems, iterative numerical algorithms are often used.
In order to approach the non linear dynamic behaviour of mechanical systems, many methods were proposed in the literature. As for the mechanical systems assimilated to a single dof system, one can mentioned different works. Whiston [1] studied the linear response of a mechanical oscillator with a single dof preloaded and subjected to a harmonic excitation. Natsiavas [2] studied the response of a vibration absorber applied to a machine with a nonlinear cubic stiffness of the Duffing type. In this chapter, numerical results are given with many parameter combinations. This leads to the improvement of the vibration absorber. The stability response, for periodic motion of harmonically excited single dof linear system with piecewise linear characteristics was studied by Natsiavas [3]. Chung et al.[4] carried out the extension of the method of incremental perturbation on a strongly nonlinear non-autonomous oscillator.
Concerning the discrete systems with several dof, many works were interested in the nonlinearity. Vakakis and Paipetic [5] studied the effect of the absorber viscosity on a conservative system with several dof. Optimal values for the parameters describing the behaviour of the absorber are determined. Natsiavas [6] discussed the response of a strongly

nonlinear system with several dof where the absorber is modeled by a linear spring and a residual force. Natsiavas and Tratskas [7] have studied a system with two nonlinear dof in translation and rotation motions by multi-scale method in time. Verros and Natsiavas [8] studied an oscillator with two dof with a cubic stiffness nonlinearity subjected to a harmonic charge. For this purpose, they used the perturbation method in order to study the effect of the parameters on the stability of the periodic solution. Zhu et al. [9] have studied a discrete two dof system with two cubic nonlinearities of stiffness and damping subjected to a harmonic excitation. They have used the method of the variation of the constant in order to obtain the periodic, quasi periodic and chaotic motions. Erkus [10] has accomplished a comparative study of the various numerical techniques of resolution for nonlinear structures, based on Newmark schema. Roy and Kumar [11] presented a method of multi-stage transverse linearization applied to a nonlinear dynamic system. El Bessiouny [12] studied effect of the quadratic and cubic nonlinearity of an elastomer shock absorber used in the torsion vibrations in an internal combustion engine. The stability of the solution is studied according to nonlinear terms. The multi-scale method was used in this work.

The majority of the optimization problems are of multi-objective nature. For their resolutions, there are several approaches of evolutionary algorithm. Schaffer [13] proposed an extension of a standard Genetic Algorithm to a modified one called Vector Evaluated Genetic Algorithm "VEGA".

Zitzler and Thiele [14] proposed a multi-objective optimization method which uses the concept of Pareto to compare the solutions. Srivinas and Deb [15] proposed an evolutionary algorithm of NSGA type (Not dominated Sorting Genetic Algorithms) based on the concept of Pareto dominance.

This chapter is based on the paper of Bouazizi et al.[16]. Firstly a two dof mechanical system composed of a one dof main system and a one dof absorber of vibration is proposed. The case of non integer power non linearity is considered also in this work in order to show the difference with integer cases.

The global system comprises generalized nonlinear stiffness and shock absorbers (cubic non linearity combined with a higher or lower power nonlinearities: 2,4 and 1.5) with a given proportioning.

For the resolution the unconditionally stable Newmark iterative scheme is used.

Here, one can seem for the non linear forces in the main system and in the absorber according to displacements and system trajectory in order to study the influence of some parameters and the type of non linearity.

Finally, the errors curves are found witch make it possible to conclude about the speed of convergence of the model compared to the importance of results.

A parametric study of the system as well as of the load and nonlinearities are carried out.

Secondly, the presence of uncertainties on the design parameters of a system can influence the results of the used models. Thus, one can use a robust multi-objective optimization methodology with respect to uncertainties on the design parameters. This methodology will be applied to a mechanical system comprising an absorber of vibration and equipped with nonlinearities of stiffness and damping of cubic, quadratic and non entire. This mixing of one of the two last nonlinearities with the cubic case constitutes our generalized non linearity.

The step of multi-objective optimization consists in seeking the first Pareto front of several linear and nonlinear objective functions by using genetic algorithm "NSGA ". Those make it possible to find the optimal nonlinear displacements.

The obtained solution guarantees the acceptable compromise of optimality/robustness. The robustness function is then introduced as supplementary objective function to each cost function Bouazizi et al. [16], Ghanmi et al. [17] and Ait Brik et al [18].

2. Basic equations

2.1 Introduction

One can consider a mechanical system called main system brought back to only one dof. An absorber with only one dof is assembled on this system. The main system is characterized by a mass m_1, a linear spring of stiffness k_1, a linear shock absorber or damping c_1, two stiffness nonlinearities of respectively $k_1^{'}$ with variable power r and $k_1^{''}$ cubic and two other similar nonlinearities of damping with the two respective coefficients of nonlinearities $c_1^{'}$ and $c_1^{''}$. It is subjected to a harmonic load of variable amplitude and fixed pulsation ω corresponding approximatively to the natural pulsation of the main system. The mechanical characteristics of the absorber are similar to those of the main system with index 2 instead of 1. The absorber parameters are then: m_2, k_2, c_2, $k_2^{'}$, $k_2^{''}$, $c_2^{'}$ and $c_2^{''}$.

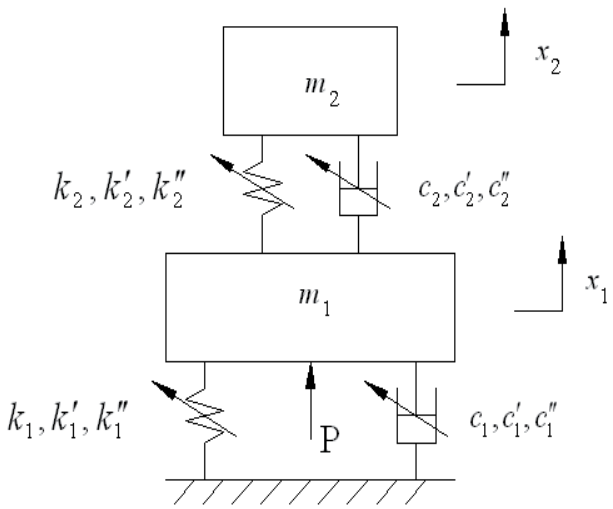


Fig. 1. Basic model

Figure 1 presents the global system of study. x_1 is defined as the absolute vertical displacement of the mass m_1 and x_2 is the vertical displacement of the mass m_2 relatively to the mass m_1.

2.2 Analytical formulation

From the Lagrange method, the equations of motion for the system can be rearranged as follows:

$$\begin{aligned} &(m_1+m_2)\ddot{x}_1 + m_2\ddot{x}_2 + f_1(x_1) + R_1(x_1) = P_0\cos\omega t \\ &m_2\ddot{x}_1 + m_2\ddot{x}_2 + f_2(x_2) + R_2(x_2) = 0 \end{aligned} \tag{1}$$

where:

$f_i(x_i) = k_i x_i + k_i^{'} x_i^r + k_i^{''} x_i^3, (i = 1,2),$ are the generalized stiffness nonlinear forces.

$R_i(x_i) = c_i \dot{x}_i + c_i^{'} x_i^{r-1} \dot{x}_i + c_i^{''} x_i^2 \dot{x}_i, (i = 1,2),$ are the generalized damping nonlinear forces.

Further, the nonlinear force on a mass m_i is referred to as the addition of the two nonlinear forces mentioned previously.

The positive real parameter r is introduced in order to consider various situations of nonlinearities i.e.:

- $r = 1$ or $r = 3$: cubic nonlinearity.
- $r = 2$: quadratic nonlinearity superposed to cubic nonlinearity.
- $r = 1.5$: non integer nonlinearity superposed to cubic nonlinearity.

These last three situations constitute what we call generalized nonlinearity.

The equations of motion (1) can be written in a matrices form:

$$[M]\{\ddot{u}\} + [C_{nl}]\{\dot{u}\} + [K_{nl}]\{u\} = \{F\} \tag{2}$$

where:

$\{u\} = \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix}$, $\{\dot{u}\} = \begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix}$, $\{\ddot{u}\} = \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix}$ and $\{F\} = \begin{Bmatrix} P_0 \cos \omega t \\ 0 \end{Bmatrix}$ are respectively the displacement, velocity, acceleration and external force vector, and $[M] = \begin{bmatrix} m_1 + m_2 & m_2 \\ m_2 & m_2 \end{bmatrix}$,

$[C_{nl}(u)] = \begin{bmatrix} c_1 + c_1^{'} x_1^{r-1} + c_1^{''} x_1^2 & 0 \\ 0 & c_2 + c_2^{'} x_2^{r-1} + c_2^{''} x_2^2 \end{bmatrix}$ and

$[K_{nl}(u)] = \begin{bmatrix} k_1 + k_1^{'} x_1^{r-1} + k_1^{''} x_1^2 & 0 \\ 0 & k_2 + k_2^{'} x_2^{r-1} + k_2^{''} x_2^2 \end{bmatrix}$

are respectively the mass, nonlinear damping and nonlinear stiffness matrix. Further, adimensional writing is used by carrying out the following changes of variables:

$$x_{sta} = \frac{P_0}{k_1}, \ \omega_1^2 = \frac{k_1}{m_1}, \bar{x}_1 = \frac{x_1}{x_{sat}},$$

$$\bar{x}_2 = \frac{x_2}{x_{sat}}, \ \bar{m}_2 = \frac{m_2}{m_1}; \bar{k}_2 = \frac{k_2}{k_1} \ ,$$

$$\bar{k}_i^{'} = \frac{k_i^{'}}{k_i} x_{sta}^{(r-1)}, \quad \bar{k}_i^{''} = \frac{k_i^{''}}{k_i} x_{sta}^2, \tag{3}$$

$$\bar{c}_i^{'} = \frac{c_i^{'}}{c_i} x_{sta}^{(r-1)}, \bar{c}_i^{''} = \frac{c_i^{''}}{c_i} x_{sta}^2 \quad (i = 1,2).$$

$$\left[\bar{M}\right]\{\ddot{\bar{u}}\} + \left[\bar{C}_{nl}(\bar{u})\right]\{\dot{\bar{u}}\} + \left[\bar{K}_{nl}(\bar{u})\right]\{\bar{u}\} = \{\bar{F}\} \tag{4}$$

After dividing with ($k_1 x_{sta}$), the equation of motion (2) becomes:

where:

$\{\bar{u}\} = \begin{Bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{Bmatrix}, \{\dot{\bar{u}}\} = \frac{1}{\omega_1}\begin{Bmatrix} \dot{\bar{x}}_1 \\ \dot{\bar{x}}_2 \end{Bmatrix}, \quad \{\ddot{\bar{u}}\} = \frac{1}{\omega_1^2}\begin{Bmatrix} \ddot{\bar{x}}_1 \\ \ddot{\bar{x}}_2 \end{Bmatrix}$ and $\{\bar{F}\} = \begin{Bmatrix} \cos\omega t \\ 0 \end{Bmatrix}$ are respectively the displacement, velocity, acceleration and external force adimensional vector.

$[\bar{M}] = \begin{bmatrix} 1+\bar{m}_2 & \bar{m}_2 \\ \bar{m}_2 & \bar{m}_2 \end{bmatrix}$, $[\bar{C}_{nl}(\bar{u})] = 2\begin{bmatrix} \alpha_1(1+\bar{c}_1'\bar{x}_1^{(r-1)}+\bar{c}_1''\bar{x}_1^2) & 0 \\ 0 & \alpha_2\sqrt{\bar{k}_2\bar{m}_2}(1+\bar{c}_2'\bar{x}_2^{(r-1)}+\bar{c}_2''\bar{x}_2^2) \end{bmatrix}$

and $[\bar{K}_{nl}(\bar{u})] = \begin{bmatrix} 1+\bar{k}_1'\bar{x}_1^{(r-1)}+\bar{k}_1''\bar{x}_1^2 & 0 \\ 0 & \bar{k}_2(1+\bar{k}_2'\bar{x}_2^{(r-1)}+\bar{k}_2''\bar{x}_2^2) \end{bmatrix}$ are respectively the mass, the nonlinear damping and the nonlinear stiffness adimensional matrix.

α_1 and α_2 are respectively the damping ratio of the main system and the absorber.

For the numerical resolution, we need the nonlinear terms of the main system $(\bar{k}_1', \bar{k}_1'', \bar{c}_1', \bar{c}_1'')$, nonlinear terms of the auxiliary system (absorber) $(\bar{k}_2', \bar{k}_2'', \bar{c}_2', \bar{c}_2'')$ and basic terms $\bar{m}_2, x_{sta}, \omega_1, \alpha_1, \alpha_2$ and $\bar{k}_2$.

The initial conditions are given by :

$$\{\bar{u}(0)\} = \begin{Bmatrix} \bar{x}_{10} \\ \bar{x}_{20} \end{Bmatrix}, \{\dot{\bar{u}}(0)\} = \frac{1}{\omega_1}.\begin{Bmatrix} \dot{\bar{x}}_{10} \\ \dot{\bar{x}}_{20} \end{Bmatrix} \text{ and } \{\bar{F}(0)\} = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix}.$$

The beginning of computation, we start by the computation of the initial acceleration:

$$\{\ddot{\bar{u}}(0)\} = \begin{Bmatrix} \ddot{\bar{x}}_{10} \\ \ddot{\bar{x}}_{20} \end{Bmatrix} = [\bar{M}]^{-1}\{-[\bar{C}_{nl}(0)]\{\dot{\bar{u}}(0)\} - [\bar{K}_{nl}(0)]\{\bar{u}(0)\} + \{\bar{F}(0)\}\} \tag{5}$$

The resolution of equation (4) can be obtained by a numerical integration with using the unconditionally stable Newmark scheme.

At the step n where $t = n.\Delta t$ (Δt is the time integration step), the equation (4) can be written:

$$[\bar{M}]\{\ddot{\bar{u}}_n\} + [\bar{C}_{nl}(\bar{u}_n)]\{\dot{\bar{u}}_n\} + [\bar{K}_{nl}(\bar{u}_n)]\{\bar{u}_n\} = \{\bar{F}_n\} \tag{6}$$

where: the index n corresponds to the value in n^{th} step.

The Newmark scheme consists in approximating the adimensional displacement and velocity vectors as follows:

$$\begin{aligned} \{\bar{u}_{n+1}\} &= \{\bar{u}_n\} + \Delta t.\omega_1\{\dot{\bar{u}}_n\} + \frac{\Delta t^2}{2}.\omega_1^2[(1\text{-}2\beta)\{\ddot{\bar{u}}_n\} + 2\beta\{\ddot{\bar{u}}_{n+1}\}] \\ \{\dot{\bar{u}}_{n+1}\} &= \{\dot{\bar{u}}_n\} + \Delta t.\omega_1\ [(1\text{-}\gamma)\{\ddot{\bar{u}}_n\} + \gamma\{\ddot{\bar{u}}_{n+1}\}] \end{aligned} \tag{7}$$

The accuracy and the stability of this scheme depends on the values of the two parameters β and γ. In our simulation, we used the habitual parameters ($\gamma = \frac{1}{2}$, $\beta = \frac{1}{4}$) corresponding to the called average acceleration method (unconditionally stable scheme).

The process consists in defining the indicators $\{\tilde{u}_{n+1}\}$ and $\{\dot{\tilde{u}}_{n+1}\}$ according to the terms known at the moment $t = n.\Delta t$ while taking $\{\ddot{\bar{u}}_{n+1}\} = \{0\}$:

$$\begin{aligned} \{\tilde{u}_{n+1}\} &= \{\bar{u}_n\} + \Delta t\ \omega_1 \{\dot{\bar{u}}_n\} + \frac{\Delta t^2}{2}\omega_1^2\ [(1\text{-}2\beta)\{\ddot{\bar{u}}_n\}] \\ \left\{\dot{\tilde{u}}_{n+1}\right\} &= \{\dot{\bar{u}}_n\} + \Delta t.\omega_1[(1\text{-}\gamma)\{\ddot{\bar{u}}_n\}] \end{aligned} \tag{8}$$

One can define the residual equation (Res) as a function of $\{\bar{u}_{n+1}\}$ at the moment $t_{n+1} = (n+1)\Delta t$ as following:

$$\text{Res}(\{\bar{u}_{n+1}\}) = \left[\bar{M}\right]\{\ddot{\bar{u}}_{n+1}\} + \left[\bar{C}(\bar{u}_{n+1})\right]\{\dot{\bar{u}}_{n+1}\} + \left[\bar{k}(\bar{u}_{n+1})\right]\{\bar{u}_{n+1}\} - \{\bar{F}_{n+1}\} = \{0\} \tag{9}$$

The nonlinear equation (9) can be solved in an iterative way by the method of Newton-Raphson in the following way.
Let $\{\bar{u}^k_{n+1}\}$, $\{\dot{\bar{u}}^k_{n+1}\}$ and $\{\ddot{\bar{u}}^k_{n+1}\}$) respectively the approximation of adimensional displacement, velocity and acceleration vector obtained in the k^{th} iteration of the time step t_{n+1}. Those can be corrected, for the next iteration, respectively in the form:

$$(\{\bar{u}^k_{n+1}\} + \{\Delta\bar{u}_{n+1}{}^k\}, \{\dot{\bar{u}}^k_{n+1}\} + \{\Delta\dot{\bar{u}}_{n+1}{}^k\}, \{\ddot{\bar{u}}^k_{n+1}\} + \{\Delta\ddot{\bar{u}}_{n+1}{}^k\}).$$

The end of the iteration cycle is carried out, when the residue of displacement falls below a certain precision level initially fixed.
In the numerical simulation (section 3), several cases will be study.
In the first time, two particular cases, the cubic nonlinear stiffness (Duffing type) and the cubic nonlinear damping (Van Der Pol type), are studied.
In the second time, the mixed of the previous cubic nonlinear cases (damping and stiffness) is considered.
Finally, the mixed generalized nonlinearity cases (power 2 "quadratic" and cubic, and power 1.5 and cubic) are studied.

3. Simulation results and discussion

In this section, we are interested in first part to the vibratory behaviour of the global two dof system through the representation of the following curves:

- $X_1(t)$: Displacement of the main system (1) according to time,
- $X_2(t)$: Relative displacement between the absorber (2) and the main system according to time,
- $V_1(X_1)$: Phase diagram of the main system 1,
- $V_2(X_2)$: Phase diagram of the absorber,
- $NL-F1$: Nonlinear force 1 of the system (1) according to X_1,
- $NL-F2$: Nonlinear force 2 of the absorber according to X_2,
- $X_2(X_1)$: System trajectory,

- ε : Relative error on displacement after convergence (limit fixed at 10^{-6}) according to time.

These curves are given for the four indicated nonlinearity cases mentioned above.

In second part, one is interested in optimization of the absorber by seeking its linear and nonlinear optimal characteristics: $\bar{k}_2$, $\bar{k}_2^{'}$, $\bar{k}_2^{''}$ *et* $\bar{m}_2$ with minimization of the maximum nonlinear displacement $X_1(t)$.

The non dimensional results are converted finally in physics results when the displacement and the velocity are multiplied by (x_{sta}) in (m) also the velocity well be in m/s.

The curves give only the non dimensional results and need to be converted are mentioned.

In this section, we will take for computation convenience and practical considerations: $\bar{m}_2 = 0.01$, $x_{sta} = 0.05\, m$, $\omega_1 = 70\ rad/s$, $\Delta t = 3.10^{-4}\, s$, $\bar{k}_2 = 0.01$ and $\alpha_1 = \alpha_2 = 0.001$.

It is noted that the excitation pulsation is selected close to the natural pulsation of the main system witch is experimented in another submitted paper. The absorber is the used only in the resonance domain in this work.

The time step is chosen to have a good convergence.

With a less step time we find the same results.

The initials conditions are equal to zero but when we modify them the results don't change

3.1 Cubic nonlinear stiffness (Duffing case)

The nonlinear mechanical characteristics of the system are: $\bar{k}_1^{'} = \bar{k}_1^{''} = \bar{k}_2^{'} = \bar{k}_2^{''} = 250$ and $\bar{c}_1^{'} = \bar{c}_1^{''} = \bar{c}_2^{'} = \bar{c}_2^{''} = 0$. It should be noted that the absolute physical characteristics in this case, have the value ($\frac{k_1^{'}}{k_1} = \frac{k_1^{''}}{k_1} = \frac{k_2^{'}}{k_2} = \frac{k_2^{''}}{k_2} = 10^5$) in international system (SI).

Figure 2a illustrates the nonlinear responses in displacements of the main system (dof 1) and of the absorber (dof 2). It is noted that the main system vibrations are weaker than those of the absorber.

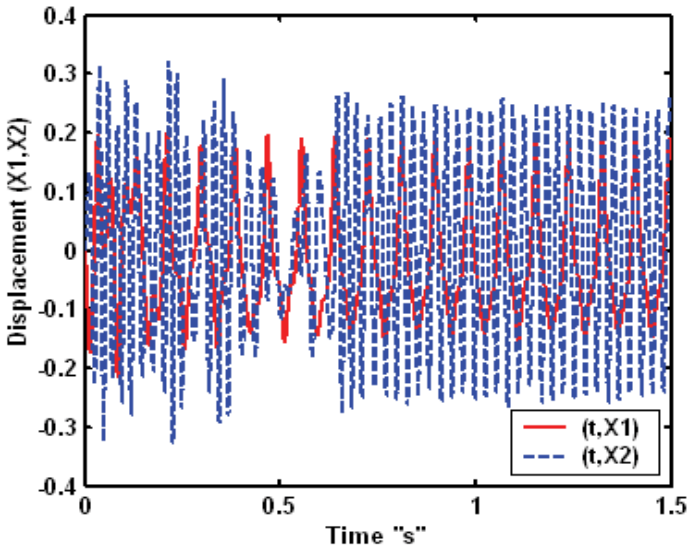


Fig. 2a. Nonlinear temporal responses "Duffing case"

After a transitory period of time, one notes a more regular vibration for the absorber and a phenomenon of periodic vibration for the main system and the absorber but with different periods. The period of the main system is similar to that of the force excitation but that of the absorber is about three times lower then the precedent period.
Figure 2b illustrates the two phase curves corresponding to the Duffing case where attractors are visible only for the main system but for the absorber we note a limit cycle orbit after the transitory period.

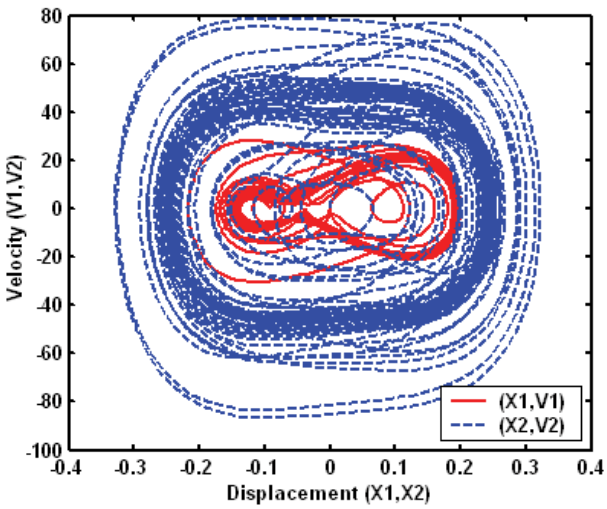


Fig. 2b. Phase diagrams "Duffing case"

The figure 3 a and b illustrate the nonlinear forces on each mass according to corresponding displacement. One notes a hysteretic cubic cycle, and an inflection point corresponding to the steady balance point.

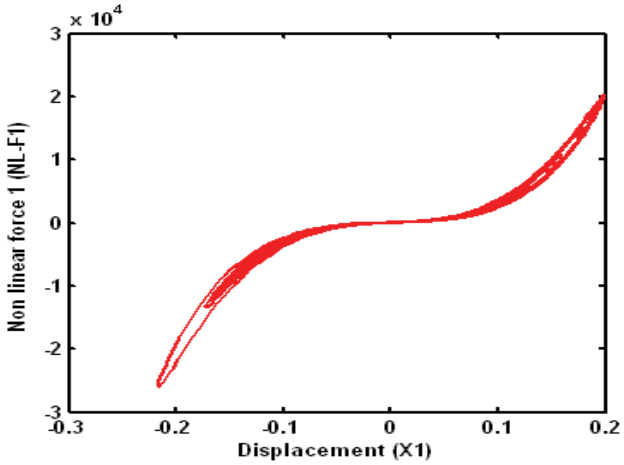


Fig. 3a. Nonlinear forces: (X1, NL-F1) "Duffing case"

This hysteresis has a low width caused by the weak linear damping; it is more visible for the absorber. In the same way the cubic form of the nonlinear force is more visible for the absorber than for the main system. One notes also the appearance of a narrow level stretch around the stable point especially for the main system.

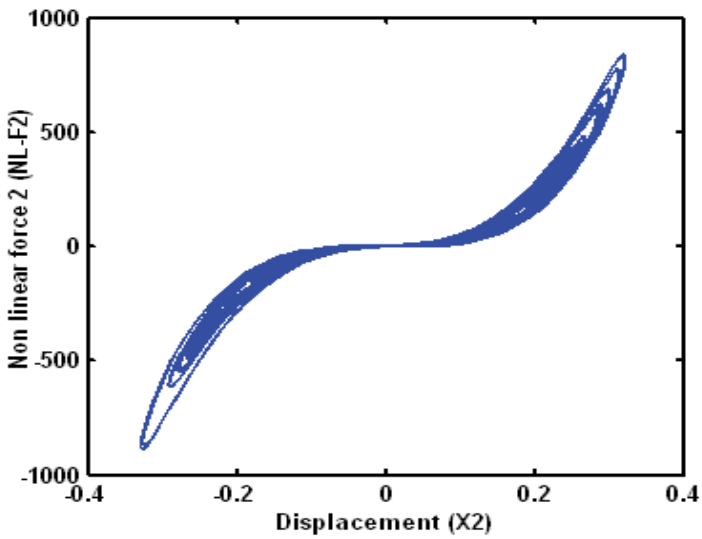


Fig. 3b. Nonlinear forces: (X2, NL-F2) "Duffing case"

Figure 4a illustrates the very complex trajectory of the global system in the plan (X_1, X_2) for these proportioning.

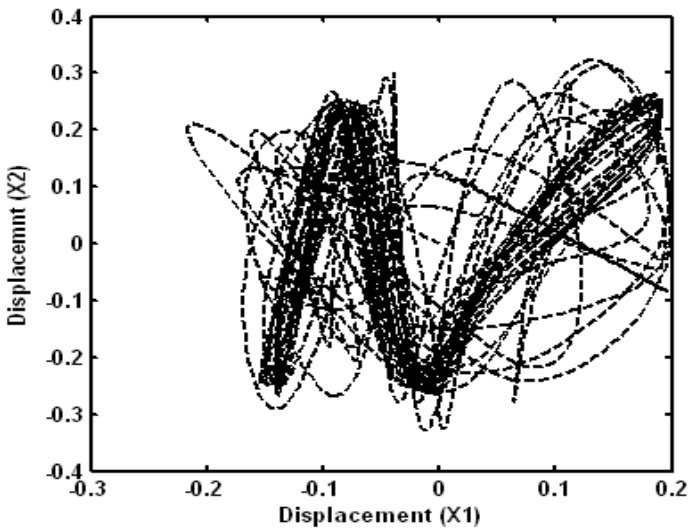


Fig. 4a. System trajectory in the plan (X1, X2) "Duffing case"

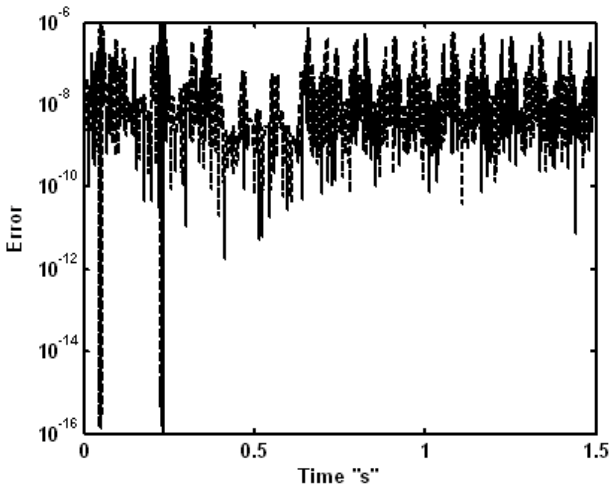


Fig. 4b. Error curve "Duffing case"

Figure 4b shows the relative error curve of displacement. It is noted that the errors are satisfactory compared to the selected limit (10^{-6}) during the iterative process, which proves a good convergence for the selected model and parameters.

3.2 Cubic nonlinear damping (Van Der Pol case)

The nonlinear mechanical characteristics of the system are: $\overline{k}_1^{'} = \overline{k}_1^{''} = \overline{k}_2^{'} = \overline{k}_2^{''} = 0$ and $\overline{c}_1^{'} = \overline{c}_1^{''} = \overline{c}_2^{'} = \overline{c}_2^{''} = 250$. It should be noted that the absolute physical characteristics in this case, have the value ($\frac{c_1^{'}}{c_1} = \frac{c_1^{''}}{c_1} = \frac{c_2^{'}}{c_2} = \frac{c_2^{''}}{c_2} = 10^5$) in international system (SI) and also these parameters are selected among several tests, guaranteeing useable results on the one hand and of good convergence on the other hand.

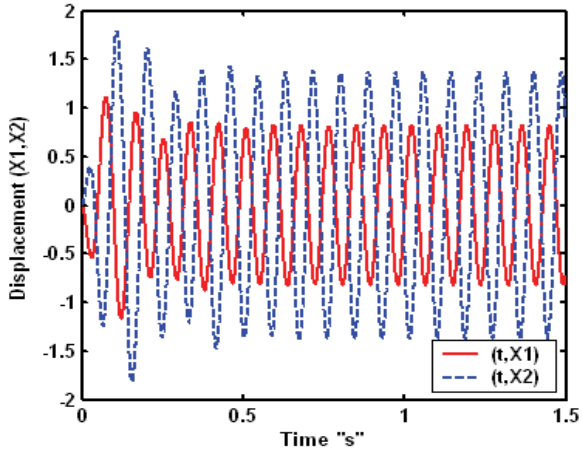


Fig. 5a. Nonlinear temporal responses "Van Der Pol case"

Figure 5a illustrates the responses in displacements of the main system and of the absorber. It is noted that the responses of the two dof are practically harmonic and in phase opposition, in which the pulsation is close to that of excitation. The behaviour resembles for this case and after a transitory period to that of a linear system. The absorber causes a notably decrease for the vibrations of the main system.

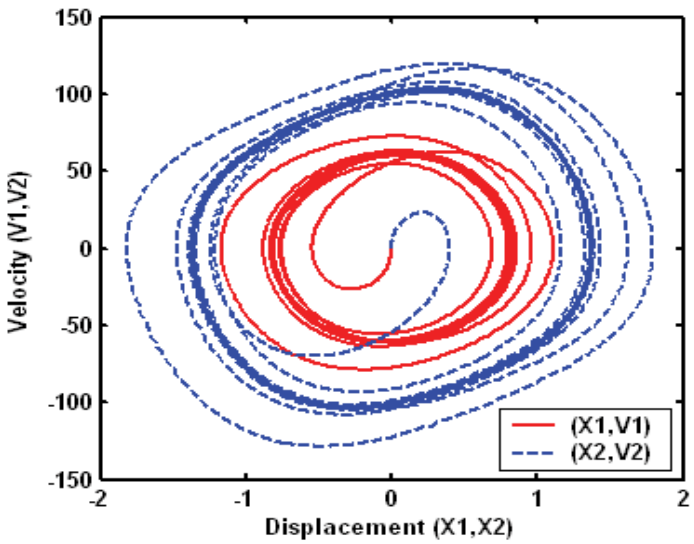


Fig. 5b. Phase diagrams "Van Der Pol case"

Consequently, the phase diagram for each dof (figure 5b) indicates a cycle limit meaning stability, these two cycles are complementary.

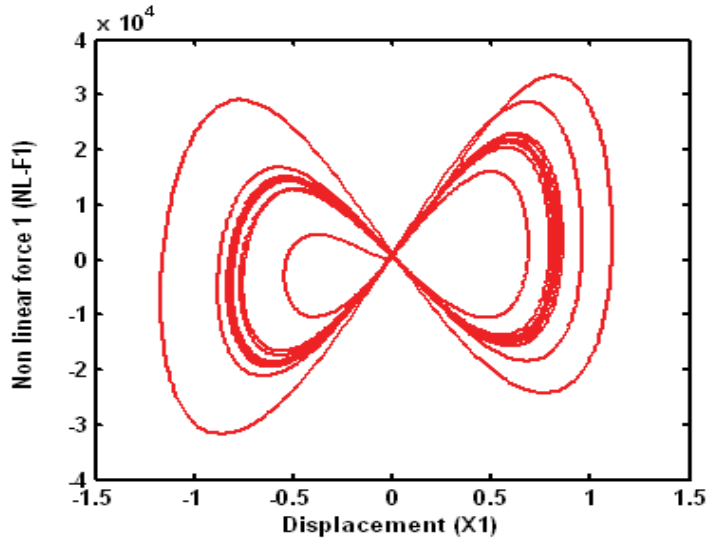


Fig. 6a. Nonlinear forces: (X1, NL-F1) "Van Der Pol case"

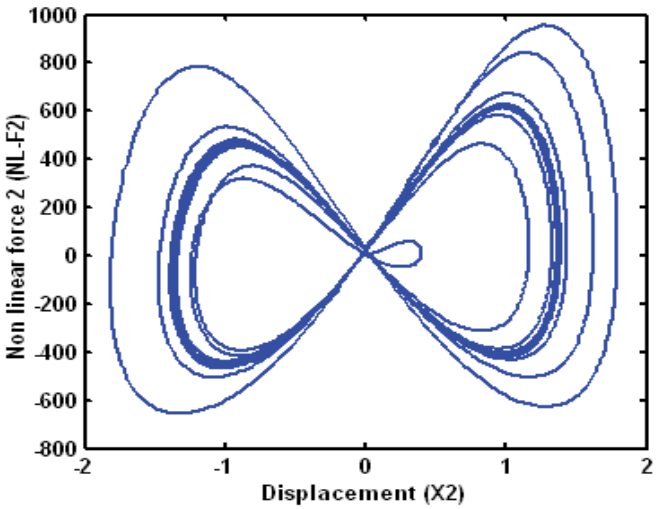


Fig. 6b. (X2, NL-F2) "Van Der Pol case"

The figure 6 a and b illustrate the nonlinear forces exerted on each mass according to corresponding displacement. One notes a broad opening hysteresis caused by strongly nonlinear damping for each dof. It is practically symmetrical compared to the steady balance position. It is also noted that the forces evolve in phase opposition for the two dof as well as the displacements.

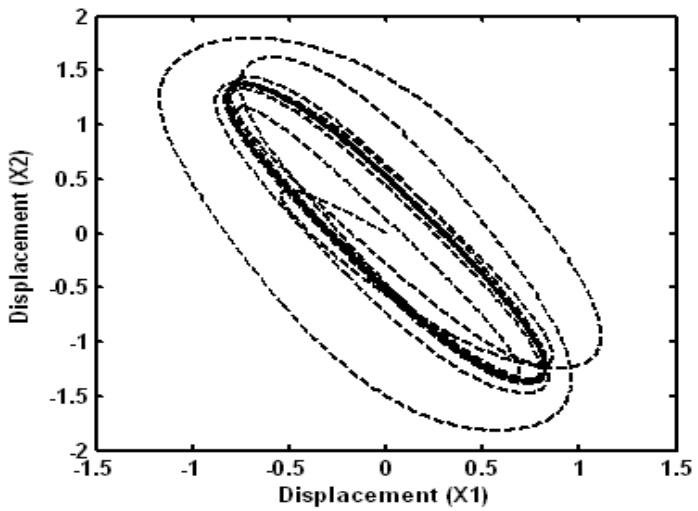


Fig. 7a. System trajectory in the plan (X1, X2) "Van Der Pol case"

Figure 7a illustrates a quasi periodic and elliptic stable trajectory of the global system in the plan (X_1, X_2). Figure 7b shows the relative error curve. It is noted that the errors are close to 10^{-6} during the iterative process, which proves a difficult convergence compared to the Duffing case for the selected model and parameters. If one continues to increase the damping proportioning, convergence becomes difficult.

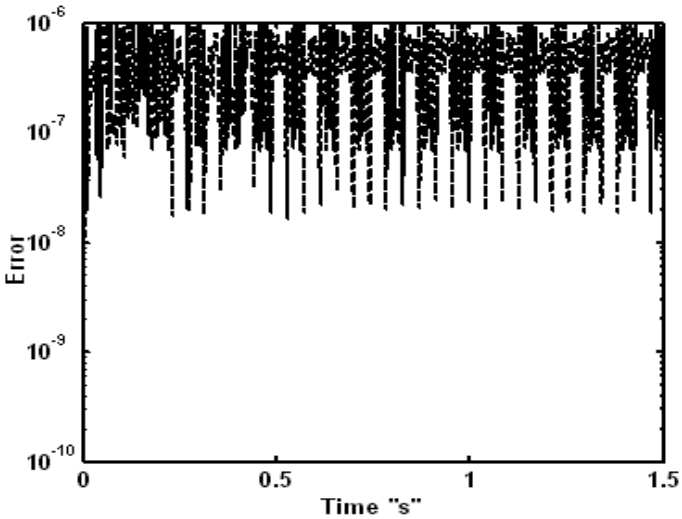


Fig. 7b. Error curve "Van Der Pol case"

3.3 Cubic nonlinear damping and stiffness: $r = 3$

In this subsection and for all that will follow, we consider mixed (damping and stiffness) non linearity cases. The nonlinear mechanical characteristics of the system are: $\overline{k}_1^{'} = \overline{k}_1^{''} = \overline{k}_2^{'} = \overline{k}_2^{''} = 250$ and $\overline{c}_1^{'} = \overline{c}_1^{''} = \overline{c}_2^{'} = \overline{c}_2^{''} = 250$. This choice respects the same nonlinear damping and stiffness proportioning previously used in (3.1) and (3.2) subsections.

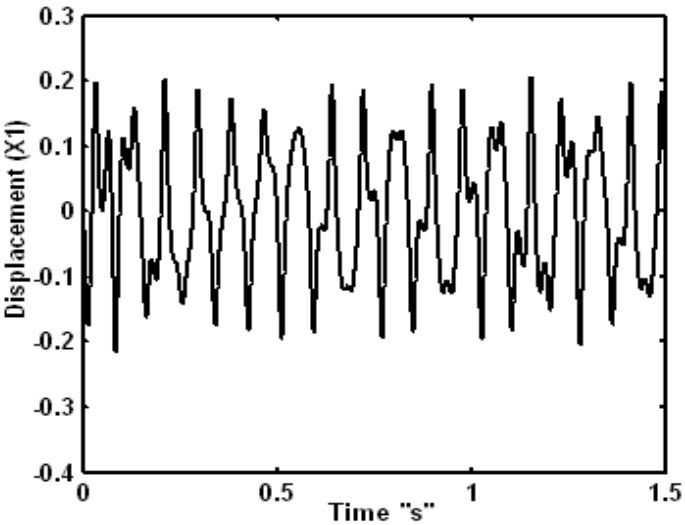


Fig. 8a. Nonlinear response (X1(t)) "Cubic damping and stiffness case"

The figure 8 a and b illustrate the response in time (a) and phase diagram (b) of the main system. One observes the strongly nonlinear behaviour in the temporal curve. We note also

the presence of two attractors more visibly that for the Duffing case and synonymous with a predominance of the stiffness non linearity. For the absorber, there is no notable modification compared to the Duffing case, in spite of the non validity of the superposition principle.

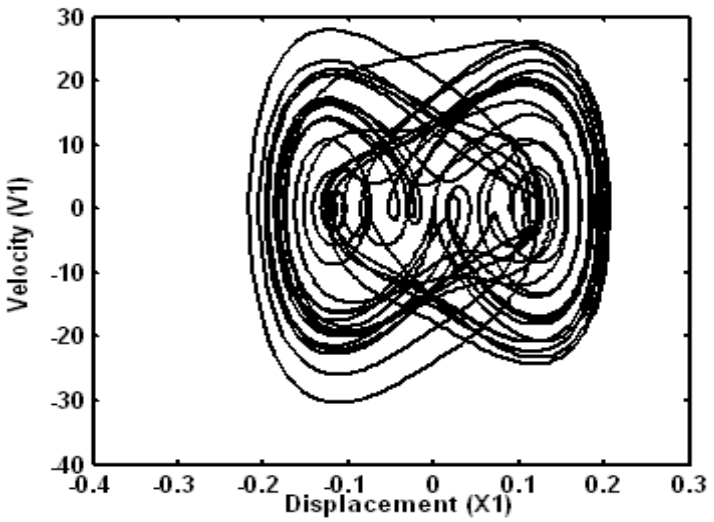


Fig. 8b. Phase diagram (X1,V1) "Cubic damping and stiffness case"

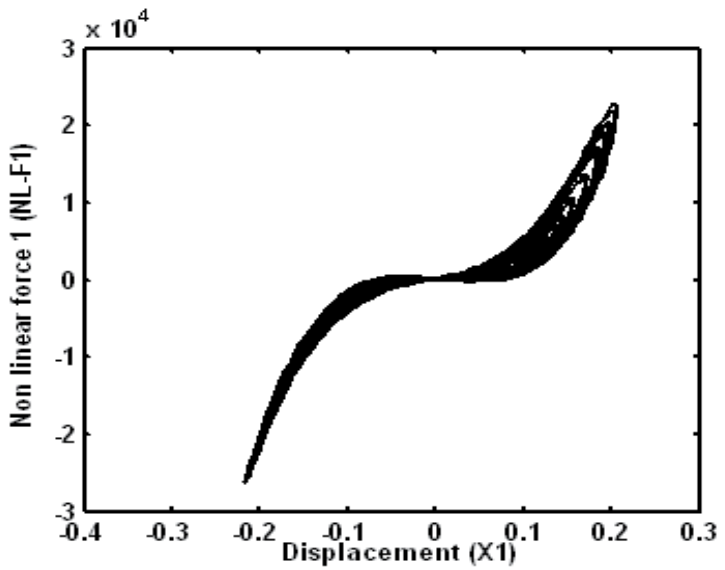


Fig. 9a. Nonlinear forces (X1, NL-F1) "Cubic damping and stiffness case"

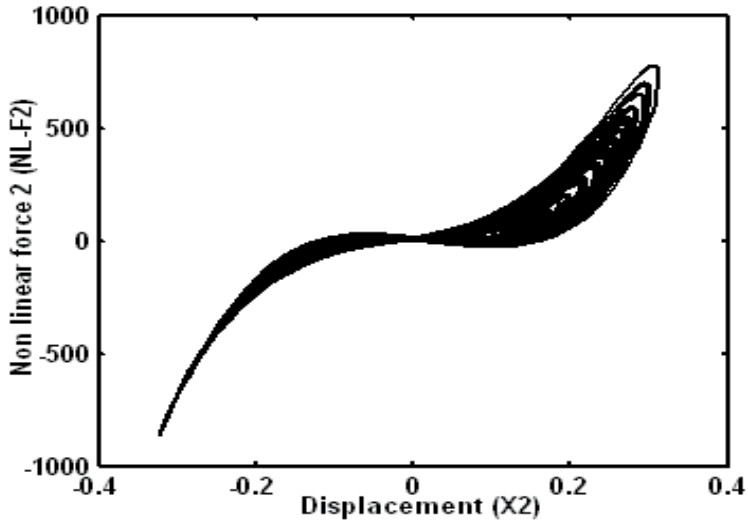


Fig. 9b. Nonlinear forces (X2, NL-F2) "Cubic damping and stiffness case"

The figure 9 a and b illustrate the nonlinear forces applied on each mass according to corresponding displacement. A low width cubic hysteretic cycle is observed for the main system but it's broader for the absorber, this may be caused by nonlinear damping. This width cycle form is dissymmetric between the two sides.

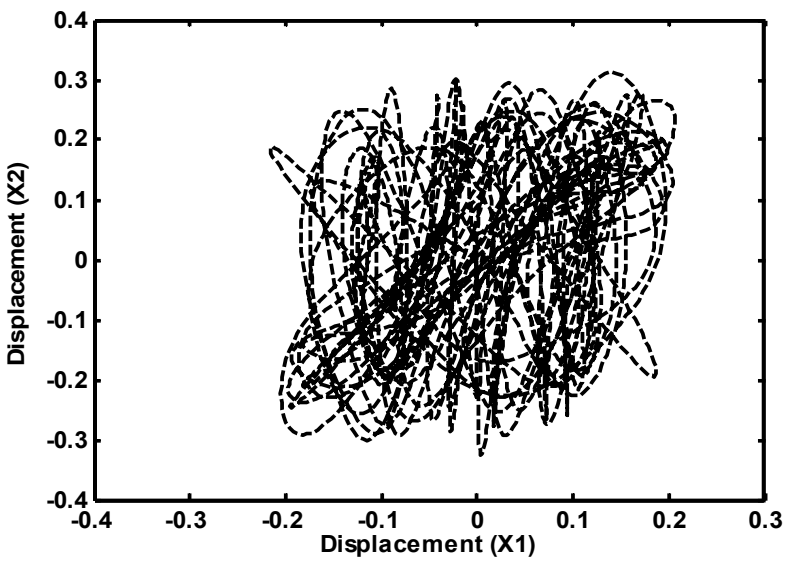


Fig. 10a. System trajectory in the plan (X1, X2) "Cubic damping and stiffness case"

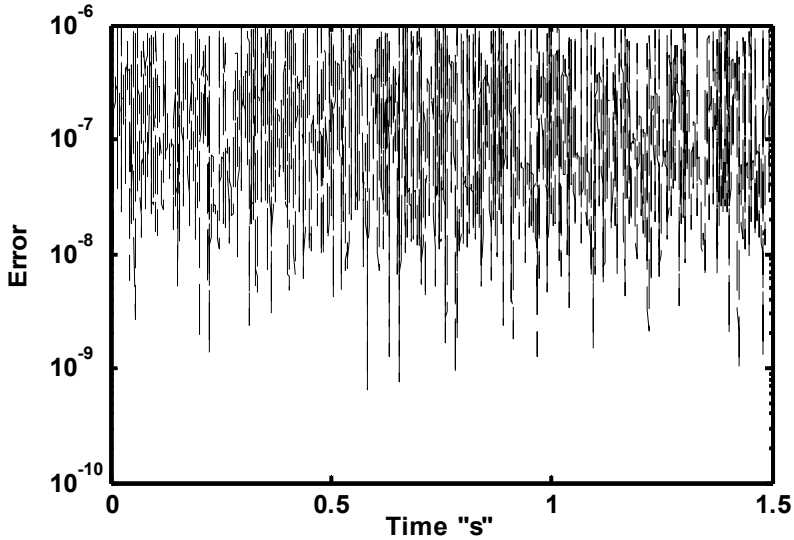


Fig. 10b. Error curve "Cubic damping and stiffness case"

It's also noted than the nonlinear forces are more affected by stiffness than by damping in form and amplitude. The narrow level stretch around the stable point disappears in this case. In the same way the nonlinear force of the main system is dominant. Figure 10a illustrates the complex trajectory of the global system in the plan (X1, X2), synonymous of instability but less than the Duffing case.

Fig. 10b shows the relative error curve. It is noted that the errors are dispersed between 10^{-9} and 10^{-6} during the iterative process, which doesn't prove a fast convergence for such high amounts.

We can conclude in this subsection that in the cubic damping and stiffness case, there are two attractors which appear clearly only when the amplitude of the excitation force increases. The strongly non linear damping may be the cause of this phenomenon.

The nonlinear hysteresis force curve is much more significant when the amplitude of the exciting force increases.

This hysteresis is not symmetric around the inflection point corresponding to the steady balance position.

Noticeable level stretch in the vicinity of this position, especially, for high loads is observed.

3.4 Nonlinear generalized damping and stiffness case: $r = 2$

In this section, the nonlinear generalized case (quadratic and cubic) in stiffness and damping is studied. The nonlinear mechanical system characteristics are: $\overline{k}_1^{'} = \overline{k}_2^{'} = \overline{c}_1^{'} = \overline{c}_2^{'} = 5$ and $\overline{k}_1^{''} = \overline{k}_2^{''} = \overline{c}_1^{''} = \overline{c}_2^{''} = 0.25$. It should be noted that the absolute physical characteristics in this case, have the same value ($\frac{k_1^{'}}{k_1} = \frac{k_1^{''}}{k_1} = \frac{c_1^{'}}{c_1} = \frac{c_1^{''}}{c_1} = 10^2$) in SI and also these parameters are selected among several tests, guaranteeing useable results on the one hand and of good convergence on the other hand.

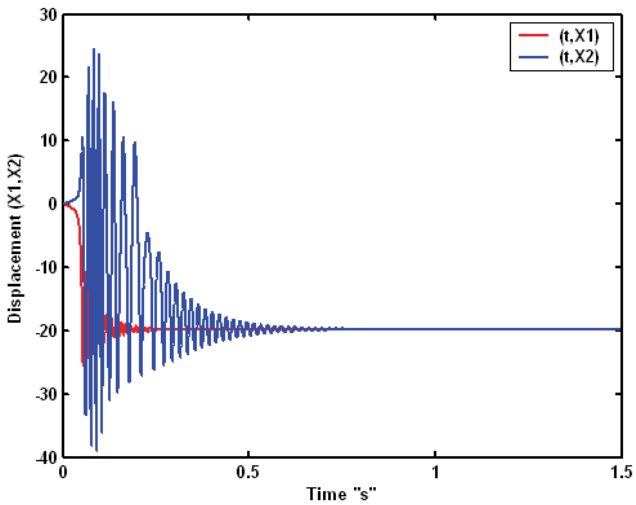


Fig. 11. Temporal nonlinear responses: (X1(t) and X2(t)) "Generalized case: quadratic and cubic for week damping"

Figure 11 shows the evolution of displacements of both dof according to time. It is noted that the absorber strangles completely the vibration in this proportioning configuration. If we increase the proportioning non linearity practically all the vibrations of the main system disappear. Moreover, as the global movement is very much diminished from t=0.6s. This is one of the effects of this generalized nonlinearity case. The displacement of the main system oscillates while remaining of negative sign, before converging quickly towards a constant value, the vibration is almost eliminated. However, absorber oscillates between the two signs before converging towards the same limit as that of the main system.

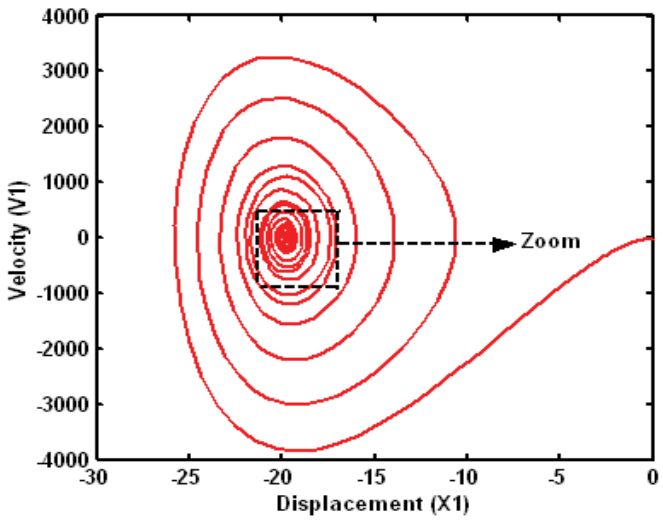


Fig. 12a. Phase diagram (X1,V1) "Generalized case: quadratic and cubic"

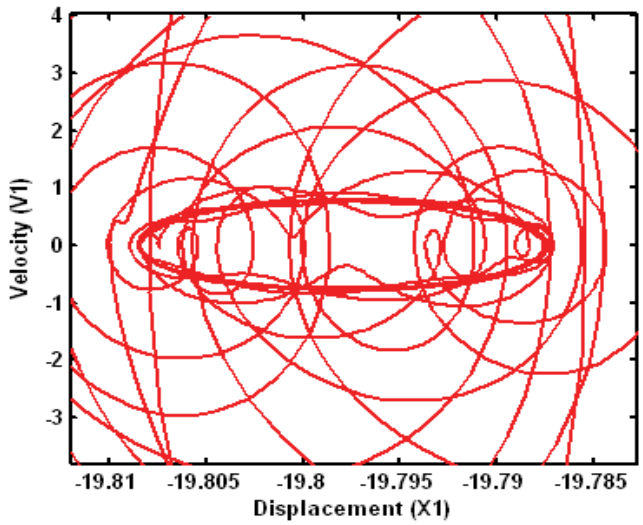


Fig. 12b. Zoom on the attractors "Generalized case: quadratic and cubic"

Figure 12a illustrates the phase diagram of the main system and shows the existence of two very close attractors, clarified by zoom (figure 12b). The latter indicates an elliptic orbit whose focuses coincide with the attractors.

Figure 13a illustrates the phase diagram of the absorber and also shows the existence of two very close attractors, clarified by zoom (figure 13b). Furthermore, an elliptic orbit is observed whose focuses coincide with the attractors.

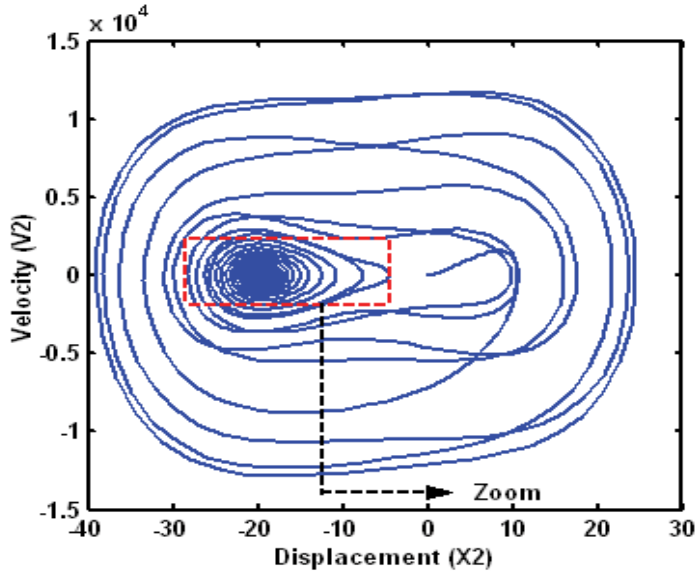


Fig. 13a. Phase diagram (X2,V2) "Generalized case: quadratic and cubic for week damping"

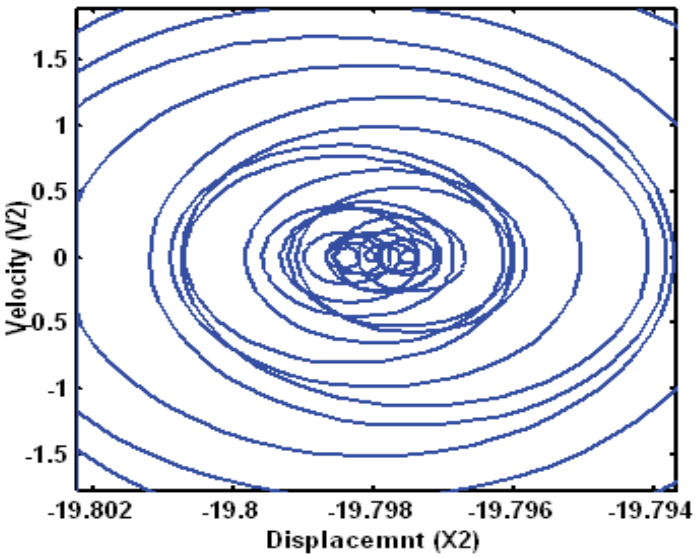


Fig. 13b. Zoom on the attractors "Generalized case: quadratic and cubic for week damping"

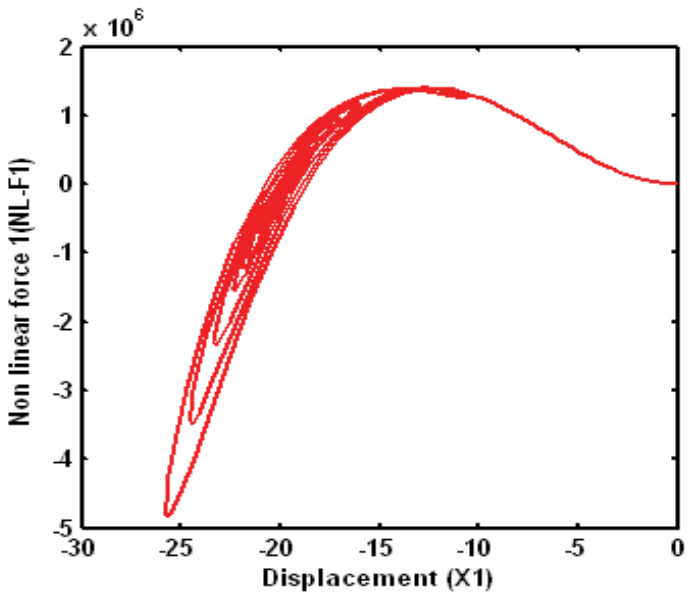


Fig. 14a. Nonlinear forces (X1, NL-F1) "Generalized case: quadratic and cubic for week damping"

The figure 14 a and b illustrate the applied nonlinear forces on each mass according to corresponding displacement. It's noted for the main system has a dominant parabolic form corresponding to a half hysteresis cycle at critical point with null force.

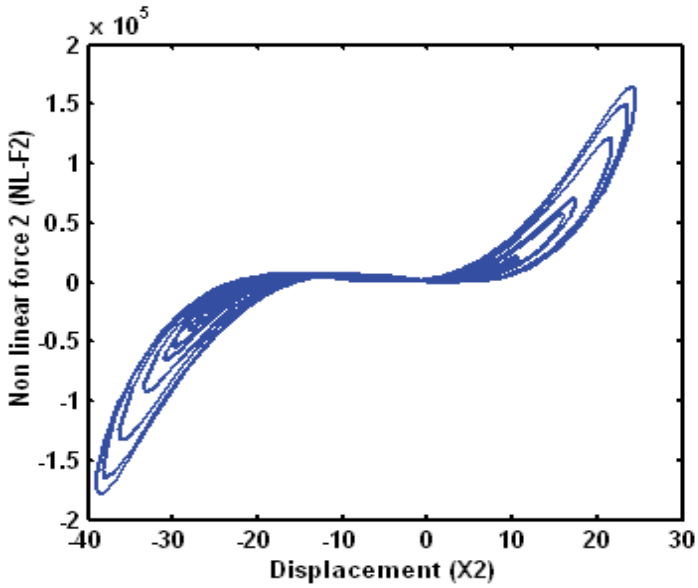


Fig. 14b. Nonlinear forces (X2, NL-F2) "Generalized case: quadratic and cubic for week damping"

This is more visible in figure 14a. But, a cubic hysteresis cycle with average width appears for the absorber, caused by nonlinear damping.
In the same way the amplitude of the nonlinear force of the main system is dominant. Also, the hysteresis opening is more significant for the absorber.

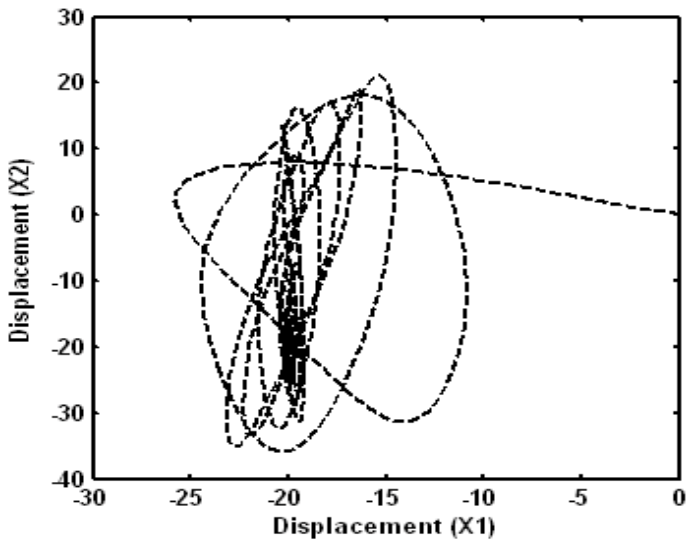


Fig. 15a. System trajectory in the plan (X1, X2) "Generalized case: quadratic and cubic for week damping"

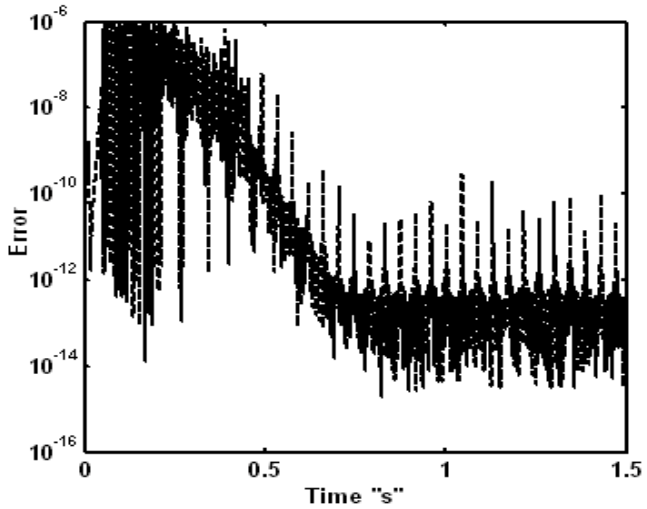


Fig. 15b. Error curve "Generalized case: quadratic and cubic for week damping"

Figure 15a illustrates the global system complex trajectory in the plan (X_1, X_2). It is quickly convergent to a critical point which differs notably from the steady balance position.
Figure 15b shows the curve of relative error. It is noted that the errors are dispersed, with a difficult convergence at the beginning but it becomes very fast after a certain time close to t=0.6s.

- ***Effect of the proportioning coefficients in this generalized case*** ($r = 2$)

The nonlinear mechanical characteristics of the system are: $\overline{k}_1^{'} = \overline{k}_2^{'} = 5$; $\overline{k}_1^{''} = \overline{k}_2^{''} = 0.25$ and $\overline{c}_1^{'} = \overline{c}_2^{'} = 500$; $\overline{c}_1^{''} = \overline{c}_2^{''} = 25$. In this case, it should be announced that we multiplied all the physical damping coefficients by 100 whereas we maintained those of stiffness.

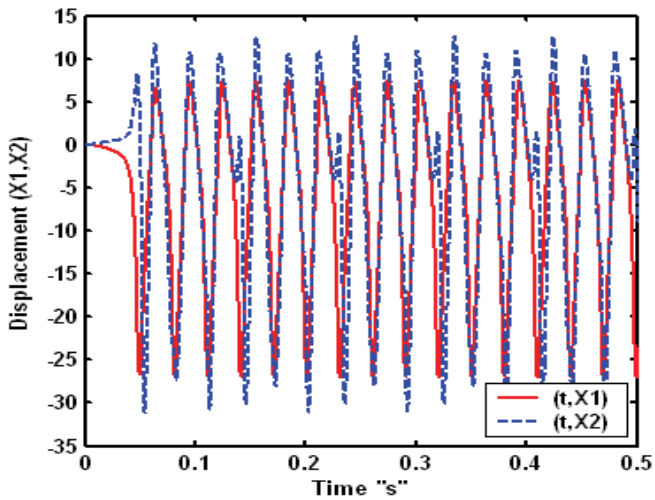


Fig. 16a. Nonlinear temporal responses (X1(t) and X2(t)) "Generalized case: quadratic and cubic for strongly damping"

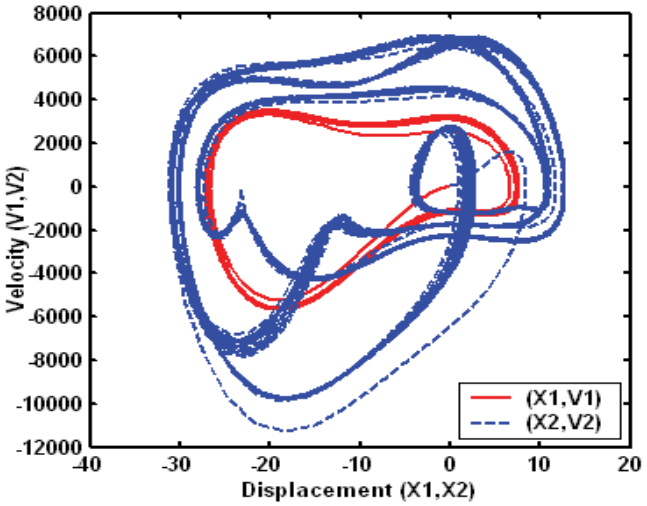


Fig. 16b. Phase diagrams "Generalized case: quadratic and cubic for strongly damping"

Figure 16a shows the displacements variation of both dof according to time. It is noted that the absorber is not the best adapted for this proportioning. The movement of the main system is similar then for linear behaviour, whereas that of the absorber takes a strongly nonlinear form.

Figure 16b illustrates the phase curves for the two dof. It is noted that when the damping proportioning of the absorber increases the main system has an elliptic limit cycle, whereas the absorber attractors persist converging with two limits cycle.

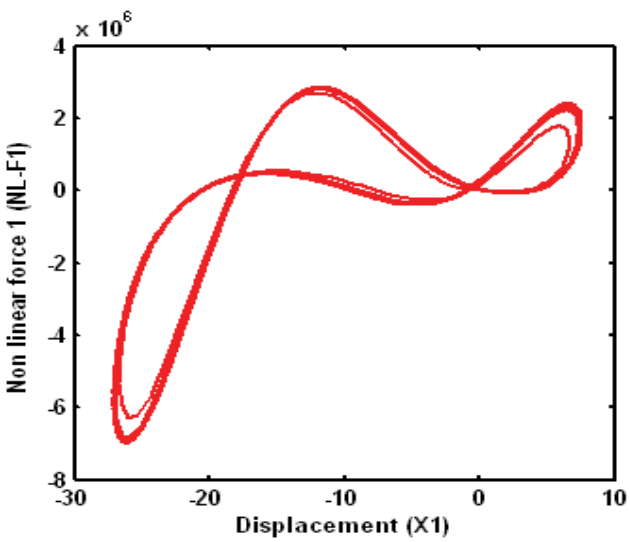


Fig. 17a. Nonlinear forces: (X1, NL-F1) "Generalized case: quadratic and cubic for strongly damping"

The figure 17 a and b show the variations of the nonlinear forces according to respective displacement.

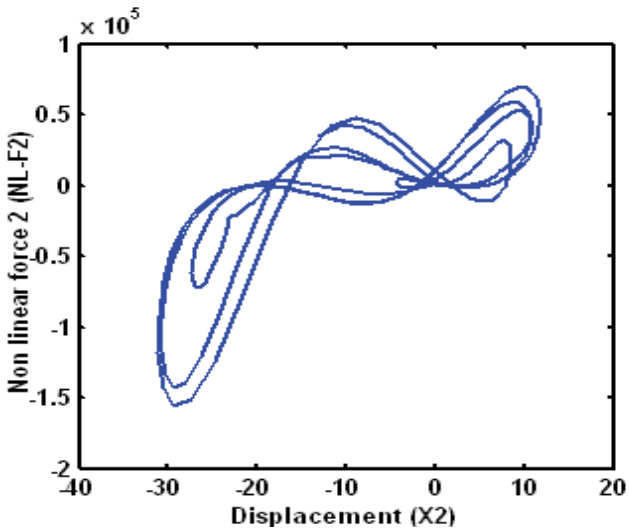


Fig. 17b. Nonlinear forces (X2, NL-F2) "Generalized case: quadratic and cubic for strongly damping"

It is noted that the increase in the damping proportioning creates a third hysteretic cycle of parabolic form for small displacements at the two critical points. One of these two points corresponds to the steady balance and the other corresponds to the extreme position of the absorber.

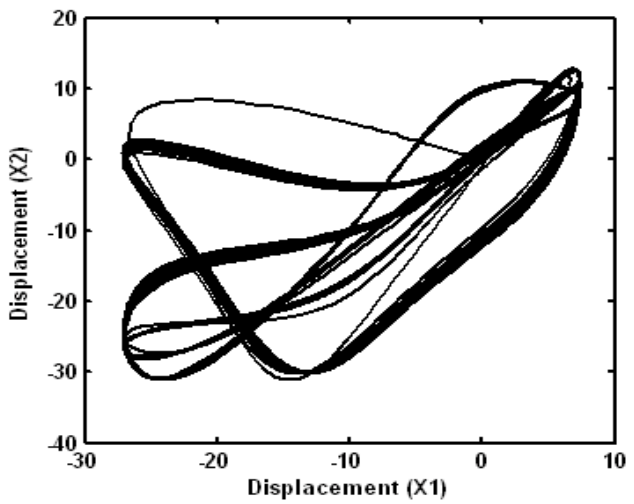


Fig. 18a. System trajectory in the plan (X1, X2) "Generalized case: quadratic and cubic for strongly damping"

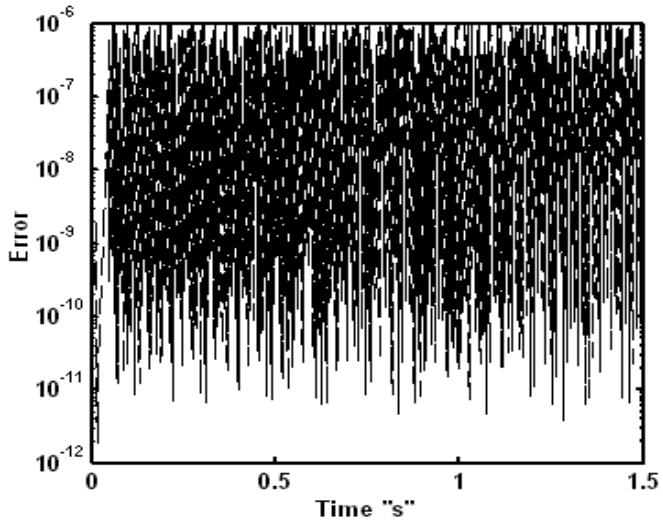


Fig. 18b. Error curve "Generalized case: quadratic and cubic for strongly damping"

Figure 18a illustrates the complex trajectory of the global system in the plan (X1, X2) but a stability is apparent with two complicate limit cycle orbits.

Figure 18b shows the curve of relative error. It is noted that the errors are dispersed between 10^{-12} and 10^{-6} during the iterative process, the convergence is better than previously and it does not change in time. Damping acts in the direction of vibration stabilization.

For this subsection, we can conclude that for this generalized case of non linearity the behaviour of the system is clearly different of the previously cases (subsections: 3.1, 3.2 and 3.3).

- The vibration of the main and the auxiliary system disappear relatively fast;
- The attractors are visible only with zooming;
- The hysteretic cycle of forces is original with a parabolic tendency and the apparition of one, two and three spindles.

These effects are strongly influenced by the proportioning of nonlinear parameters.

3.5 Nonlinear generalized damping and stiffness case: $r = 1.5$

In this section, the nonlinear generalized case (power 1.5 and cubic) in stiffness and damping is studied. The nonlinear mechanical system characteristics are: $\overline{k}_1^{'} = \overline{k}_2^{'} = \overline{c}_1^{'} = \overline{c}_2^{'} = 22.36$ and $\overline{k}_1^{''} = \overline{k}_2^{''} = \overline{c}_1^{''} = \overline{c}_2^{''} = 0.25$. It should be noted that the absolute physical characteristics in this case, have the same value ($\frac{k_1^{'}}{k_1} = \frac{k_1^{''}}{k_1} = \frac{c_1^{'}}{c_1} = \frac{c_1^{''}}{c_1} = 10^2$) in SI. These proportioning are decreased to make more quick the convergence and more significant physical results.

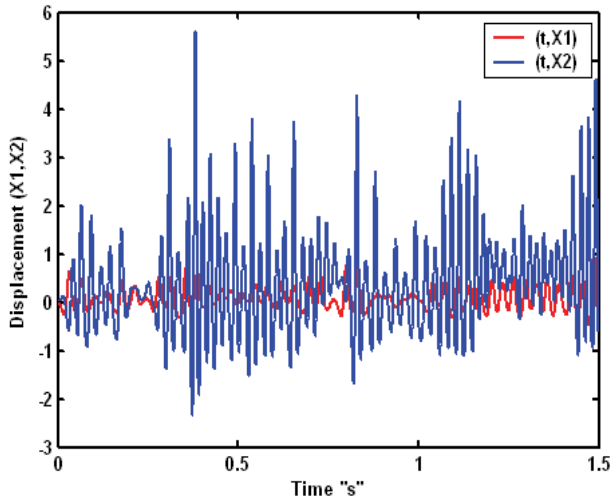


Fig. 19. Temporal nonlinear responses: (X1(t) and X(t)) "Generalized case: power 1.5 and cubic for week damping"

Figure 19 shows the displacements evolution for the two dof according to time. It is noted that the auxiliary system completely absorbs the vibration in this proportioning configuration. It is as to note as the movement is relatively irregular similar of specified non linearity. That may be one of the effects of the generalized non linearity for the non integer power less then 2.

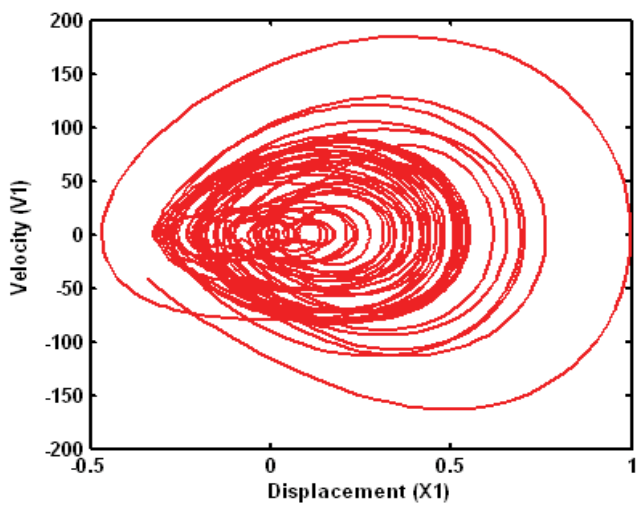


Fig. 20a. Phase diagrams (X1, V1) "Generalized case: power 1.5 and cubic for week damping"

The figure 20 a and b illustrate the phase curves of the main system and the absorber. They show the existence of two attractors very brought closer with an oval cycle orbit.

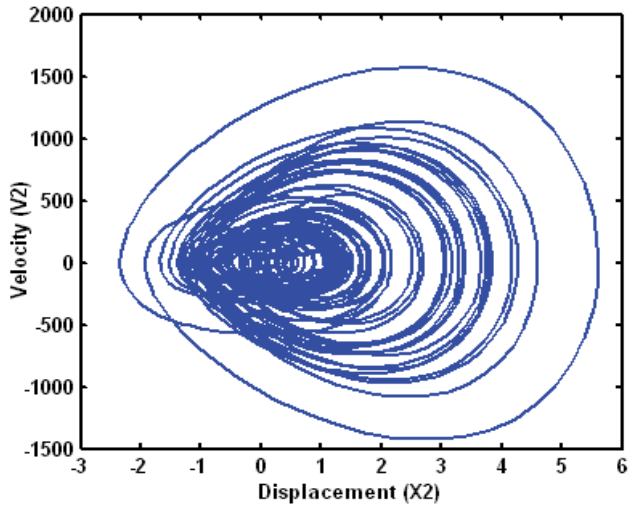


Fig. 20b. Phase diagrams (X2, V2) "Generalized case: power 1.5 and cubic for week damping"

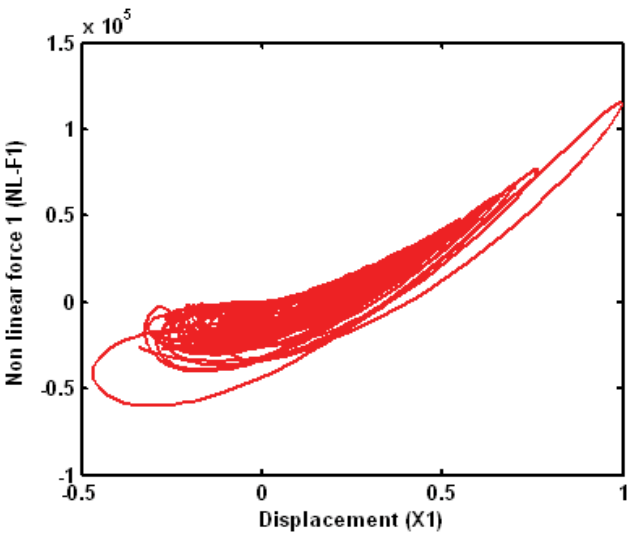


Fig. 21a. Nonlinear forces (X1, NL-F1) "Generalized case: power 1.5 and cubic for week damping"

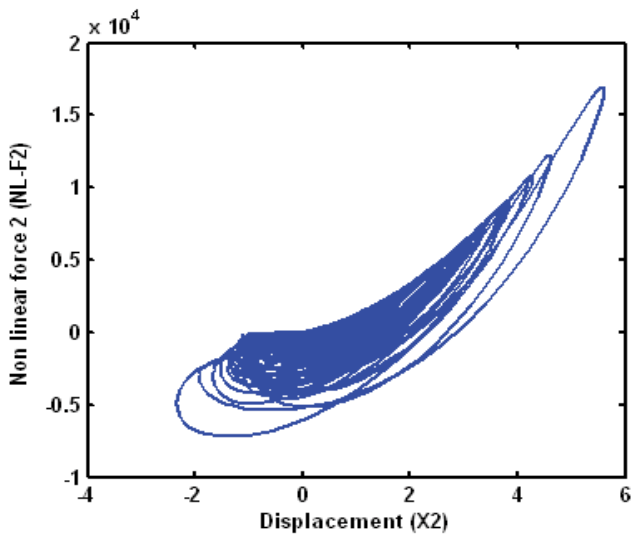


Fig. 21b. Nonlinear forces (X2, NL-F2) "Generalized case: power 1.5 and cubic for week damping"

The figure 21 a and b illustrate the nonlinear applied forces on each mass according to corresponding displacement. It is noted that for the two dof one has a dominant parabolic form with a fast convergence towards the steady balance with null forces. If one compares these results with the generalized quadratic case one notices a significant difference mainly for the absorber, which shows the originality of this case with a non integer power between 1 and 2.

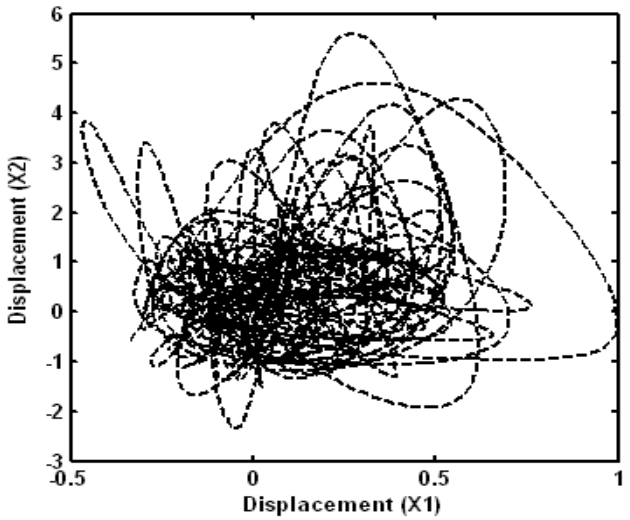


Fig. 22a. The trajectory of the system in the plan (X1, X2) "Generalized case: power 1.5 and cubic for week damping"

Figure 22a illustrates the global system complex trajectory in the plan (X_1, X_2). One notes a less regular trajectory with very difficult convergence as well as for the main system as for the

absorber. Figure 22b shows the curve of relative error. It is noted that the errors are dispersed between 10^{-9} and 10^{-6} during the iterative process, what proves an acceptable convergence.

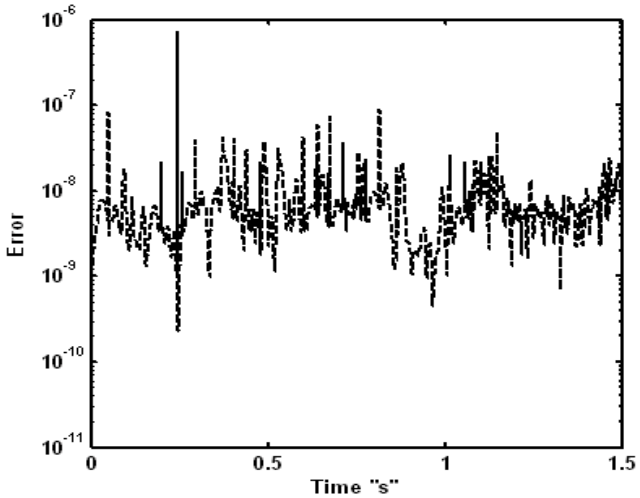


Fig. 22b. Error curve "Generalized case: power 1.5 and cubic for week damping"

- ***Effect of the proportioning coefficients in this generalized case*** ($r = 1.5$)

The nonlinear mechanical system characteristics are: $\overline{k}_1^{'} = \overline{k}_2^{'} = 22.36$; $\overline{k}_1^{''} = \overline{k}_2^{''} = 0.25$; and $\overline{c}_1^{'} = \overline{c}_2^{'} = 2236$; $\overline{c}_1^{''} = \overline{c}_2^{''} = 25$. In this case, we keep the stiffness constant and we multiplied those of damping by 100.

Figure 23 shows the variation of displacements of the two dof according to time.It is noted that the absorber is not the best adapted for this proportioning.

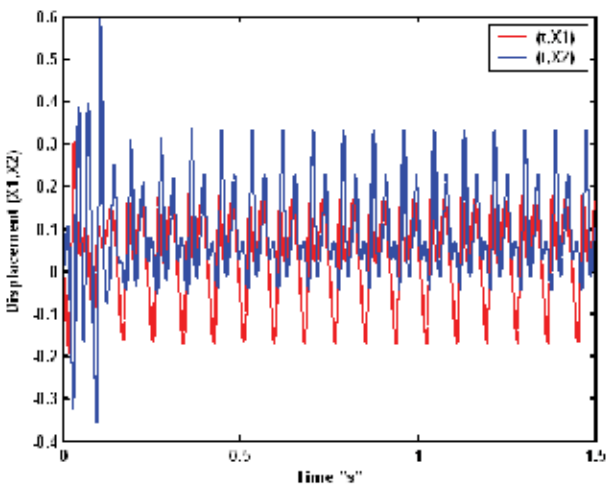


Fig. 23. Temporal nonlinear responses: (X1(t) and X2(t)) "Generalized case: power 1.5 and cubic for strongly damping"

It is also notable that the main system movement is similar to that of the absorber in amplitude but with a great dephasing. The form of these curves is strongly nonlinear but, after the transitory phase, the movement is periodic and it is stabilized by damping. The period of these motions is similar to that of excitation but with a high non linearity.
The figure 24 a and b illustrate the phase curves of the main system and the absorber. They especially show the existence of two limits cycle for the absorber. For the main system we observe only one limit cycle.

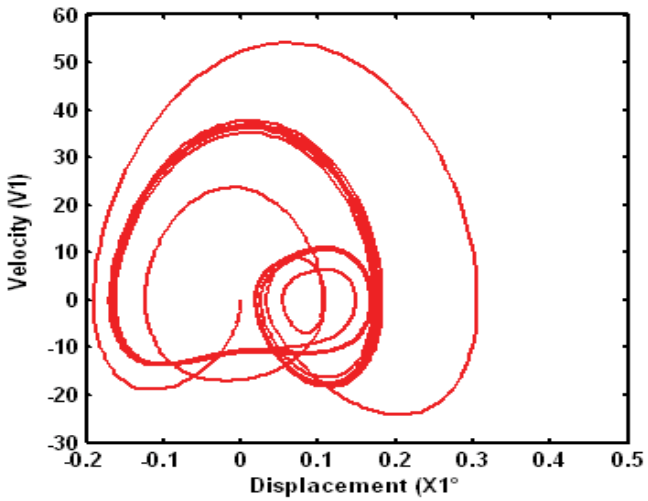


Fig. 24a. Phase diagrams (X1, V1) "Generalized case: power 1.5 and cubic for strongly damping"

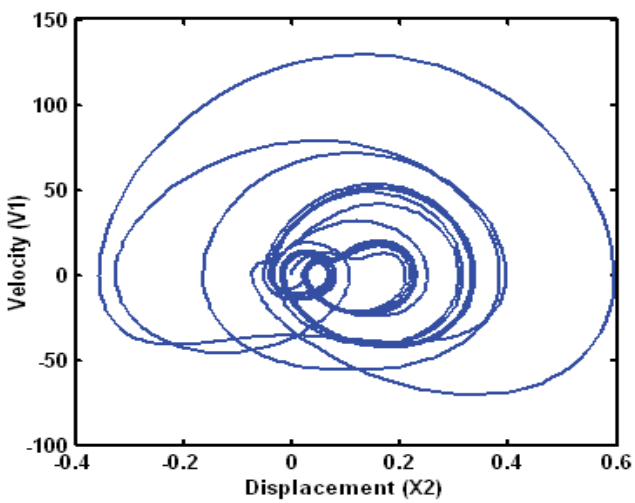


Fig. 24b. Phase diagrams (X2, V2) "Generalized case: power 1.5 and cubic for strongly damping"

The figure 25 a and b illustrate the nonlinear forces on each mass according to corresponding displacement. It is noted that for the two dof we observe more opened hysteretic form than the previous case with week damping.
Figure 26-a illustrates the complex trajectory of the global system in the plan (X1, X2). One notes a less regular trajectory with difficult convergence notably for the absorber.

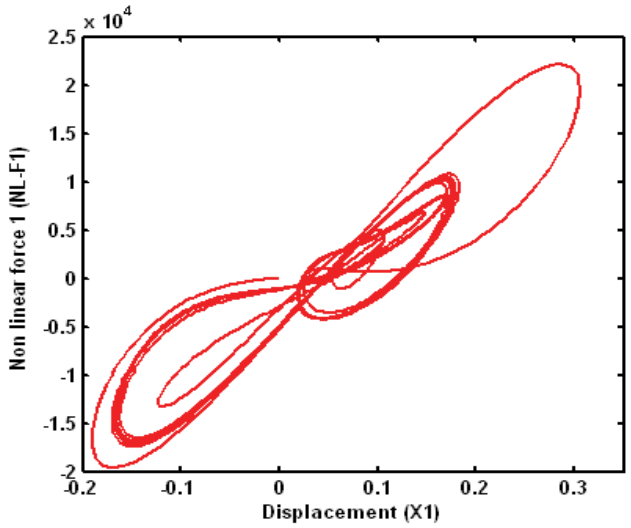


Fig. 25a. Nonlinear forces (X1, NL-F1) "Generalized case: power 1.5 and cubic for strongly damping"

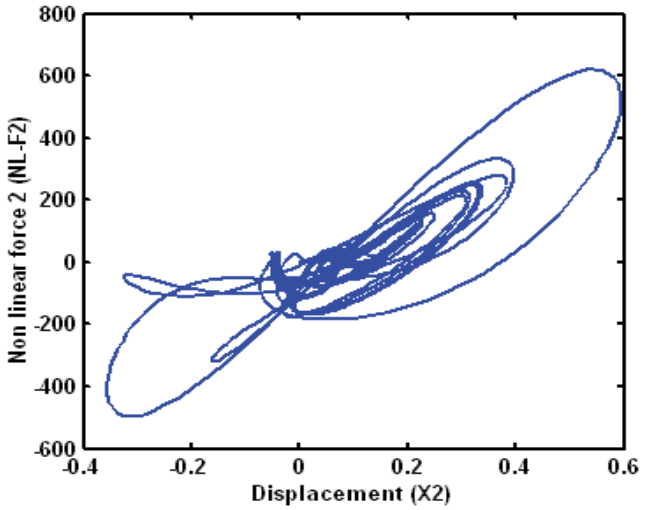


Fig. 25b. Nonlinear forces (X2, NL-F2) "Generalized case: power 1.5 and cubic for strongly damping"

Figure 26b shows the curve of relative error. The errors are dispersed between 10^{-8} and 10^{-6} during the iterative process, which proves difficulty of the convergence of this model compared to the case with a weak nonlinear damping. We can conclude:

- Increase the proportioning of the non linear damping of the system stabilizes the system vibration;
- The attractors become less visible with damping and the hysteretic cycle well be more symmetric around the steady balance position;
- The convergence is more difficult when the non linear damping increases.

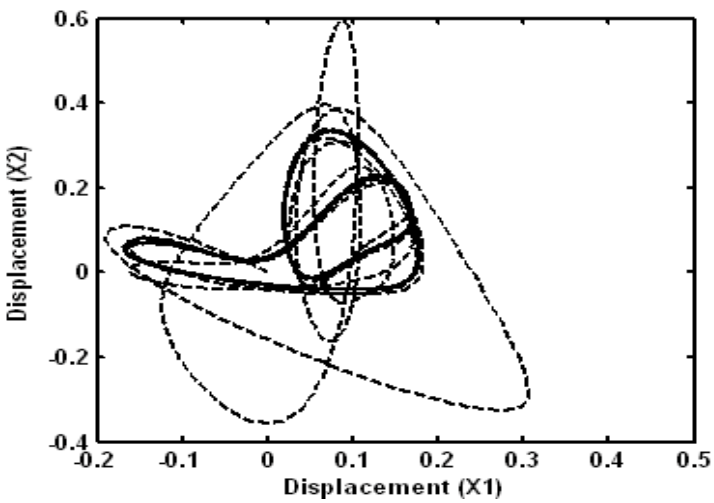


Fig. 26a. The trajectory of the system in the plan (X1, X2) "Generalized case: power 1.5 and cubic for strongly damping"

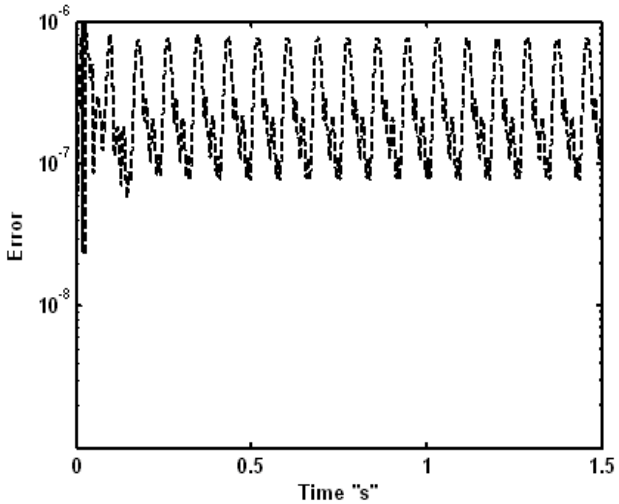


Fig. 26b. Error curve "Generalized case: power 1.5 and cubic for strongly damping"

Model	Generalized case : r=1.5		Generalized case : r=2		Generalized case : r=3	Van Der Pol case	Duffing case
Type	Strongly stiffness and damping	Strongly stiffness and Week damping	Strongly stiffness and damping	Strongly stiffness and Week damping	Strongly stiffness and damping	Strongly damping	Strongly stiffness
Displacement	- Periodic - Non linear	-Strongly irregular	-Non linear - Regular and periodic	- Quickly converge - Decreasing motion	Very irregular	-Harmonic - the 2 dof in opposition phase	Periodic with 2 periods
Phase	- Complex limits of cycle	- Oval hidden 2 attractors	-2 limit cycles	2 attractors with zoom	2 visible attractors	- Limits of cycles	-2 attractors -Lim it cycle
Forces	- Open cubic cycle - Quickly convergent towards the steady balance	-Parabolic cycle	- 3 lobes of Hysteretic cycles-	Half of cycle with Critical point	-Cubic cycle -More open in one side	-Cycle with 2 spindles -Broad and symmetric	Cubic hysteretic with level stretch
Trajectory	- Irregular - Quickly convergent	- very Irregular	- Stable complex	Elliptic	Complex	Elliptic	Complex
Error	- High iteration count - Difficult convergence	Week	-Low -Instable	Great then low after transition time	Intermediary	Great	Intermediary
Observations	Hysteretic cycle changing tendency with displacement	-Good convergence - Cycle with week power and convergence point	Quadratic dominance for low displacement	Hysteretic cycle with Critical point	Dissymmetrical hysteretic cycle	Bad convergence	Good convergence

Table 1. Recapitulation

General comments:

To recapitulate one can include commenting the two first cases treated, witches are often mentioned in the literature.

Concerning the cubic damping and stiffness case (r=3), we note that for this model the non linearity of stiffness is rather dominant in spite of the strong amounts chosen for damping.

When one goes up in nonlinearity power beyond the power three, few changes appear on the nonlinear behaviour of the system for a week damping and a cycle with three spindles when the damping increases. For the last case the trajectory becomes more complicated.

When one interferes a non integer power nonlinearity (here power r=1.5) with cubic non linearity the convergence of computations becomes more difficult and it becomes even impossible for a rational between 2 and 3.

Indeed the attractors become closer to each other and only a zoom can separate them.

The appearance of half hysteretic cycle of nonlinear force at critical point is typical of this model.

Table 1 recapitulates these results.

4. Robust multi-objective optimization

The problems of optimization in the field of the machinery of the structures are often multi-objective; the latter can be in conflict. For that, it is necessary to choose a strategy of multi-objective optimization able to propose the best alternatives among several. A multi-objective optimization requires two steps witch is:

- Determination of the objective functions to be optimized

Obtaining these objective (or cost) functions are carried out directly during optimization by sampling.

- Choice of the technique of search for the optimal solutions

To search the optimal solutions, the genetic algorithm of NSGA type is used. Indeed, this algorithm makes it possible to better explore the design space and to exploit the whole of the Pareto front.

Generally, a problem of multi objective optimization is expressed by the equation below:

$$\min F(x)=\left(f_1(x), f_2(x),\ldots\ldots, f_n(x)\right)^T ; x \in S$$

where: $f_1(x), f_2(x),\ldots\ldots, f_n(x)$ are the cost functions and $x=\left(x_1, x_2,\ldots\ldots, x_n\right)^T$ is the vector of n parameters of optimization, $S \in R^n$ represents the whole of the realizable solutions and $F(x)$ is the vector of the functions to be optimized.

The whole of the optimal solutions is that formed by the solutions which are not dominated by others. The physical parameters used to describe a structure are often random. These parameters are generally identified like random variables and are introduced into an approach of resolution of the problems such as optimization. In the field of the optimization of the structures, the taking into account of the robustness of the solutions is essential in the research of the optimal design. In fact it is well-known that a theoretically excellent solution can prove to be catastrophic in practice if the errors made during manufacture for example do not make it possible to obtain the values of the variables of design with a sufficient

precision. A sub-optimal but stable solution with respect to the tolerances of manufacture can prove much more interesting for the originator.

Generally the construction of a robustness function is based on the mean value and the standard deviation. Indeed, many authors proposed to define the robustness function f^r of an original objective function f as being the relationship between the mean value and the standard deviation: $f^r = \left(\sigma_f / \mu_f\right)^{-1}$.

The ratio σ_f / μ_f indicates dispersion (or the vulnerability function $f^v(x)$) of f, where μ_f and σ_f are respectively the mean value and the standard deviation calculated on the whole of the samples $(f_i)_{1 \le i \le N}$ of a function $f(x)$. N is the number of simulations of Monte Carlo.

The robust multi-objective optimization problem is thus built by simultaneously optimizing the original cost functions and their robustness: $\begin{cases} M \text{in } F(x) = \left(f_1(x), f_1^r(x), ..., f_m(x), f_m^r(x)\right)^T \\ avec\ x \in S \end{cases}$ $f_i^r(x)$ is the robustness of the objective function $f_i(x)$ $i = 1, ..., m$.

The robust solutions with respect to uncertainties are those which simultaneously make it possible to minimize the original cost functions. $\left(f_1(x), f_2(x), ..., f_m(x)\right)$ and to maximize their robustness $\left(f_1^r(x), f_2^r(x), ..., f_m^r(x)\right)^T$

In this case we can study the cubic *nonlinear stiffness*.

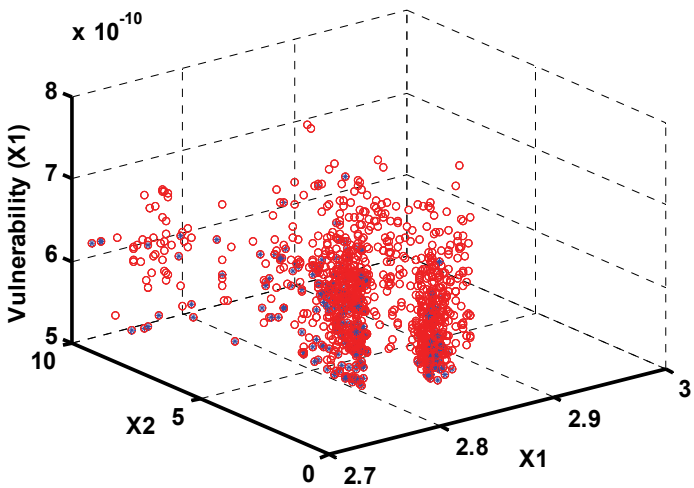


Fig. 27. r=3: Cubic nonlinear damping and stiffness – periodic motions: Overall solutions in space (X1, X2, Vulnerability (○) and robust solutions (*);.

Figure 27 illustrates the whole of the solutions obtained by simulation under the constraints given and in the band of sampling. Among these solutions one finds those which are robust (blue).

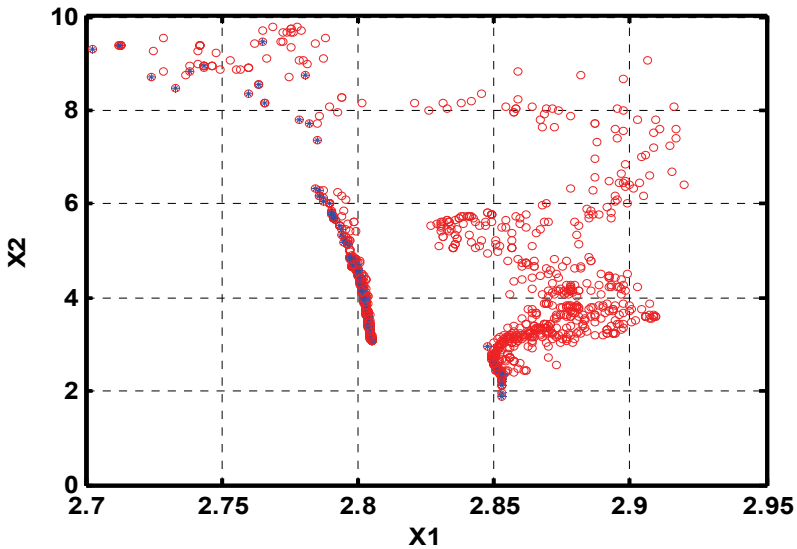


Fig. 28. Overall solutions in the plan (X1, X2) and robust solutions

Figure 28 presents the nonlinear displacement of the absorber according to the principal structure vulnerability. It is noted, that the designer has a flexibility to choose the absorber which is better appropriate with guarantee of robustness (minimal dispersion).
Figure 29 carries only the whole of the robust solutions in the plan X1 and X2. Figure 30 shows that the robust solution "A" which favours the minimal displacement of the absorber. The absorber maximum vibration is reduced to approximately 200%. This proves the importance of the proposed robust multi-objective optimization strategy in the periodic motions.

5. Conclusion

In this chapter, we proposed a mechanical system modeled by only one dof relating to its first mode and equipped with a dynamic absorber with only one dof too.
Thus the system obtained comprises a cubic and quadratic generalized nonlinearity combined stiffness and of damping. We used the numerical diagram of Newmark with linear acceleration in order to find the temporal response of the principal system with absorber subjected to a harmonic load.
Determinist calculations made it possible to highlight the contribution of this type of nonlinearity on the absorption of the vibrations and the behavior of the nonlinear system through the curves of phases (attractive, ovalization, etc).
The bifurcation diagram of the absorber in the cubic case for the chosen parameters illustrates three kinds of motions in several frequency bands: periodic, quasiperiod and chaotic motions.

In the optimization part, one determined the optimal characteristics of the damper located on the first face of Pareto with the compromise optimality-robustness. These solutions contribute to the robust design of the nonlinear system.

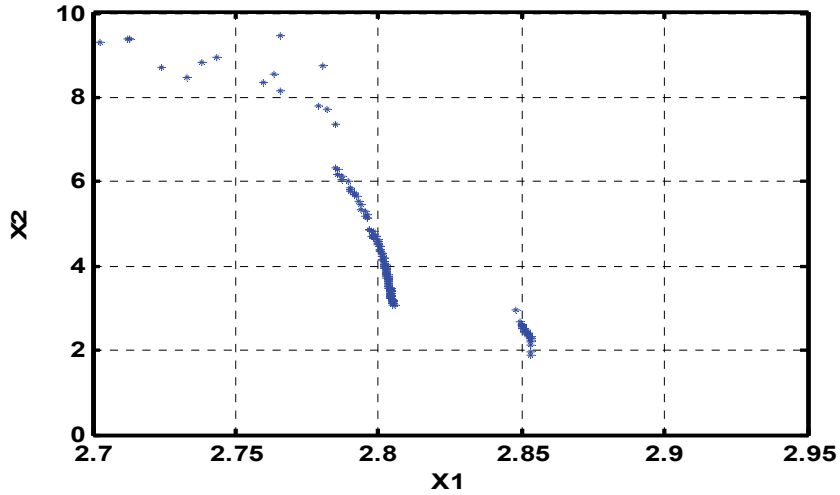


Fig. 29. Robust solutions in the plan (X1, X2)

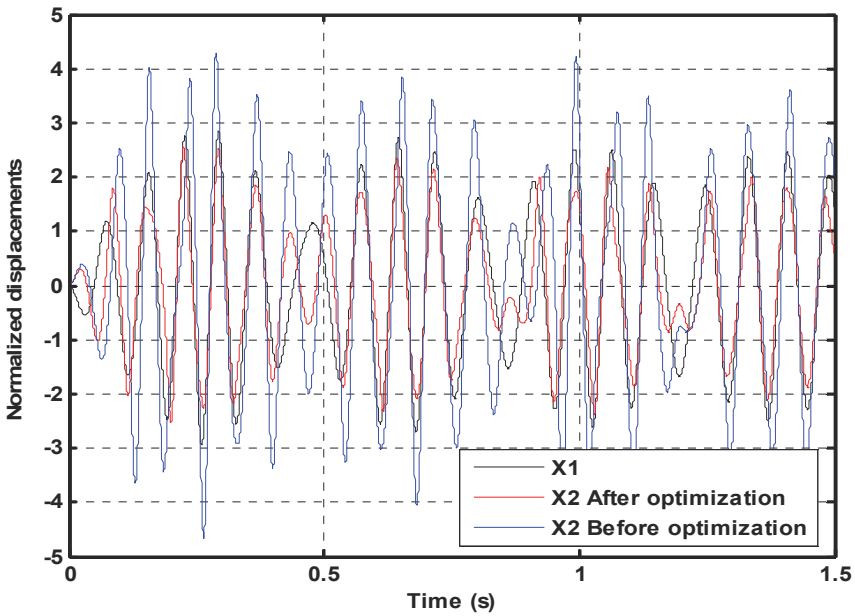


Fig. 30. Non-linear displacements before and after optimisation: Case of solution A

6. References

[1] G.S. Whiston, The vibro-impact response of a harmonically excited and preloaded one-dimensional linear oscillator, *Journal of Sound and Vibration* 115(2), pp. 303-319, 1987.

[2] S. Natsiavas, Steady state oscillations and stability of non linear dynamic vibration absorber, *Journal Sound and vibration* 156(2), pp. 227-245, 1992.

[3] S. Natsiavas, Stability of piecewise linear oscillators with viscous and dry friction damping, *Journal Sound and vibration*, Vol. 217, Issue 3, pp. 507-522, 1998.

[4] K.W. Chung, C. L. Chan, Z. Xu, J. Xu, A perturbation- incremental method for strongly non linear non autonomous oscillators, *International Journal of Non linear Mechanics*, 40, pp. 845-859, 2005.

[5] A.F. Vakakis and S.A Paipetis, the effect of viscously damped dynamic absorber on a linear multi-degree-of-freedom system, *Journal of sound and vibration*, vol. 105, Issue 1, pp 49-60 22 , 1986.

[6] S. Natsiavas, Dynamics of multiple-degree-of-freedom oscillators with colliding components, *journal of sound and vibration* 165 (3), 439-453 ,1993.

[7] S. Natsiavas and Tratskas P. On vibration isolation of mechanical systems with non-linear foundations *journal of sound and vibration*, volume 194, issue 2, 11 July 1996, p 173-185.

[8] G.Verros and S.Natsiavas, Forcing induced asymmetry on dynamical systems with cubic non linearity, *Journal of sound and vibration* volume 233, Issue 2, 1 June 2002, p 279-295.

[9] S.J.Zhu, Y.F Zheng and Y.M. Fu, Analysis of non-linear dynamics of a two-degree-of-freedom vibration system with non-linear damping and non linear spring, *Journal of sound vibration* 271, pp.3 15-24, 2004.

[10] B. Erkus, Comparison of the techniques used in the New mark analysis of non linear structures.*17th ASCE Engineering Mechanic conference June* 13-16, University of Delware Newmark de EM, 2004.

[11] D.Roy and R. kumar, A multi-step transversal linearization (MTL) method in non-linear structural dynamics, *journal of sound and vibration* 287, pp. 203-226, 2005.

[12] A.F.EL-Bassiouny, Effet of non linearities in elastomeric material dampers on torsional oscillation control, *Applied Mathematics and Computation* 162 (2005) 835-854.

[13] D. Schaffer, Multiple Objective Optimisation with Vector Evaluated Genetic Algorithm, In genetic algorithm and their applications, *Proceedings of the First International Conference on Genetic Algorithm*, pp. 93-100, 1985.

[14] E. Zitzler and L. Thiele, Multiobjective Evolutionary Algorithms: A comparative case study and the strength Pareto approach, *IEEE Transaction on evolutionary computation*, vol. 3, pp. 257-271, 1999.

[15] N. Srivinas, K. Deb, Multiobjective Optimisation using Non dominated Sorting in Genetic Algorithms, *Technical Report*, Department of Mechanical Engineering, Institute of Technology, India, 1993.

[16] M.- L. Bouazizi, S. Ghanmi, R. Nasri, N. Bouhaddi, Robust optimization of the non-linear behaviour of a vibrating system, *European Journal of Mechanics*-A/Solids, volume 28 (2009), pp: 141- 154.

[17] S. Ghanmi, M.-L. Bouazizi, N. Bouhaddi, Robustness of mechanical systems against uncertainties, 24rd, *Finite Elements in Analysis and Design* 43(2007), pp. 715 – 731.

[18]B. Ait Brik , S. Ghanmi, N. Bouhaddi, S. Cogan, Robust Multiobjective optimisation Using Response Surfaces, 23 rd, *International Modal Analysis Conference* (IMAC-IVX), orlando, USA,Inc., 2005.

26

Harmonic Potential Fields: An Effective Tool for Generating a Self-organizing Behavior

Ahmad A. Masoud
King Fahd University of Petroleum & Minerals
Saudi Arabia

1. Introduction

This work provide a proof-by-example of the ability of harmonic potential fields (HPF) to exhibit a self-organizing behavior that can be utilized in building decentralized, evolutionary, multi-agent systems. It is shown that the strong relation the single agent HPF approach has to the evolutionary artificial life (AL) approach may be utilized at the multi-agent level to synthesize decentralized controllers that can be applied to a large variety of practical problems. We first provide a background of the single agent HPF approach along with its relation to the AL approach. Different multi-agent, HPF-based methods are presented along with simulation examples to demonstrate the utility of these techniques.

Humans have long attempted to bridge the gap between actions under their direct command (control variables) and directly inaccessible desired aspects of the environment they want to influence. This is carried-out by constructing a chain of causality linking the two together; hence making those directly inaccessible aspects indirectly accessible to the human operator. The process that realizes this chain of causality is called a servo-process. There are more than on type of problems that a servo-process have to rectify in order to enable causality to flow from the control side to the desired outcome side. The failure could be caused by insufficient quantity of effort that is being exerted at the control variable side. It may be the result of incompatibility of the control effort with the aspects of the environment that is to be influenced. The lack of organization in terms of the proper spatial-temporal distribution of the assets comprising the servo-process is a serious and difficult to detect source of failure. The sufficiency of the level of information available to constructor of the servo-process is also a fundamental cause of failure. Attention in this chapter is paid to the third type of failure concerning the faulty organization of the servo-process resources.

Any servo-process must, among other things, regulate the interaction among its sensory, processing, communication and actuation components. There are a number of distinct modalities in which these components are governed each suited to tackle a certain situation. Each one of these modalities gives rise to a family of planners. A planner is an intelligent, goal-oriented, context-sensitive controller that instructs the servo-process on how to deploy its actuators of motion so that a target situation may be reached in a constrained manner. Probably the most common modality used by a servo-process is the: know-plan-Act modality which is commonly called the: model-based approach (figure 1). Here, the servo-process uses its sensors to collect data about the situation it has to deal with. This data is converted into a representation. The representation is processed to generate a plan or sub-

tasks that have to be sequentially executed. These tasks are in turn fed to an actuation stage. Despite the popularity of this approach, it was found to suffer from problems such as: slow reaction to changes in the environment, aging of information and outdated plans and issues in converting the generated plan into successful actions.

Fig. 1. The model-based approach

To solve the problems of slow reaction to changes in the environment and outdated plans, the sense-act modality was suggested (figure 2). This reactive modality proved to be highly practical; however, its applicability is limited to simple tasks.

Fig. 2. The reactive approach

With limited success, attempts to improve the robustness of the model-based approach were carried-out by hybridizing it with the reactive approach (figure 3).

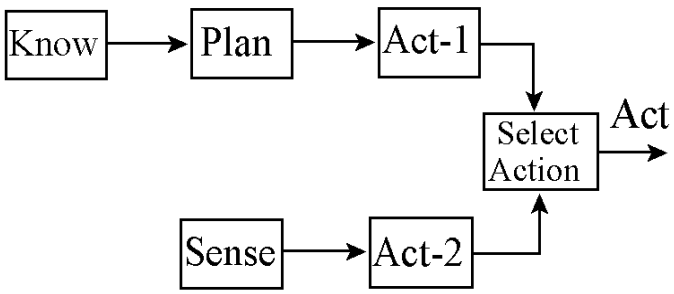


Fig. 3. The hybrid reactive – Model-based approach

Hardwiring sensors to an algorithm that directly feeds an actuation stage was found to be quite an effective modality for behavior generation (figure 4). To best of this author's knowledge, planning techniques based on this modality are difficult to adapt in situational spaces that have dimensionality more than two.

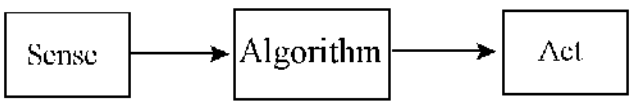


Fig. 4. The algorithmic, sensor-based approach

The subsumbtion architecture (figure 5) proved to be an effective modality for building servo-processes that have high chance of success operating in a realistic environment. This modality relies on direct sensory feed from the environment to a group of nested behaviors which the servo-process can project. These hierarchical behaviors can override each other

when the situation makes necessary to do so. The behavior of the resulting servo-process is situated, embodied, intelligent and emergent.

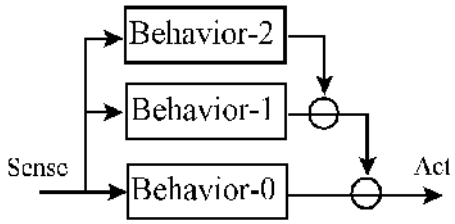


Fig. 5. The subsumbtion architecture

A belief-based approach (figure 6) is a powerful modality that allows a servo-process to act without the need for a common globally-agreed-on representation of the environment. Instead, the process forms its personal representation by continuously shaping its belief based on its experience of the environment.

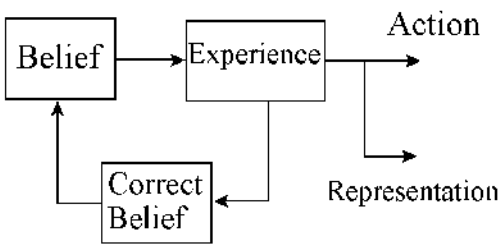


Fig. 6. Belief-based structures

Servo-processes may be built for any task and in any environment physical and nonphysical. A modality that suits the nature of the task is crucial for success. Modern technology has made great advances in miniaturizing and replicating devices and processes. This has strongly brought forward the possibility of building a distributed systems that are actualized by a group of, usually, identical agents to collectively perform a task. There are many modes in which this group may function. A highly sought-after mode requires that the group be able to function without a central, omni-aware supervisor. The group must have distributed asynchronous processing, perception and actuation. Communication among the agents is also limited in terms of reach and connectivity. In other words, the group has to self-organize in order to carry-out the task. The belief-based modality which can only exist if the group is decentralized seems to best fit such situation.

2. Centralized versus decentralized systems

In the following, general properties of centralized and decentralized systems are discussed. A definition equating decentralization to self-organization in a collective of agents is proposed. The artificial life (AL) paradigm and the harmonic potential field approach are suggested to realize a planner that is based on this definition.

Whether it is one or more agents, successful, context-sensitive, purposive behavior requires the presence of a process for generating a regulating control action. This process receives

data from the environment, the agent/s, the target/s, and the constraints on behavior, and converts them into a control action that will successfully propel the agents, in a constrained manner, towards their goals. There are two ways for generating such a regulating action: a centralized approach, and a decentralized approach.

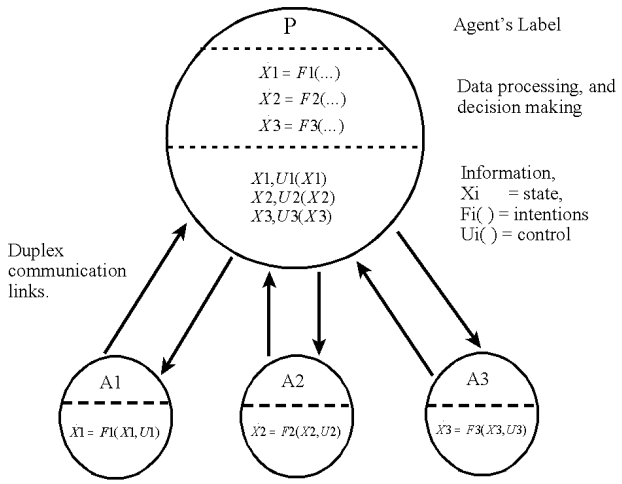


Fig. 7. Centralized approach to control

The centralized approach (figure 7) has a holistic-in-nature, top-down view to the behavior synthesis process. Here, a central agent that has a duplex communication link to each member of the group simultaneously observes the states of the agents and the environment, and processes the database in a manner that is in accordance with the aim of the group and the constraints on behavior. It then generates synchronized sequences of action instructions for each member of the group. The instructions are then communicated to the respective agents for them to progressively modify their trajectories and safely reach their destinations. In this mode of behavior, the generation of the constraint-satisfying, goal-fulfilling, conflict-free solution (i.e. sequence of state-control pair) begins by constructing the hyper action space (HAS) of the group. HAS houses the space of all admissible point actions which the agents may attempt to project. The HAS is then searched for a solution that is in turn communicated to the agents. The agents "blindly and mindlessly" execute the solution with a rigidity that is based on a trust that their actions will lead to the desired conclusion. It is a well-known fact that, in real life, any solution generated by a centralized mechanism is short lived. The dynamic nature of real environments will cause a mismatch between the conditions assumed at the time the controller begins generating the solution, and the actual conditions at the time the solution is handed to the agents for execution. Despite the attempt to alleviate this problem by equipping the agents with local sensory and decision making capabilities, centralized systems still suffer serious problems some of which are stated below:

1. Almost all centralized planning and control problems are known to be PSPACE-complete with a worst case complexity that grows exponentially with the number of agents. The large number of agents a realistic system contains will prevent the central controller from responding to environmental changes in a timely manner, if not cripple the control process altogether.

2. Centralized systems are inflexible in the sense that any changes to the characteristics of one or more agents may translate into a change in the whole HAS. This makes it necessary to repeat the expensive search for a solution. In turn, the desirable property that the size of the effort needed to adjust the control should be commensurate with the size of changes in the setting is not satisfied.
3. Centralized systems are prone to problems in communication and action synchronization. This makes it difficult to reliably operate the system even if the central planner has the assets needed to meet the demands of a realistic environment.
4. Centralized systems are not robust in the sense that the failure of one agent to fulfill its commitment towards the group could lead to the failure of the whole group.

In real life, no agent, no matter how sophisticated it is, has omniscient awareness of its surroundings, let alone infinite resources to instantly store and process data. Sometimes, even reliable communication links between the central agent and the others is difficult to establish. It may even be impossible due to the lack of universally accepted technical language, even vocabularies, for communication. The above are a few reasons why central planning strategies may not succeed in real life. Ruling out the feasibility of a central planning agent leaves only the option of the regulating control action arising from the agents themselves. The fact that the agents possess only local sensing, reasoning, and action capabilities makes it impossible to capture a complete spatial and/or temporal representation of the process. This, in turn, makes it impossible to build an HAS. As a result, the traditional way of control generation that first starts with a representation module followed by reasoning and control action generation can no longer be applied.

A major departure from such a linear, traditional way of thinking is needed. Since any finite sensory data an agent acquires does not reflect the actual content of the environment, the representation based on such data can only be classified as a belief. Under the above described conditions, an agent starts from a "seed" belief about its environment. This belief is coupled to an experiential stage that requires a sensory action continuously engaging the environment. Feedback is then applied to condition belief by experience. The control action is the outcome of this process.

Unlike the traditional approach where representation is an *a priori* that is needed to generate the control action, the suggested approach has representation as a *posteriori,* a byproduct of the action generation process. The local control synthesis modules based on the above approach are set to interact within the confines of their environment. From such nuclear activities of properly designed modules, a global regulating control action will emerge, and the group will "self-organize".

Obviously it is not feasible for agents in a large group with distinct goals to be *a priori* aware of each other's presence, to communicate with each other or with a central agent regarding advice on what action to take. As mentioned above, the only remaining option is for each agent to make its own decision on how to act based on the sensory data which the agent dynamically extracts from its local surroundings (Figure 8). Knowing that there is more than one interpretation of decentralization, the author considers a multi-agent system decentralized if each agent in the group is independent from the others in sensory data acquisition, data processing, and action projection. In a decentralized system, these faculties are configured in a mode that would give rise to coordination in the group without a coordinator. In other words, the group is capable of self-organization. Unlike centralized, top-down approaches, self-organization is a bottom-up approach to behavior synthesis where the system designer is only required to supply the individual agents with basic, "self-

control" capabilities. The overall control action that shapes the behavior of the agents evolves in space and time as a result of the interaction of the agents among themselves and with their environment.

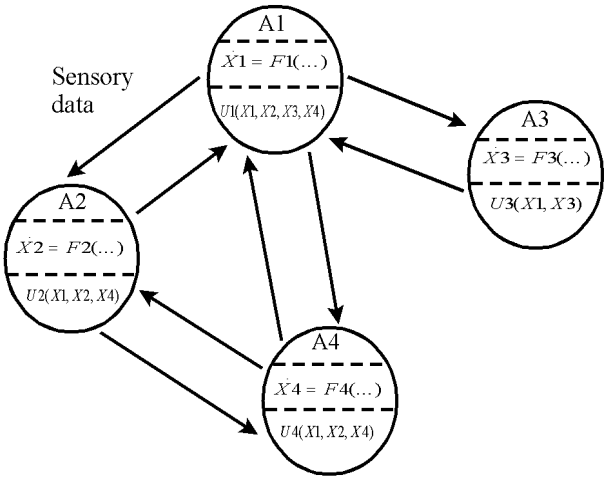


Fig. 8. Decentralized approach to control

Some properties of decentralized systems that conform to the above definition are:

1. There is no need to search or, for that matter, construct the HAS of the group in order to generate a solution. For a decentralized system, the solution emerges as a result of the agents interacting among themselves and with their environment.
2. No inter-agent communication, or communication with a supervisory agent is required. All that an agent is required to do is to observe (not communicate with) other agents in its local neighborhood. No preexisting awareness of the whole group, or the whole environment is required.
3. Synchronous behavior is an emergent phenomenon (instead of an imposed one) that results from asynchronous interaction.
4. The complexity of control in the group grows linearly with the number of agents.
5. Decentralized systems, where every one of its member agents independently sense its environment, process data, and actuate motion, form open systems that enable any agent to join or leave the group without the others having to adjust the manner in which they process information or project action.
6. Unlike centralized systems which are informationally-closed, and organizationally-open, decentralized systems are informationally-open and organizationally-closed.

The difference between centralized and decentralized systems goes far beyond the manner in which the behavior generation faculties are related to the agents. They reach as deep as the process enabling the system to generate the information needed for behavior synthesis. Centralized systems use reasoning coupled with search as the driver of the action selection process (it ought to be mentioned that function/al minimization is a form of search). The search of the system's space of possible actions for a feasible solution may be carried out in a brute force manner, or in an intelligent manner that utilizes heuristics and side information for speed. No matter what form the search assumes or how it is applied, systems relying on search have very serious problems, some of which are mentioned above, if they operate in a

dynamic environment. On the other hand, the action selection driver in decentralized systems that satisfies the above requirements is a synergy-driven evolution. In this mode of behavior information synthesis is the result of the synergetic interaction of the agents among themselves under the influence of their environment. The information that is *a priori* encoded into each agent in the form of self-capabilities to project actions is usually simple and not adequate, on its own, to handle the usually complex planning task which faces the group. It is synergetic interaction within the context of the environment that augment the level of information which the group has to a level that is sufficient for the members to carry out the task at hand (an act of knowledge amplification).

3. The artificial life approach

Despite the abundance of evolutionary techniques (e.g. neural nets, genetic algorithms, reinforcement learning, Bayesian belief networks, etc.) a relatively new area in evolutionary behavior synthesis that is called artificial life (AL) (Langton, 1988) seems to provide a powerful paradigm for explaining the behavior of decentralized systems. It also provides constructive guidelines for their synthesis. In an AL system, the members of the group are equipped with the proper elementary, *a priori* known capabilities for self-control which are called the Geno-type of behavior (G-type). On the other hand, the overall control action that actually governs the behavior of the whole group evolves in space and time as a result of the interpretation of the G-type in the context of a particular environment. The whole control action is called the Pheno-type (P-type) of behavior. This behavior cannot be, exactly, *a priori* predicted, only certain aspects of it can be *a priori* known. It is very flexible, highly adaptive, and far exceeds in complexity and informational content the G-type control. There are two requirements for constructing a proper G-type control action:

1. Each agent must individually develop a control action to drive it toward its goal. Such a control need not take into consideration the control actions generated by the other agents of the group.
2. Each agent must have the ability to generate a control that can resolve conflict with other agents through bilateral interaction.

Despite being an inherently multi-agent approach, the AL approach is applicable to the case of an isolated agent trying to synthesize a regulating control action utilizing only finite sensory and data processing resources. This is possible despite the fact that more than one agent is needed for synergetic interaction to take place. The agents needed to trigger synergy need not be physical, they can be a virtual construction of the agent concerned. For this case, the agent starts by densely spreading micro-agents, in its own image, all over the situational space the agent has the potential of occupying (Figure 9).

The only difference between the "mother-agent" and a micro-agent is that the state of the former agent evolves in space and time, while the state of a micro-agent is stagnant and immobilized to only one *a priori* known point in state space. Only the control action associated with each micro-agent is allowed to evolve. The micro-agent concept is used to construct a control action group for the agent by first covering the state space with a manifold that has locally (point-wise) extractable vector features which homogeneously cover the domain on which the control is defined. The vector features are determined by the vector partial differential operator that is used to operate on the manifold to induce a vector field that may be used to describe the action structure of the micro-agent group, therefore generating the action field of the agent. The second step is to provide each micro-agent with

the ability to generate a proper differential behavior. Differential behavior is a self-behavior where a micro-agent does not attempt to influence the other micro-agents with which it is directly interacting. Instead, it forms a soft informational coupling with them where it only observes their behavior and uses this information to derive a self-action that governs its and only its behavior in state space (figure 10). This may be achieved by constraining the vector partial differential operator that is used to emulate the actions of the micro-agents using another partial differential operator. This operator encodes how a micro-agent is going to constrain its behavior with respect to the behavior of the other micro-agents it is interacting with. The third step is to induce a proper action structure over the micro-agent group. In centralized approaches, each micro-agent has to search for the "correct" action in order to generate a group structure that unifies all the micro-agents in one goal-oriented unit. In the proposed approach, a micro-agent is only required not to exert the "wrong" actions that could result in the failure of the agent to reach the goal.

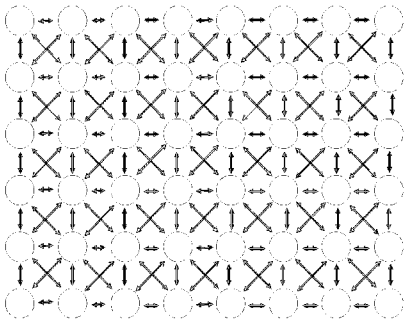


Fig. 9. A dense matrix of interacting virtual agents

Not selecting the wrong actions is not enough, on its own, for each micro-agent to restrict itself to one and only one admissible action that would constitute a proper building block of the global structure needed to turn the micro-agent group into a functioning unit. The additional effort needed to induce the global structure on the micro-agents is a result of the evolution of the behavior of the micro-agent group in time and space under the guidance of the environment (i.e. morphogenesis,(Thom, 1975)). This guidance is what eventually limits each micro-agent to one and only one action that is also a proper component in a functioning group structure. The environment guidance may be factored into the behavior generation process as state boundary conditions which play the role of self-preserving actions that the agent is *a priori* equipped with. The behavior of a micro-agent at a location which the agent believes to be harmful is constrained to an *a priori* known survival action that would drive motion away from it and towards a safe region. The environment guidance could also be in the form of instructions restricting the behavior of the agent at certain region in the admissible space (e.g. constraints on direction along which motion should proceed). The above approach was used to derive a new class of intelligent, emergent, situated, end embodied class of controller called evolutionary, hybrid, pde-ode controllers (EHPCs) that are suitable for constructing the self-control component (G-Type control) of a multi-agent system. the evolutionary, hybrid, PDE-ODE control (EHPC), Figure 11. An EHPC consists of two parts:

1. a discrete time-continuous time system to couple the discrete-in-nature data acquisition process to the continuous-in-nature control action release process;

2. a hybrid, PDE-ODE controller (HPC) to convert the acquired data into in-formation that is encoded in the structure of the micro-control action group. For more details about this type of control see (Masoud & Masoud, 2000a; Masoud & Masoud,1998; Masoud & Masoud, 1997; Masoud & Masoud,1994; Masoud & Masoud, 2000b; Masoud & Masoud, 2002).

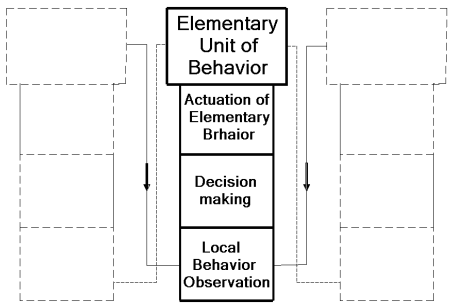


Fig. 10. Layers of functions in an interactive micro-agent

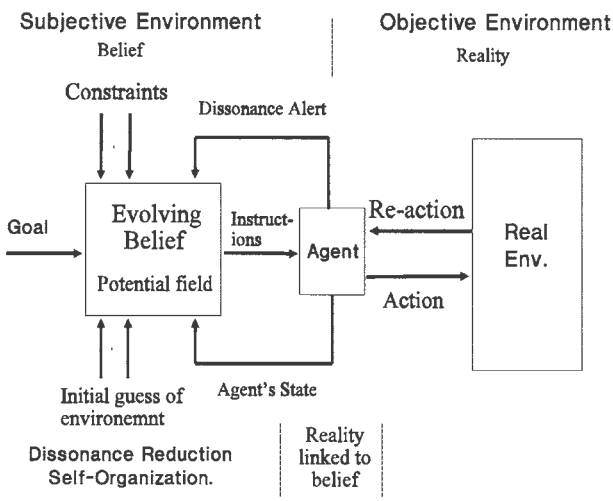


Fig. 11. A structure for an EHPC

4. The harmonic potential field approach: a background

The HPF approach is a realization of EHPCs. It mathematically captures the behavior of an AL system by first emulating the dense collective of micro-agents using a situation space cover in the form of a potential field (V) that is acted on by a differential operator (usually the gradient operator, ∇V). The vector differential elements, which may be perceived as the micro-agents, are sensitized to each other using a vector differential relation that locally imposes dependence on the behavior of these vectors. The form to which the structure of the vector differential elements converge to (i.e the interpretation of the G-type action in the context of the environment) may be controlled by boundary conditions or by making G-type

action environment-dependant. Figure 12 shows an HPF-based group action evolving in space and time until it finally reaches a form that allows it to function as desired.

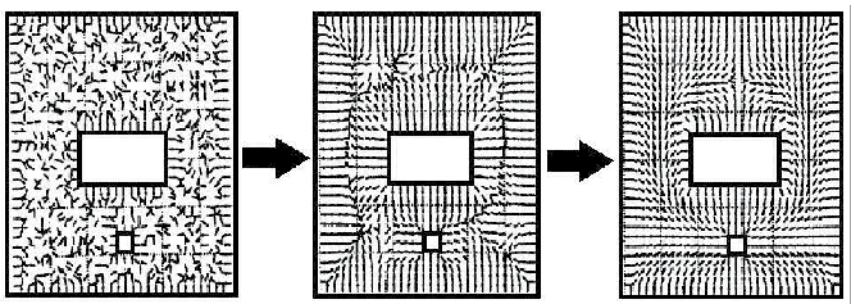


Fig. 12. Evolution of the control action in a PRF component

The harmonic potential field approach is a powerful, versatile and provably-correct means of guiding motion in an N-dimensional abstract space to a goal state subject to a set of constraints. The approach works by converting the goal, representation of the environment and constraints into a reference velocity vector field using the mechanism described above (figure 13). This reference field is usually generated from a properly conditioned negative gradient of an underlying potential field.

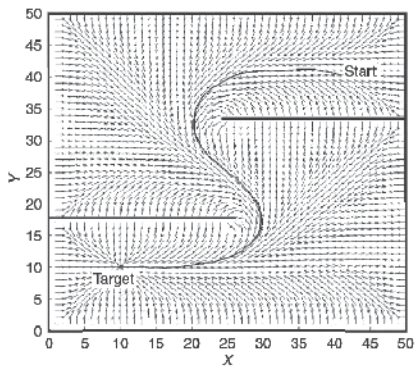


Fig. 13. The velocity field from an HPF along with the resulting trajectory

A basic setting of the HPF approach (1) is:

solve: $$\nabla^2 V(X) \equiv 0 \qquad X \in \Omega \tag{1}$$

subject to: $V(X) = 1$ at $X = \Gamma$ and $V(X_T) = 0$,

A provably-correct path may be generated using the gradient dynamical system (2):

$$\dot{X} = -\nabla V(X) \tag{2}$$

where X is a point in an abstract N-dimensional space (usually N=3), Ω is the workspace, Γ is its boundary and X_T is the target point.

Many variants of the above setting were later proposed to extend the capabilities of the HPF approach. For example, it is demonstrated that the approach can be used for planning in

complex unknown environment (Masoud & Masoud, 1997) relying on local sensing only (figure 14),

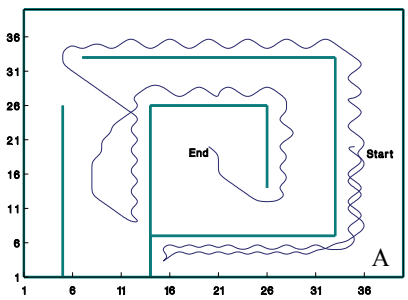


Fig. 14. HPF-based planning in unknown environments

The HPF approach can also incorporate directional constraints along with regional avoidance constraints (Masoud & Masoud, 2002) in a provably-correct manner to plan a path to a target point (figure 15),

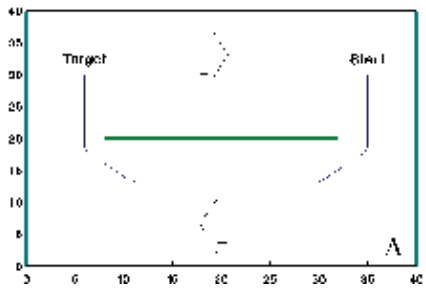


Fig. 15. HPF-based planning with directional and regional avoidance constraints

The HPF approach may also be modified to deal with inherent ambiguity (Masoud, 2009a) that prevents the partitioning of an environment into admissible and forbidden regions (figure 16),

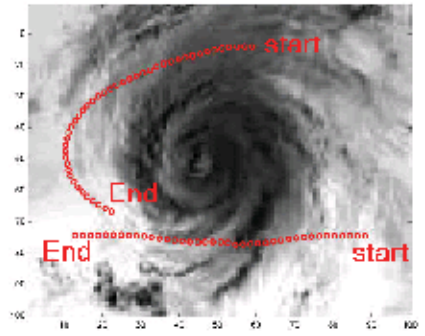


Fig. 16. HPF-based planning in non-divisible environments

It can also be adapted to deal with environments containing obstacles and a drift field which suits planning for energy exhaustive missions (figure 17).

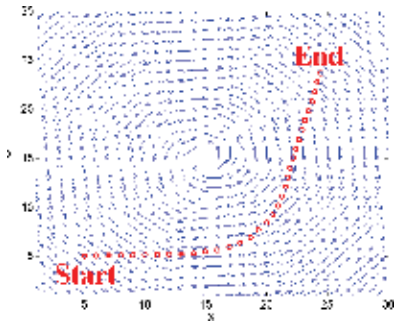


Fig. 17. HPF-based planning in the presence of drift fields

It was demonstrated in (Gupta et al, 2006) that the HPF approach can work with integrated navigation systems that can efficiently function in a real-life situation (figure 18).

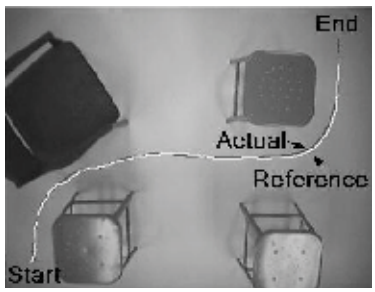


Fig. 18. HPF-based integrated navigation system

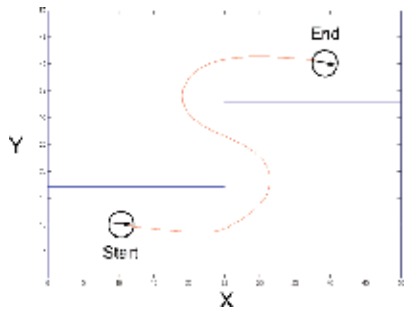


Fig. 19. HPF-based, nonholonomic, dynamical navigation

Work on extending the HPF approach to work with dynamical and nonholonomic systems (figure 19) may be found in (Masoud, 2009b; Masoud, 2010).

5. Decentralized multi-agent HPF planners

The single-agent HPF approach has amassed and is still gaining a wide array of capabilities that makes it applicable to a large number of practical and challenging problems in planning. In the remainder of this chapter, it is demonstrated by examples that the HPF approach does extend to the multi-agent case while maintaining an adaptive, evolutionary, decentralized, self-organizing nature that is compliant with the AL paradigm to behavior

synthesis. It is shown that a single HPF planner by itself or slightly modified does play the role of the G-Type controller (self-control or control protocol) in a multi-agent system.
Formally a multi-agent planner must maximize both the minimum inter-agent distance as well as the minimum distance between the agents and the clutter populating the environment while guaranteeing that each member reaches its destination in the desired manner. Unfortunately formulating the problem in this manner for a large group of agents leads to an intractable situation. Self-organizing optimization methods (Wu & Chow, 2007; Kohonen, 1997; Lampinen & Storn, 2004; Hao et al, 2007) may be used for such a purpose. They are known for their ability to handle nonlinear functions having large degrees of freedom. Neglecting the fact that these methods are not provably correct and cannot guarantee that a solution can be found if one exist, they do not provide acceptable transient behavior that allows them to serve online as trajectory controllers. Instead of seeking a formal and optimal solution to the problem, a practical solution with acceptable properties is suggested using the HPF approach. The solution sought is built around a decentralized paradigm that employs local interaction and sensing among agents (figure 20) in regulating the group's motion. The artificial life paradigm to behavior synthesis does support this mode of operation. Therefore designing a controller for the collective reduces to designing the proper G-Type controller which each agent must use. The controller should be designed such that for the overall system conflict is eliminated and goal for each member is attained while enforcing additional constraints on the individual trajectories if needed.

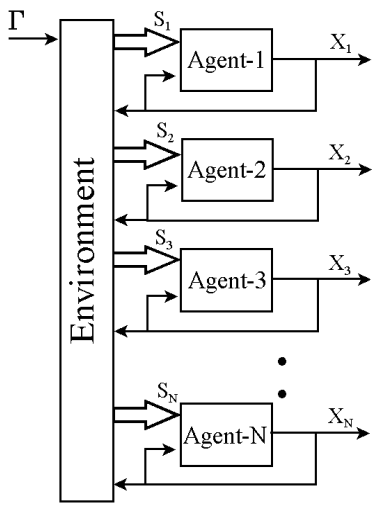


Fig. 20. Overall, decentralized system

5.1 The vector-harmonic multi-agent potential field approach

This decentralized, self-organizing, multi-agent, HPF-based method relies on local information in de-conflicting the workspace. Each agent, independently, uses an HPF method to guide it to its target in the desired manner. The control protocol individually used by the agents to reach their goals is constructed by augmenting the HPF controller with sensor-actuated, local vector potential fields. The HPF component of the protocol is called the purpose field (PRF) and vector potential component is called the conflict resolving field (CRF).

5.1.1 Formulation

In this section the problem of decentralized, multi-agent motion planning in the face of incomplete information is formulated. An agent ($D_i(x)$) is assumed to be massless, and occupy a multi-dimensional, hyper sphere ($x \in R^M$) with a radius ρ_i and a center x_i:

$$D_i(x) = \{x : |x - x_i| \le \rho_i\} \quad i=1,...,L, \tag{3}$$

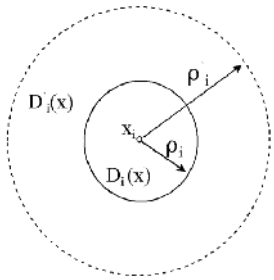


Fig. 21. Zones related to D_i

where L is the number of agents occupying the workspace (figure 21). An enlarged circular region ($D`_i(x)$) with radius $\rho`_i$ ($\rho`_i > \rho_i$) and center x_i is assumed to be surrounding $D_i(x)$:

$$\begin{aligned} D_i(x) &= \{x : |x - x_i| \le \rho_i\} \quad i=1,...,L, \\ D_i(x) &\subset D'_i(x) \end{aligned} \tag{4}$$

The ring $S_i(x)$ ($S_i(x) = D`_i(x) - D_i(x)$) surrounding $D_i(x)$ marks the region illuminated by the sensors of the i'th agent. The time between an agent sensing an event and releasing a control action (data processing and action release delay) is assumed to be small enough to be neglected in practice. Therefore, this region is a dual sensory and action zone. Besides the agents, the environment is assumed to contain static, forbidden regions (O) which the agents must not occupy at any time ($O \cap D_i \equiv \varphi$, $\forall$ t, i=1,..,L). The agents are only allowed to exist in the workspace Ω (Ω=R^M -O). The boundary of the forbidden regions is referred to as Γ (Γ= ∂ O). The destination of the i'th a gent is surrounded by the spherical region $T_i(x)$ with a center C_i (figure 22). T_i`s are chosen so that:

$$\begin{aligned} D`_i(x) &\subset T_i(x) && x_i - C_i \\ T_i(x) &\cap T_j(x) = \phi && i \neq j \\ O &\cap T_i(x) \equiv \phi && i=1,...,L. \end{aligned} \tag{5}$$

The last two conditions, respectively, mean that the goals of the different agents should not be conflicting, and should be attainable (i.e. lie inside Ω). The partial knowledge the i'th agent has about its stationary environment is represented by $\Gamma`_i$ ( $\Gamma \supseteq \Gamma`_i \supseteq \varphi$, i=1,..,L). The binary variable Q_i ($Q_i \in \{0,1\}$) marks the event of an agent discovering part of the forbidden regions not previously known to it, i.e.

$$\begin{aligned} S_i(x) \cap \Gamma &\neq \phi, \\ (S_i(x) \cap \Gamma) \cap \Gamma`_i &\equiv \phi \quad i=1,...,L. \end{aligned} \tag{6}$$

If at any instant in time (t_n), this condition becomes true, the content of Γ_i is adjusted so that:

$$\Gamma_i(t_n) = \Gamma_i(t_n - dt) \cup (S_i(x) \cap \Gamma). \tag{7}$$

If such a situation transpires, Q_i is set to 1, otherwise, its value is set to zero. The i'th agent also actively monitors its immediate neighborhood for the presence of other agents. It forms the set:

$$\chi_i(t) = \{x_j : S_i(x) \cap D_j(x) \neq \phi, j = 1,..,K_i(t), i \neq j\}\ , \tag{8}$$

where K_i is the number of agents lying in the proximity of the i'th agent at time t.

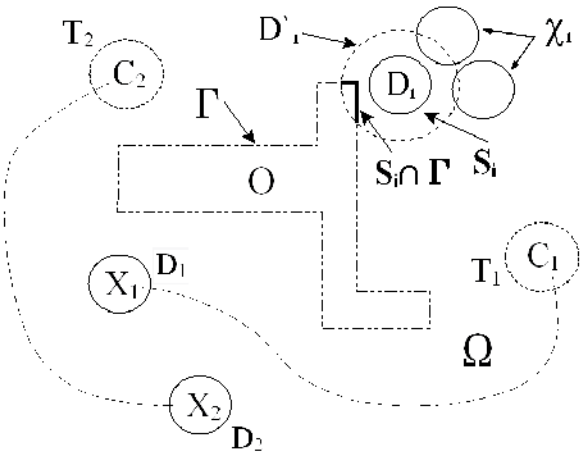


Fig. 22. Goal oriented agents in a cluttered environment

Designing the multi-agent controller requires the synthesis of the dynamical systems:

$$\dot{x}_i = h_i(x_i, C_i, Q_i, \chi_i, \Gamma_i) \qquad i=1,..,L, \tag{9}$$

such that the following constraints are satisfied for the overall system:

$$\begin{aligned} &\dot{X} = \mathbf{H}(X, C, Q, \Gamma), \\ &\lim_{t\to\infty} x_i(t) \to C_i \qquad i=1,...,L \\ &D_1(x) \cap D_j(x) \equiv \phi \qquad \forall t, 1 \neq j \\ &O \cap D_i(x) = \phi, \end{aligned} \tag{10}$$

where $x_i \in R^M$, $X=[x_1 \dots x_L]^t$, $C=[C_1 \dots C_L]^t$, $Q=[Q_1 \dots Q_L]^t$, $\Gamma=[\Gamma_1 \dots \Gamma_L]^t$, $\mathbf{H}=[h_1 \dots h_L]^t$.

5.1.2 Controller design

As discussed earlier an AL approach to behavior synthesis reduces the job of the designer to only constructing the self-controllers (G-type control) of the agents as individuals. The

overall control action that regulates the behavior of the agents as a group operating in the context of some environment (P-type control) evolves as a result of the constrained synergetic interaction among the agents. The designer is required to synthesize controls for the systems:

$$\dot{x}_i = \mathbf{u_i} = h_i(x_i, \mathbf{C}_i, Q_i, \chi_i, \Gamma_i) \qquad i=1,..,L\,. \tag{11}$$

The i'th self-control is divided into the following three components:

$$\mathbf{u_i} = \mathbf{ug_i}(x_i, C_i, Q_i, \Gamma_i) + \mathbf{uc_i}(x_i, \chi_i) + \mathbf{uo_i}(x_i, \Gamma_i), \tag{12}$$

where $\mathbf{ug_i}$ is the PRF component of the i'th self-control, $\mathbf{uc_i}$ is the CRF component, and $\mathbf{uo_i}$ is an optional control component that is included as an extra precaution against collision with stationary obstacles. It ought to be mentioned that $\mathbf{ug_i}$ includes, among other things, the ability to avoid collision. Details about how to construct $\mathbf{uo_i}$ may be found in (Khatib, 1985).

5.1.2.1 The PRF control

The PRF component of the multi-agent controller is required to guide a single agent in a stationary, cluttered environment assuming other agents are not present. In its simplest form, a PRF control assumes the form of simple vector fields that play the role of vocabularies for a language of behavior. For examples, the vector fields in figure 23 denote the behavioral vocabularies: "go to the center point", "move right", and "move right along a straight line" respectively.

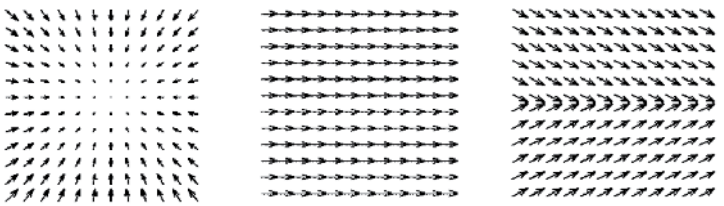


Fig. 23. Vector field-based behavioral primitives for a single agent

Such fields can be useful in simple situations where the agent is operating in a lightly cluttered environment and there are few constraints on behavior. In a realistic situation the environment may consist of heavy, irregular clutter that is not *a priori* known to the agent. The agent may also be required to constrain its behavior in the vicinity of the forbidden regions and inside the workspace. In such situations the approach of using behavioral primitives spatially and/or temporally foliated using a syntax that is determined by an algorithm or a human operator may lead to an undesired outcome. What is needed in such situations is the design of a goal-oriented, context sensitive, intelligent control action that can semantically embed the agent in the context of an environment that need not be *a priori* known. The approach adopted for PRF control synthesis is similar to the one described at the end of section 3. It assumes the lucidity of the control action. In this approach, state space is assumed to be covered with a dense set of freely-configurable control vectors. The structure that converts the individual micro-control actions into a group that can project the desired macro-control is induced on the substrate of micro-control actions using a decentralized, AL-based method (Figure 24).

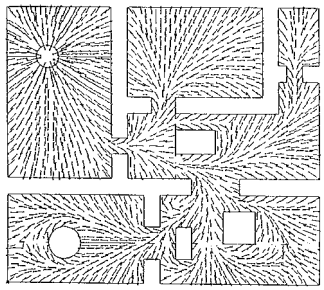


Fig. 24. A PRF field generated for a geometrically complex environment

The EHPC representing the i'th PRF control component is:

$$\mathbf{ug_i} = -\nabla V_i(x_i, C_i, Q_i(t_n), \Gamma_i(t_n)) \tag{13}$$

so that for the gradient dynamical system:

$$\begin{aligned} &\dot{x}_i = -\nabla V_i(x_i, C_i, Q_i(t_n), \Gamma_i(t_n)) \ , \\ &\lim_{\substack{n\to Z \\ t\to\infty}} x_i(t) \to C_i \qquad i=1,..,L,\ n=1,...,Z \\ &D_i(t) \cap O \equiv \phi \qquad \forall t, \end{aligned} \tag{14}$$

where n represents the n'th instant at which condition (4) becomes valid (t_n) , Z is a finite, positive integer, and ∇ is the gradient operator. At t_n , which marks the transition of $Q_i(t_n)$ from 0 to 1, first the contents of Γ_i are adjusted according to (5). The boundary value problem (BVP) below is then used for synthesizing V_i:

$$\begin{aligned} &\nabla^2 V_i(x) = 0 \qquad\qquad x \subset R^N - \Gamma_i - C_i \\ &V_i = 0|_{x \in C_i} \ \& \ V_i = 1|_{x \in \Gamma_i} \ . \end{aligned} \tag{15}$$

It ought to be mentioned that the above BVP is not the only one that can be used for generating the gradient field. Many other BVPs such as the ones reported in section 4 may be used for such a purpose.

5.1.2.2 The CRF control

There are only two ways conflict could arise in a workspace occupied by more than one purposive, mobile agents, each of which is capable of reaching its target in the absence of other agents:

1. Two or more agents may attempt to occupy the same space at the same time.
2. Two or more agents may block each other's way preventing the movement towards the targets.

A conflict resolving control ($\mathbf{uc_i}$) that can prevent the above two events from happening will enable the utilizing agent to reach its target.

It is obvious that an agent can prevent another from moving towards it, hence occupying the same space it is using, by exerting a force that is radial ($\mathbf{ucr_i}$) to its boundary (i.e. pushing the other agent away from it, figure 25). On the other hand, an agent can prevent others

from blocking its path by exerting a force that is tangential (**uct**$_i$) to its boundary (i.e. moving out of the way, figure 25)

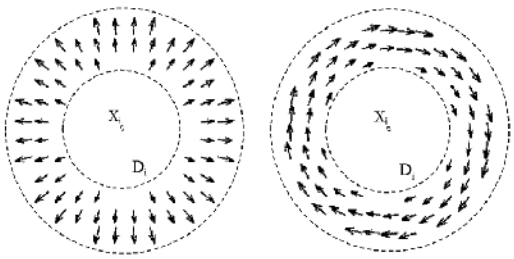


Fig. 25. Radial and Tangential components of the CRF

The CRF component is the sum of the above two actions:

$$\mathbf{uc_i} = \mathbf{ucr_i} + \mathbf{uct_i} \,. \tag{16}$$

The radial component of the control (**ucr**$_i$) may be constructed as:

$$\mathbf{ucr}_i = \sigma(|x - x_i|)\frac{\nabla Vr_i(|x - x_i|)}{|\nabla Vr_i(|x - x_i|)|}\,, \tag{17}$$

where both the weighting function σ, and the scalar potentials Vr's are positive, spherically symmetric, monotonically decreasing functions whose values are zero for $|x\text{-}x_i| \geq \rho_i$. As for **uct**$_i$, it may be constructed as :

$$\mathbf{uct}_i = \sigma(|x - x_i|)\frac{\nabla \times \mathbf{A_i}(x - x_i)}{|\nabla \times \mathbf{A_i}(x - x_i)|} \qquad \nabla\cdot\mathbf{A_i} \equiv 0, \tag{18}$$

where $\nabla\cdot$ is the divergence operator, and $\mathbf{A_i}$ is a vector potential field (Masoud & Masoud, 2000b) selected so that:

$$\nabla Vr_i(|x - x_i|)^t \nabla \times \mathbf{A_i}(x - x_i) \equiv 0\,. \tag{19}$$

For the local tangent fields to form a continuous, global tangential action that has the potential to push the interacting agents out of each other's way and prevent deadlock, all the individual tangent fields must circulate along the same direction (figure 26).

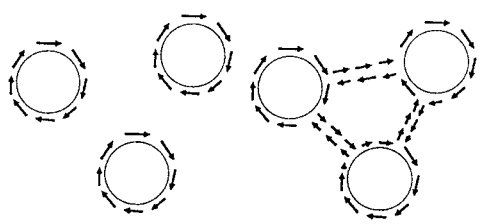


Fig. 26. Same circulations guarantees a larger one circulation along the same direction

The overall controller governing the i'th agent is described by the dynamical system:

$$\begin{aligned}\dot{x}_i &= \mathbf{ug}_i + [\mathbf{ucr}_i + \mathbf{uct}_i] + \mathbf{uo}_i \\ &= -\nabla V_i(x_i, C_i, Q_i(t_n), \Gamma_i(t_n)) + \\ &\quad \sum_{j=1}^{K_i(t)} \sigma(|x_i - x_j|)\left[\frac{\nabla Vr_i(|x_i - x_j|)}{|\nabla Vr_i(|x_i - x_j|)|} + \frac{\nabla \times \mathbf{A}_i(x_i - x_j)}{|\nabla \times \mathbf{A}_i(x_i - x_j)|}\right] + \\ &\quad \nabla Vo_i(x_i, \Gamma_i),\end{aligned} \tag{20}$$

where $\mathbf{uo_i} = \nabla Vo_i(x_i, \Gamma_i)$, and Vo_i is a scalar, repelling potential field that is strictly localized to the vicinity of the obstacles. The dynamical equation governing the behavior of the collective is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_L \end{bmatrix} = \begin{bmatrix} \mathbf{ug_1}(x_1, C_1, Q_1(t_n), \Gamma_1(t_n)) \\ \mathbf{ug_2}(x_2, C_2, Q_2(t_n), \Gamma_2(t_n)) \\ \vdots \\ \mathbf{ug_L}(x_L, C_L, Q_L(t_n), \Gamma_L(t_n)) \end{bmatrix} + \begin{bmatrix} \mathbf{uo_1}(x_1, \Gamma_1(t_n)) \\ \mathbf{uo_2}(x_2, \Gamma_2(t_n)) \\ \vdots \\ \mathbf{uo_L}(x_L, \Gamma_L(t_n)) \end{bmatrix} + \begin{bmatrix} \sum_{\substack{j=2, \\ j\neq 1}}^{K_1(t)} \mathbf{uc_1}(x_i - x_j) \\ \sum_{\substack{j=1, \\ j\neq 2}}^{K_2(t)} \mathbf{uc_2}(x_i - x_j) \\ \vdots \\ \sum_{\substack{j=1, \\ j\neq L}}^{K_L(t)} \mathbf{uc_L}(x_i - x_j) \end{bmatrix} \tag{21}$$

5.1.3 Motion analysis

A detailed proof of the ability of the agents, individually, to reach their respective destinations in an unknown cluttered environment may be found in (Masoud & Masoud, 2002; Masoud & Masoud, 2000a). While it is not hard to see that the ability of the robots to avoid collision with each other and with the obstacles can be guaranteed by making the barrier controls ($\mathbf{uo_i}$, $\mathbf{ucr_i}$) strong enough (some techniques set the strength of the control to infinity at the inner boundary of the robots (Khatib, 1985), their ability to converge to their respective destinations, as a group, needs careful examination. In the following it is shown that the first order dynamical systems in (20) are potentially capable of driving the robots from anywhere in the workspace to their respective destinations provided that the narrowest passage in the workspace is wide enough to allow the largest two robots to pass at all times.

Here, it is shown that under certain conditions the solution of the system in (21) is globally, asymptotically stable. The proof is dependent on a theorem by LaSalle (Theorem-3, (LaSalle, 1960), pp. 524]. The theorem is restated below with minor changes to the notations.

Theorem: Let $\Xi(X)$ be a scalar function with continuous first partials with respect to X. Assume that:

$$\begin{aligned} &1- \quad \Xi(X) > 0 && \forall\, X \neq C, \\ &2- \quad \dot{\Xi}(X) \leq 0 && \forall\, X\,. \end{aligned} \tag{22}$$

Let E be the set of all points where $\dot{\Xi} = 0$, and M be the largest invariant set in E. Then every solution of the system:

$$\dot{X} = \mathbf{H}(X, C, Q, \Gamma') \tag{23}$$

bounded for $t \geq 0$ approaches M as $t \to \infty$.

Proposition-1: For the system in (21), $\exists$ a set of **uct**'s that can guarantee

$$\lim_{t \to \infty} X(t) \to C, \tag{24}$$

provided that: 1- for the gradient dynamical systems:

$$\begin{aligned} &\dot{x}_i = -\nabla V_i(x_i, C_i, Q_i(t_n), \Gamma'_i(t_n)), \\ &1- \quad \lim_{t \to \infty} x_i(t) \to C_i \qquad\qquad i=1,\ldots,L \\ &2- \quad D_i \cap D_j \equiv \phi, \qquad\qquad i \neq j \\ &\qquad D_i \cap O \equiv \phi, \\ &3- \quad \forall x' \in \Omega \quad \exists xc, \ni \\ &\qquad x' \in \{ x : |x - xc| \leq \xi \} \subset \Omega \end{aligned} \tag{25}$$

where $\xi = \rho'_1 + \rho'_2$, where ρ'_1 and ρ'_2 are the expanded radii of the two largest robots in the group. The third condition guarantees that nowhere in Ω will the geometry of the environment prevent the agents from resolving the conflict, and instead forces them to project motion along environmentally-determined degrees of freedom (Figure 27) that may not lend themselves to the resolution of the conflict (a restrictive environment).

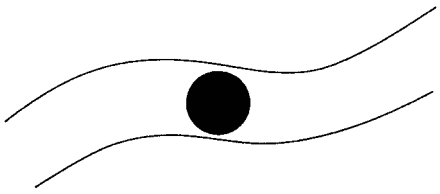


Fig. 27. Restrictive environments force *a priori* determined spatial movement patterns

By guaranteeing that there always exists a local, simply-connected region that is large enough to enable any two robots to interact, guarantees that whatever pattern of motion which the agents arrive at to resolve the conflict can be realized.

Proof: consider the following Liapunov function candidate (LFC):

$$\Xi(X) = \sum_{i=1}^{L} V_i(x_i) \ , \tag{26}$$

where $V_i(x_i)$ is used to refer to $V_i(X_i, C_i, Q_i(t_n), \Gamma'_i(t_n))$, and $Vo_i(x_i)$ refers to $Vo_i(x_i, \Gamma'_i)$. It was shown in (Masoud & Masoud, 2002; Masoud & Masoud, 2000a) that harmonic potential fields are LFCs, i.e. $V_i(x_i)=0$ for $x_i = C_i$, and $V_i(x_i)>0$ for $x_i \neq C_i$. Therefore the above sum is a

valid LFC, i.e. $\Xi(X) = 0$ for X= C, and $\Xi(X) > 0$ for $X \neq C$. The time derivative of Ξ may be computed as:

$$\begin{aligned}\frac{d}{dt}\Xi &= \sum_{i=1}^{L} \nabla V_i(x_i)^t \frac{d}{dt} x_i \\ &= \sum_{i=1}^{L} \nabla V_i(x_i)^t [-\nabla V_i(x_i) + \\ &\sum_{\substack{j=1 \\ i\neq j}}^{K_i(t)} \sigma(|x_i - x_j|)\left[\frac{\nabla V r_i(|x_i - x_j|)}{|\nabla V r_i(|x_i - x_j|)|} + \frac{\nabla \times \mathbf{A_i}(x_i - x_j)}{|\nabla \times \mathbf{A_i}(x_i - x_j)|}\right] + \\ &\nabla V o_i(x_i)].\end{aligned} \tag{27}$$

The above expression is examined term by term to determine the nature of the time derivative of Ξ. It is obvious that the term:

$$\sum_{i=1}^{L} -\nabla V_i(x_i)^t \nabla V_i(x_i) \tag{28}$$

is negative definite with a zero value (stable global equilibrium) at and only at $x_i = C_i$, i=1,..,L, (X=C). As for the term:

$$\sum_{i=1}^{L} \nabla V_i(x_i)^t \nabla V o_i(x_i), \tag{29}$$

One must first notice that ∇ Vo is a local field that is strictly limited to a thin narrow region surrounding Γ. Its value is zero everywhere else in Ω. By construction, the field lines of ∇ Vo_i emanate normal to Γ (in order to drive the robot away from the obstacles):

$$\nabla V o(x_i) = \begin{cases} \alpha(x_i)\mathbf{n} & x_i \in \Gamma_i \\ 0 & \text{elsewhere} \end{cases} \tag{30}$$

where **n** is a unit vector that is normal to Γ, and α is a smooth, positive, monotonically decreasing scalar function with a value set to zero a small distance (ε) away from the boundary of the obstacles (x_i^t **n**), i.e. $\alpha(x_i)=0$ for $x_i^t \mathbf{n} > \varepsilon$. The BVPs used for constructing the potential field associated with the PRF control (V_i) admits only two types of basic boundary conditions (BCs):

1- homogeneous Neumann BCs:

$$\frac{\partial}{\partial \mathbf{n}} V_i(x_i) = \nabla V_i(x_i)^t \mathbf{n} \equiv 0, \qquad x_i = \Gamma_i \tag{31}$$

2-homogeneous Dirichlet BCs: $V_i(x_i) = 1$

which in turn makes: $$\frac{\partial}{\partial \mathbf{n}} V_i(x_i) = \nabla V_i(x_i)^t \mathbf{n} < 0, \tag{32}$$

(i.e. the maximum of V_i is achieved at $x_i = \Gamma_i$ and its value decreases with motion away from $x_i = \Gamma_i$). As a result the above term is in one of the two forms in (33):

$$\begin{aligned} &\sum_{j=1}^{\cdot} \nabla V_i(x_i)^t \nabla Vo_i(x_i) \equiv 0, \\ &\sum_{i=1}^{\cdot} \nabla V_i(x_i)^t \nabla Vo_i(x_i) < 0, \qquad\qquad x_i = \Gamma_i. \end{aligned} \tag{33}$$

As for the second term of (27), it ought to be mentioned that forces surrounding the mobile agents (CRFs) have a local, reactive passive nature. In view of the above, this guarantees that no unbounded growth in the magnitude of the x_i's can occur. The worst case is for those forces to cause a deadlock in motion (i.e, X - C = constant, $t \to \infty$). Since in the worst case scenario, motion will be brought to a halt (i.e, $\dot{\Xi} = 0$), also taking into consideration the negative definiteness of the other terms, the time derivative of Ξ is always less than or equal to zero:

$$\dot{\Xi} \le 0. \tag{34}$$

If the i'th robot is in static equilibrium (assuming that the target was not reached), the following identity must hold:

$$\sum_{j=1}^{K_i(t)} \sigma(\|x_i - x_j\|) \left[\frac{\nabla Vr_i(\|x_i - x_j\|)}{|\nabla Vr_i(\|x_i - x_j\|)|} + \frac{\nabla \times \mathbf{A}_i(x_i - x_j)}{|\nabla \times \mathbf{A}_i(x_i - x_j)|} \right] = \nabla Vo_i(x_i) - \nabla V_i(x_i). \tag{35}$$

Therefore, the set E is equal to:

$$\begin{aligned} &E = E1 \cup E2 = \{x_i : \frac{d}{dt}\Xi = 0\}, \\ &E1 = \cup_i E1_i, \quad E1_i = \{x_i : x_i = C_i\} \qquad\qquad i=1,...,L \\ &E2 = \cup_i E2_i, \end{aligned} \tag{36}$$

$$E2_i = \{x_i : \nabla Vo_i(x_i) - \nabla V_i(x_i) + \sum_{\substack{j=1 \\ i \ne j}}^{K_i(t)} \sigma(\|x_i - x_j\|) \left[\frac{\nabla V_i(\|x_i - x_j\|)}{|\nabla V_i(\|x_i - x_j\|)|} + \frac{\nabla \times \mathbf{A}_i(x_i - x_j)}{|\nabla \times \mathbf{A}_i(x_i - x_j)|} \right] = 0, \quad x_i \ne C_i\}. \tag{37}$$

The largest invariant set $M \subset E$ is the subset of E that satisfies the equilibrium condition on (21). Before computing M, let us first examine if E2 is an equilibrium set of system (21). For this case the system forces may be computed using the equation:

$$h_i = \nabla Vo_i(x_i) - \nabla V_i(x_i) + \sum_{\substack{j=1 \\ i \ne j}}^{K_i(t)} \sigma(\|x_i - x_j\|) \left[\frac{\nabla Vr_i(\|x_i - x_j\|)}{|\nabla Vr_i(\|x_i - x_j\|)|} + \frac{\nabla \times \mathbf{A}_i(x_i - x_j)}{|\nabla \times \mathbf{A}_i(x_i - x_j)|} \right]. \quad i=1,..,L. \tag{38}$$

It should be noticed that if the second condition of (23) holds, the magnitude of the radial reaction forces (∇Vo_i, and ∇Vr_i) is determined by the self-forces (∇V_i), and the geometric

configuration the robots assume during deadlock. On the other hand, the magnitude of the circulating forces ($\nabla \times \mathbf{A_i}$) is totally independent of the self-forces. Since the individual circulating forces are made to rotate in the same direction, such fields contain no singularities (Figure 28). In other words, the circulating forces never vanish, always guaranteeing that relative motion among the agents can be actuated. Hence, their strength can be independently set by the designer anywhere in the workspace. Since the goal is to eliminate E2 from M, this freedom is used to guarantee that $h_i \neq 0 \quad x_i \neq C_i$, i=1,..L.

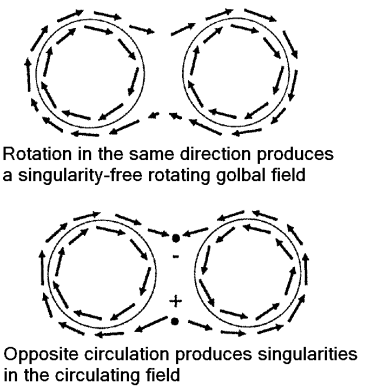


Fig. 28. A global tangent field constructed by local tangent fields circulating in the same direction is free of singularities

Since the self-forces are generated from the gradient flow of a harmonic potential, their magnitude in Ω is bounded:

$$\left|\nabla V_i(x_i)\right| \leq B_i, \quad x_i \neq C_i \qquad i=1,..,L, \tag{39}$$

where B_i is a positive, and finite constant. Also notice that it is not possible for the magnitude of the passive reaction forces to exceed that of the self-forces. Therefore, a conservative choice of the magnitude of the circulating field that would guarantee that E2 is not an equilibrium set of (23) is

$$\left|\nabla \times \mathbf{A_i}(x_i - x_i)\right| \geq \sum_{i=1}^{L} B_i. \tag{40}$$

It should also be noticed that if the third condition of (25) is not satisfied (i.e. there is not enough free space for the largest two robots to move at all times) and the circulating fields have to push against a static obstacle (a static obstacle can exert infinite reaction force), no realizable choice of B_i's would exist to satisfy condition (38). The above treatment amounts to the simple physical fact that whenever the radial reaction forces of one or more robots are in equilibrium the circulating forces intervene to pull the system out of deadlock. If the above condition is satisfied, E2 is eliminated from M. Also, since the robots have convex geometry, no equilibrium paths can form trapping one or more robots in a limit cycle. As for E1, the fact that the T_i's are taken so that $D_i \subset T_i$, guarantees that once the robots reach their respective destinations, no interactions among their fields can happen (i.e $\mathbf{uc_i} = 0$, and $\nabla Vo_i = 0$, i=1,..,L). Also since:

$$\nabla \mathbf{V_i}(\mathbf{x_i}) = 0, \quad \mathbf{x_i} = \mathbf{C_i}, \tag{41}$$

system (21) reduces to:

$$\dot{x}_i = 0, \quad x_i = C_i \tag{42}$$

making the largest invariant set equal to:

$$M = \cup_i \{x_i : x_i = C_i\} \qquad i=1,..,L. \tag{43}$$

Therefore, according to LaSalle's theorem, the robots will globally, asymptotically converge to their respective destinations, i.e.

$$\lim_{t \to \infty} x_i \to C_i \qquad i=1,..,L \tag{44}$$

As mentioned earlier, the suggested planner is complete provided that conditions (25) and (40) hold. To examine why imposing the third condition of (25) is necessary for the suggested planner to guarantee completeness, begin by noting that behavior, in general, has two components: a spatial one that consists of a vector field that assigns to each point in the workspace a direction along which motion should proceed, and a temporal one which consists of a scalar field that assigns a speed to each point in the workspace. Therefore, completeness for a general class of workspaces implies the existence of a spatio-temporal pattern of behavior which, if executed by the agents, leads to the satisfaction of the goal. In general, tractable environments, where a solution exists provided that behavior be spatially and temporally manipulated, the environment may at one stage deprive the planner from the ability to fully manipulate spatial behavior by forcing one agent or more to follow predetermined spatial behavioral patterns that are set by the geometry of the workspace (figure 27). If such a situation occurs, the planner can only resolve the conflict by manipulating the temporal component of behavior (i.e speed up or slow down the movements of the agents, as well as halt motion or reverse it). Since the suggested planner is totally reliant on manipulating spatial behavior only, it may fail if it encounters situations where both spatial and temporal behavior are to be manipulated. The third condition of (25) guarantees that the environment will never be able to prevent the planner from spatially manipulating behavior in order to resolve a conflict. In a recent work by the author (Masoud & Masoud, 2002; Masoud & Masoud, 2000a), a method for synthesizing a PRF control component that can jointly enforce regional avoidance, and directional constraints, may be used to guarantee that deadlock will not happen in environments with tight passages (see the last example, figures-[39-44]). Unfortunately, this approach for avoiding deadlock may reduce the set of potential solutions to the multi-agent planning problem. In other words, the controller will no longer be complete.

5.1.4 Results

Several simulation experiments were conducted to explore the behavior of the suggested method. Each example is presented as a sequence of frames with each frame depicting the state of the robots at different instants of the solution. The notation used is the same as that in the theoretical development (i.e. D_i represents the i'th robot, x_i its center, and C_i the center of the target zone).

5.1.4.1 A basic example

In figure 29 two robots sharing the same obstacle-free workspace are required to exchange positions. In doing so, each robot makes the simple, but naive, decision of moving along a straight line to the target. Despite the apparent conflict which each is heading towards, each robot proceeds with its plan as if the selected action is conflict-free. Once the conflict is in a phase that is detectable by the local sensors each robot has, corrective actions are taken by each to modify their behavior in order to resolve the conflict (i.e. the CRF control component is activated). As mentioned before, the "seed" CRF activities consist of a component to prevent collision, and another to move the agents out of each other's way. Once the conflict is resolved, the behavior modification activities dissipate and guidance is fully restored to the PRFs (figure 30).

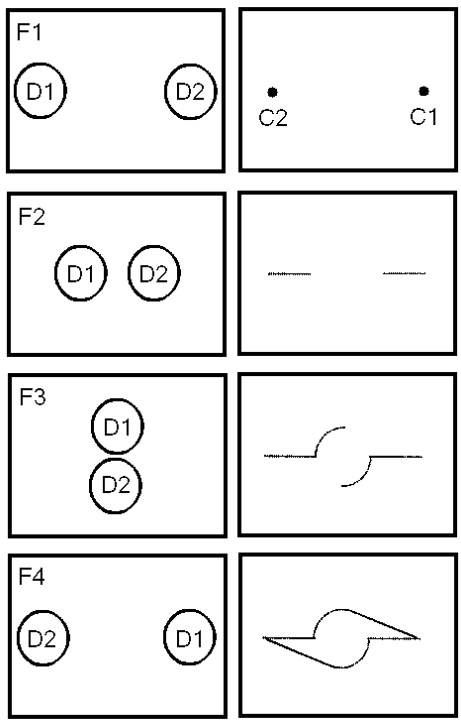


Fig. 29. Two robots exchanging positions

In figure 31, three robots operating in an obstacle-free space, and initially positioned on the vertices of an equilateral triangle are required to proceed towards their symmetric targets. As in the two-robot example, each robot chooses to proceed along a straight line to its target ignoring the apparent conflict to which this choice leads. For this case the response of the robots, once a conflict is detected, exhibits an interesting emergent nature. By reducing the degrees of freedom of the system from six to one, the three robots act as one rotating body to position themselves where each can proceed unimpeded towards its target. It is interesting to note that without being *a priori* programmed to do so, the robots choose to cooperate in order to resolve the conflict. This cooperation is manifested as a reduction in the degrees of freedom of the system during the period of the conflict.

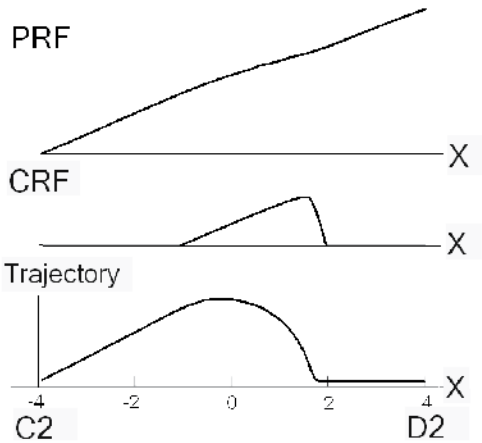


Fig. 30. CRF activities dissipate after conflict is resolved (trajectory of D2)

5.1.4.2 Fault tolerance

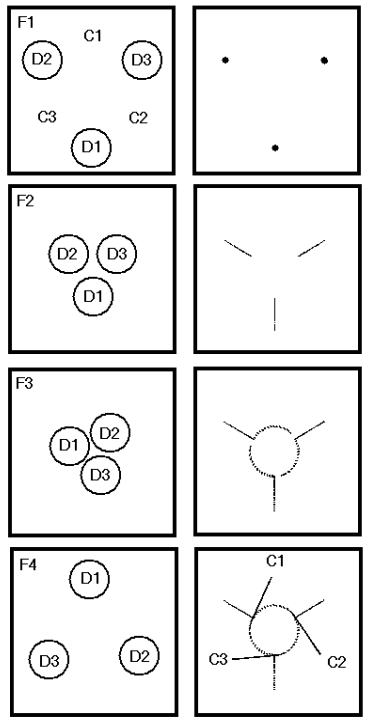


Fig. 31. Three robots moving to their goals, all functioning

In a centralized system the supervisory control assigns each agent the duties it has to fulfill for the whole group to avoid conflict. If one agent fails to fulfill its obligation towards the group, the whole group may be affected. In decentralized systems, conflict evasion has a lucid nature where conflict evasion activities dynamically shift from the unable, or

unwilling agents, to the remaining functional agents. Here, an agent's role keeps adapting to the situation in a manner that would, to the best of the agent's ability, enable all the agents (this includes the offending agents) to reach their targets. The following example examines this intriguing property of decentralized systems. In figure 32, a setting similar to the one in figure 31 is used. The only difference is that D2 refuses to participate in conflict resolution and, instead, follows the plan encoded by its PRF requiring it to move along a straight line to its target. As can be seen, the remaining two agents adjust their behavior to compensate for the intransigence of D2 in such a manner that allows all the agents to reach their destinations.

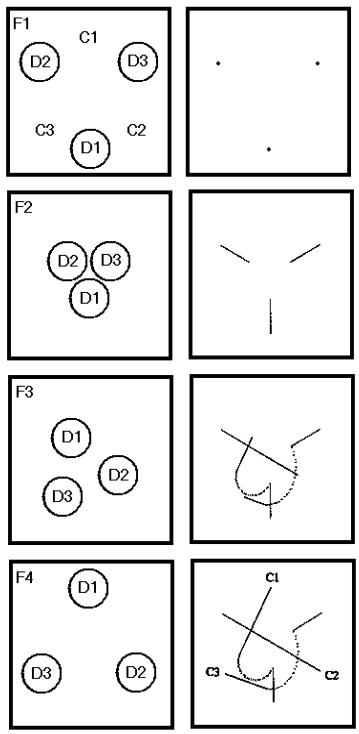


Fig. 32. Three robots moving to their goals, D2 malfunction

5.1.4.3 Morphogenesis of CRF activities

The action of the controller may be mistaken with that of a controller equipped with simple reflexive capabilities. One should keep in mind that the simple actions of collision avoidance and tangent motion are nothing but the G-type of the control. The G-type control should only be viewed as the kernel of the global control action (P-type) which is the one actually controlling the systems. The, more complex, P-type control action emerges from the seed G-type control in a flexible, situation-responsive manner. Figure 33 shows the tangent circulating fields of ten agents approaching each other. As can be seen, once the agents start to get close to each other, the fields begin to interact and their structure begins to gradually mutate until it finally assumes a global form that is very different from the form of the individual fields.

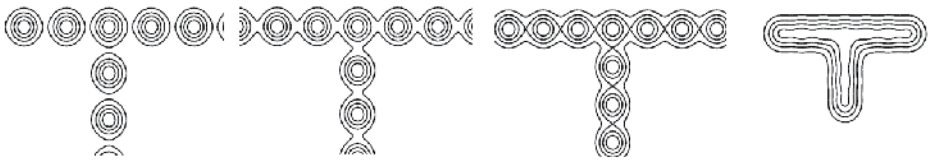


Fig. 33. Context-sensitive evolution of CRF activities

5.1.4.4: Self-organization

In the following two examples the evolutionary, cooperative, self-organizing nature of the controller is clearly demonstrated. In Figure 34 two groups of four robots each are moving in opposite directions along a road with side rails blocking each other's way. The goal is for the left group to move to the right side, and right group to move to the left side. The groups collectively solve the problem by forming right and left lanes and confining the motion of each group to one of the lanes. It should be noted that lane formation is a high-level, holistic organizational activity that fundamentally differs from the local capabilities with which each robot is originally equipped.

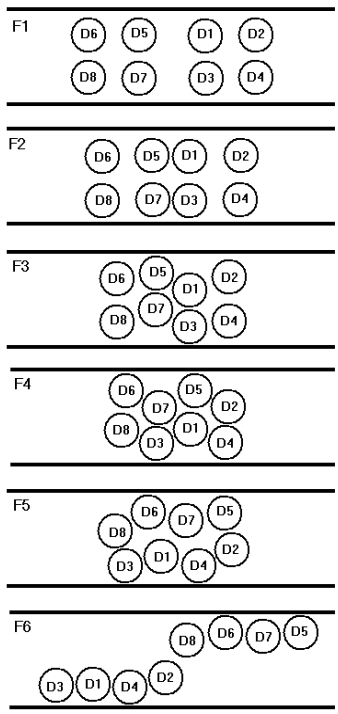


Fig. 34. Two groups of robots passing each other in a confined space

In Figure 35 eight robots are confined in a box with very little room to move. The goal is for D1 to move to C9. The robots collectively reach a solution that efficiently utilizes free space. The robots solve the problem by keeping the center robot stationary, with the remaining robots rotating around it until D1 reaches its target.

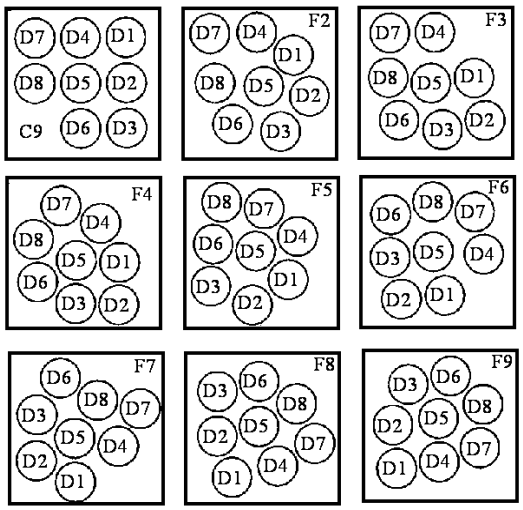


Fig. 35. A group of robots self-organizing to allow D1 to reach C9

5.1.4.5 CRF fields strength and deadlock prevention

In the following example the importance of the circulating fields for conflict resolution is demonstrated. Here a group of eight agents each is required to hold its position, except for D8 which is required to move to C8. No circulating field are used in figure 36. As can be seen, while D8 managed to pass the first group of agents, it got trapped in a deadlock formation when it attempted to pass the second group. In figure 37 circulating fields are added. As can be seen D8 is able to reach its target, and the remaining agents maintained their original positions.

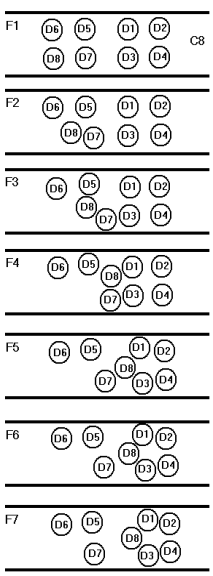


Fig. 36. D1-D7 hold positions, D8 moves to C8, no Circulating fields

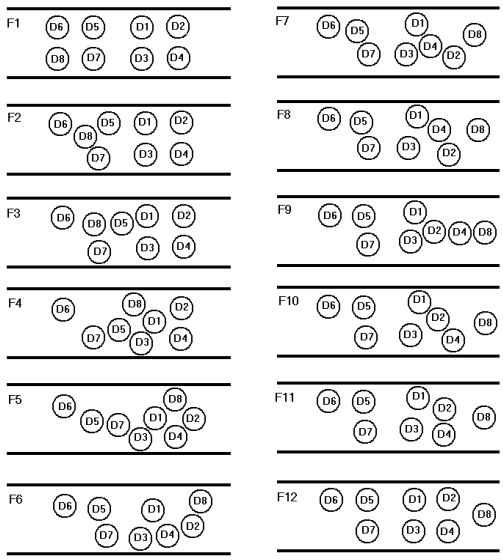


Fig. 37. D1-D7 hold positions, D8 moves to C8, Circulating fields present

5.1.4.6 All purposive agents in a congested space

In Figure 38, the difficult planning task of exchanging positions in a confined area is assigned to the robots. The order of the exchange is as follows D1 ↔ D6, D2 ↔ D5, D3 ↔ D8, D4 ↔ D7. As can be seen, the group successfully carried out the task.

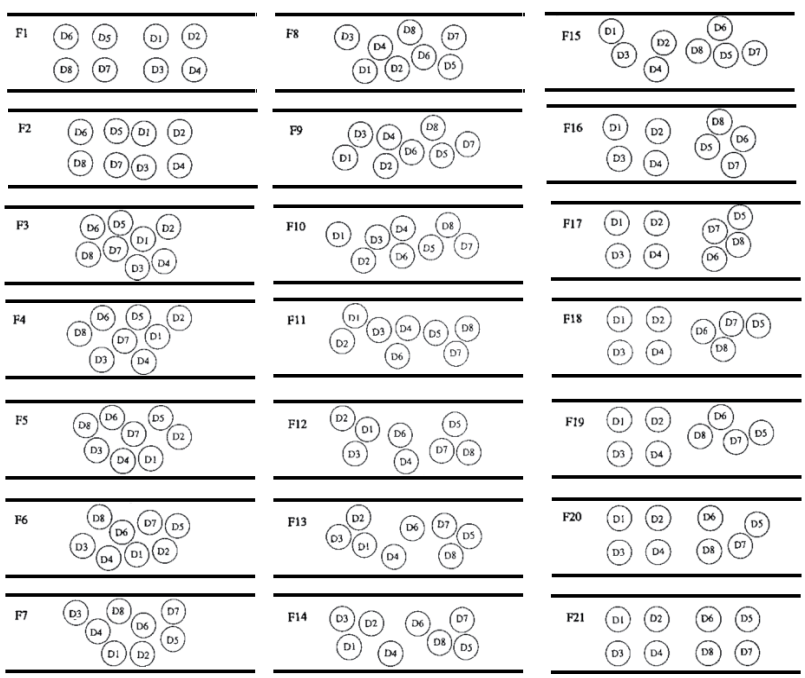


Fig. 38. A group of robots exchanging positions in a confined space

5.1.4.7 Environments with tight passages

While the third condition of (25) is by no means stringent (after all, it is only reasonable for a two-way street to be wide enough to allow two vehicles to pass at the same time), there are, nevertheless, environments with tight passages that have only room for one robot at a time. In such a situation there are no guarantees that the multi-agent planner will function properly. One way to remedy this situation is to mark a tight passage as a one-way street (i.e. constrain motion in such passages to become unidirectional). This may be accomplished by using the NAHPF-based EHPC scheme in (Masoud & Masoud, 2002; Masoud & Masoud, 2000a) for synthesizing the PRF control component of the multi-agent controller. The following example illustrates the use of NAHPFs for such a purpose.

Consider the workspace in figure 39. Two robots D1 and D2 are required to exchange positions. As can be seen, the passages in Ω are not wide enough for the two robots to pass at the same time.

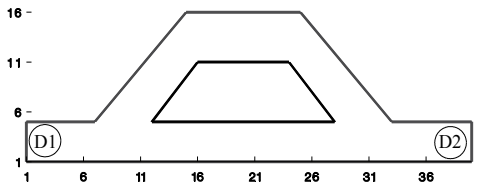


Fig. 39. A workspace with tight passages

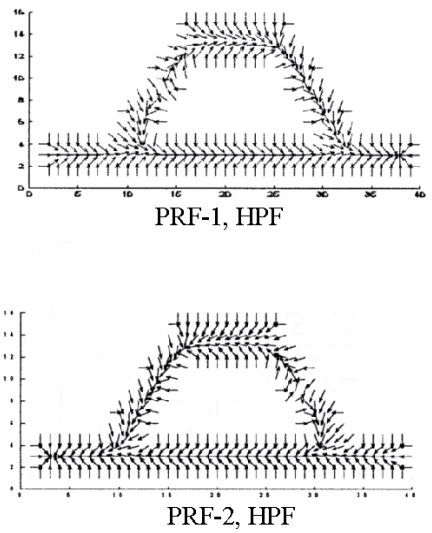


Fig. 40. PRF components

Figure 40 shows the HPF-based PRFs for both D1 and D2. Figure 41 shows, using snapshots, the locations of the robots that are generated by the multi-agent controller at different instants of the solution. As can be seen, an unresolvable conflict arises between D1 and D2. Figure 42 shows the NAHPF-based PRFs for D1 and D2. Figure 43 shows the corresponding locations of the robots at different instants in time. As can be seen, conflict was resolved by marking the tight passages as one-way streets.

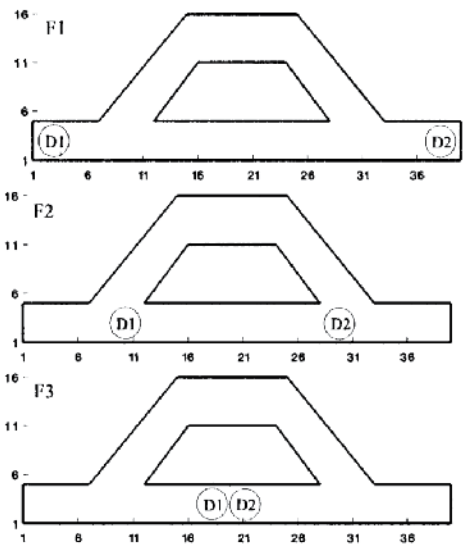
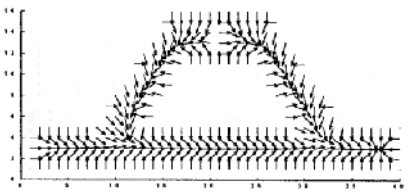
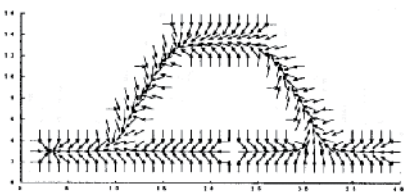


Fig. 41. Deadlock caused by a tight passage



PRF-1, NAHPF



PRF-2, NAHPF

Fig. 42. PRF components, NAHPF-based EHPCs

Unfortunately, using NAHPF-based EHPCs to avoid conflict in environments with tight passages has some drawbacks. Marking a passage as a one-way street leads to a loss in potentially realizable solutions. Consider for example the environment in figure 44. It is obvious that a solution exist to move D1 to the location of D2 and vice versa. Marking the tight passage as a one way street makes it impossible for D2 to move to the left. To solve the tight passage problem, the planner must jointly manipulate spatial and temporal behavior.

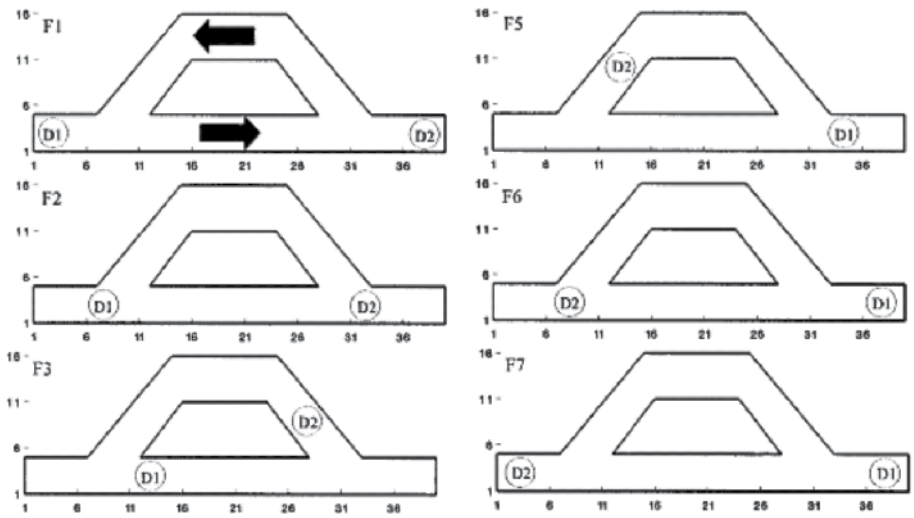


Fig. 43. NAHPF-PRF component can resolve deadlock in spaces with tight passages

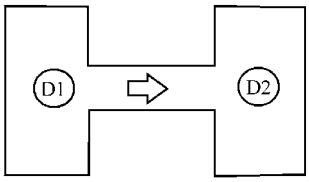


Fig. 44. NAHPF-PRF could lead to a loss in potential solutions

5.2 Individually-motivated, multi-agent, HPF-based planner

The single agent HPF has a social nature that allows it to co-exist with other agents using a similar navigation procedure in the same cluttered space (a proof is also provided in section-5.2.1). This makes a single-agent, HPF planner a valid G-type controller (figure 45) in a multi-agent system. The reason for that is: the HPF approach treats other agents as obstacles to be avoided. Hence the same procedure used to avoid clutter can also be used to accommodate the presence of the other agents.

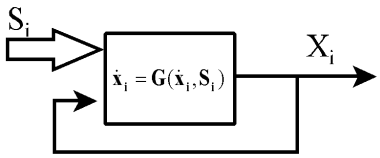


Fig. 45. The G-type controller

The agent could use the basic BVP shown below (45) for generating the self-control action or a BVP corresponding to any of the HPF setting discussed earlier in this chapter. The trajectory can be easily generated by the gradient dynamical system resulting from the computed potential field (figure 46).

$$\begin{aligned} \nabla^2 V(x_i) &= 0 \qquad\qquad x_i \in \Omega \ . \\ V(x_i)\big|_{x_i \in S_i} &= 1 \ \ \& \ \ V(x_i)\big|_{x_i \in \Gamma_i} = 0 \\ \dot{x}_i &= -\nabla V(x_i) \ . \end{aligned} \tag{45}$$

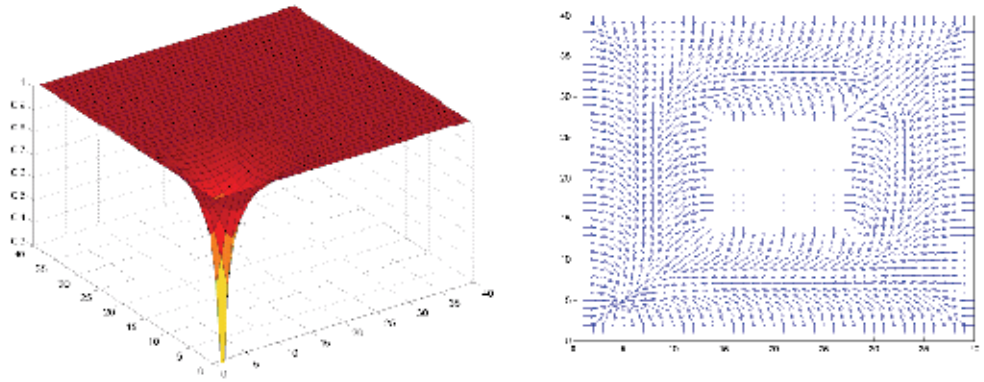


Fig. 46. Potential and gradient guidance field for a goal-oriented, G-type controller

5.2.1 Analysis: the goal-oriented case:

In this section proofs of the ability of the goal-oriented multi-agent controller to avoid obstacles and steer each of its members to its target are provided**.**

Proposition: If $x_i(0) \in \Omega$, the motion steered by the gradient dynamical system in (45) will always remain inside Ω (i.e. $x_i \cap S_i \equiv \varphi \ \ \forall t, \ \forall i$) .

Proof: Consider the part of Ω near a forbidden region (S_i). Let $n(x_i)$ be a vector that is normal to the surface of the obstacle. Let S_i' be a region created by infinitesimally expanding the forbidden region S_i such that $S_i \subset S_i'$. The radial derivative of $V(x_i)$ along S_i may be computed as:

$$\frac{\partial V}{\partial n} = \frac{V(x_i) - V(x_i')}{\Delta r} \tag{46}$$

where x_i' is taken as the minimum distance between x and S_i' and Δr is a positive differential element. Since by the maximum principle the value of the potential in Ω is less than 1, and x_i' lies inside Ω, the radial derivative of the potential along $n(x_i)$ is negative, i.e.

$$n(x_i)^T \nabla V(x_i) < 0 \tag{47}$$

Since motion is steered using the negative gradient of V, the agent will be pushed away form S_i and x_i will remain inside Ω.

Definition: Let V(X) be a smooth (at least twice differentiable), scalar function ($V(X): R^N \rightarrow R$). A point Xo is called a critical point of V if the gradient vanishes at that point ($\nabla V(Xo)=0$); otherwise, Xo is regular. A critical point is Morse, if its Hessian matrix (H(Xo)) is nonsingular. V(X) is Morse if all its critical points are Morse (Milnor, 1963).

Proposition: If V(X) is a harmonic function defined in an N-dimensional space (R^N) on an open set Ω, then the Hessian matrix at every critical point of V is nonsingular, i.e. V is Morse.

Proof: There are two properties of harmonic functions that are used in the proof:

1. a harmonic function (V(X)) defined on an open set Ω contains no maxima or minima, local or global in Ω. An extrema of V(X) can only occur at the boundary of Ω,
2. if V(X) is constant in any open subset of Ω, then it is constant for all Ω.

Other properties of harmonic functions may be found in (Axler et al, 1992).

Let Xo be a critical point of V(X) inside Ω. Since no maxima or minima of V exist inside Ω, Xo has to be a saddle point. Let V(X) be represented in the neighborhood of Xo using a second order Taylor series expansion:

$$V(X) = V(Xo) + \nabla V(Xo)^T (X - Xo) + \frac{1}{2}(X - Xo)^T H(Xo)(X - Xo) \qquad |X\text{-}X0| << 1. \tag{48}$$

Since Xo is a critical point of V, we have:

$$V' = V(X) - V(Xo) = \frac{1}{2}(X - Xo)^T H(Xo)(X - Xo) \qquad |X\text{-}X0| << 1. \tag{49}$$

Notice that adding or subtracting a constant from a harmonic function yields another harmonic function , i.e. V` is also harmonic. Using eigenvalue decomposition:

$$V' = \frac{1}{2}(X - Xo)^T U^T \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & . & 0 \\ . & . & . & . \\ 0 & 0 & . & \lambda_N \end{bmatrix} U(X - Xo)$$
$$= \frac{1}{2}\xi^T \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & . & 0 \\ . & . & . & . \\ 0 & 0 & . & \lambda_N \end{bmatrix} \xi = \frac{1}{2}\sum_{i=1}^{N} \lambda_i \xi_i^2 \tag{50}$$

where U is an orthonormal matrix of eigenvectors, λ's are the eigenvalues of H(Xo), and $\xi=[\xi_1\ \xi_2\ ..\xi_N]^T = U(X\text{-}Xo)$. Since V` is harmonic, it cannot be zero on any open subset Ω; otherwise, it will be zero for all Ω, which is not the case. This can only be true if and only if all the λ_i's are nonzero. In other words, the Hessian of V at a critical point Xo is nonsingular. This makes the harmonic function V also a Morse function.

Proposition: If the G-type controller of the multi-agent system in (1) is selected as the HPF planner in (45) then every agent is guaranteed to converge to the target (T_i),

$$\lim_{t\to\infty} x_i = T_i \qquad i=1,2, ..N \tag{51}$$

Proof: Since $V_i(x_i)$ is shown to be a valid Liapunov function candidate (LFC) (Vidyasagar, 1987). i.e.

$$\begin{aligned} &V_i(x_i) = 0 \quad \text{at and only at } x_i = T_i \ \& \\ &V_i(x_i) > 0 \quad \text{elsewhere,} \end{aligned} \tag{52}$$

Their summation is also an LFC:

$$\begin{aligned} &V(X) = \sum_{i=1}^{N} V_i(x_i) \\ &V(X) = 0 \quad \text{at and only at } X=T \quad \& \\ &V(X) > 0 \quad \text{elsewehre,} \end{aligned} \tag{53}$$

where $X=[x_1\ x_2\ ...\ x_N]^t$, and $T=[T_1\ T_2\ ...T_N]^t$. The time derivative of V(X) is:

$$\begin{aligned} \dot{V}(X) &= -\sum_{i=1}^{N} \nabla V_i(x_i)^t \dot{x}_i \\ \dot{x}_i &= -\nabla V_i(x_i) \\ \dot{V}(X) &= -\sum_{i=1}^{N} \left|\nabla V_i(x_i)\right|^2 \end{aligned} \tag{54}$$

Since harmonic functions are Morse, the stable equilibrium, target points (T_i's) are the only points in the minimum invariant set of the system. By applying the LaSalle invariance principle (LaSalle, 1960) it can be easily shown that each agent will converge to its respective target.

5.2.2 Results

The ability of agents equipped with an HPF-based, G-type controller to cooperatively solve the planning problem while treating space as a scarce resource is tested. Five agents positioned opposite to each other are required to move to a specified target from a starting point selected so that a high probability of conflict scenario is established. In figure 47 the five agents utilizing a full communication graph attempt to solve the planning problem they are faced with. As can be seen, all agents reach their destination safely maintaining at all time a nonzero, minimum inter-agent distance (DM).

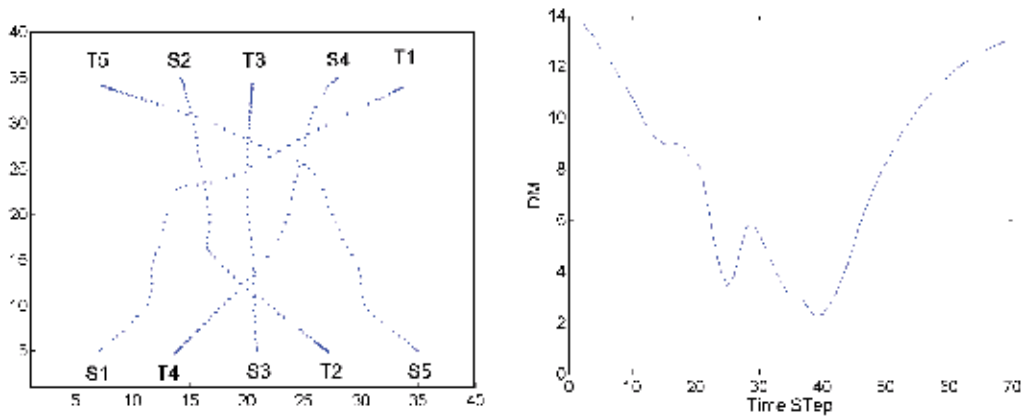


Fig. 47. Goal-oriented mode, full communication: a. trajectories, b. minimum distance

In figure 48 the agents attempt to deal with the same situation. However, this time instead of using a full communication graph, each agent only communicates its position to its closest

neighbor. Again the agents were able to safely resolve the conflict and arrive at their destination.

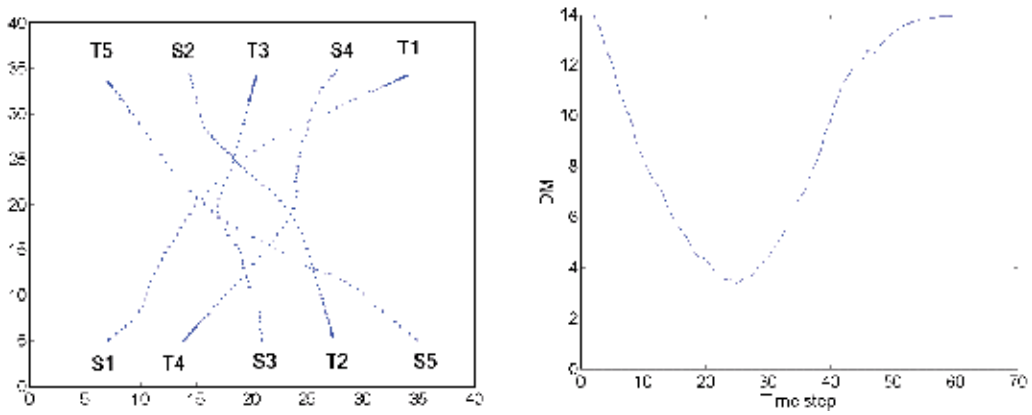


Fig. 48. Goal-oriented mode, nearest neighbor communication: a. trajectories, b. minimum distance

5.3 Multi-agent harmonic separation maintenance planner

Sometimes a group of agents are required to operate in a flexible formation mode where the members distribute themselves within a confine whose shape and motion are determined by the leader agent (figure 49). The overall trajectory of the agent is constructed by superimposing the trajectory supplied by the leader on the trajectory the agents generate to avoid collision with the components of the environment and stay within the specified geometric confines.

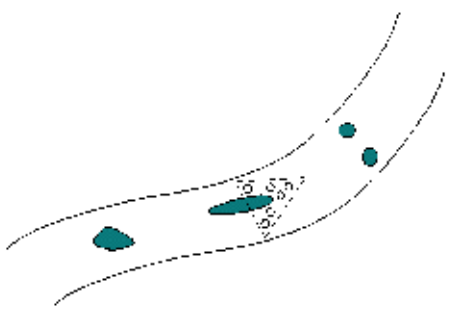


Fig. 49. Agents in separation mode

Harmonic potential fields can provide the G-Type controller for a multi-agent formation separation mode controller. Unlike the goal-oriented mode where the target point is given and the group need only to lay a conflict-free path to it, the separation mode requires the group to jointly generate the target point for each agent as well as lay a safe trajectory to that point. The HPF approach may still be used to generate a self-controller for this case. The BVP generating the potential is similar to the one in (45) with no target point having a potential preset to zero. The control action that dynamically distribute the agent in specified space may be derived from the BVP:

$$\begin{aligned} \nabla^2 \mathbf{V}(\mathbf{x}_i) &= 0 \qquad & x_i \in \Omega \\ \mathbf{V}(\mathbf{x}_i)\big|_{x_i \in S_i} &= 1 \end{aligned} \tag{55}$$

The above BVP may appear to be of little use since by the maximum principle, the solution of V in Ω is a constant. This means that the gradient field degenerates everywhere in Ω. The potential field from an environment similar to the one in figure 46 is shown in figure 50.

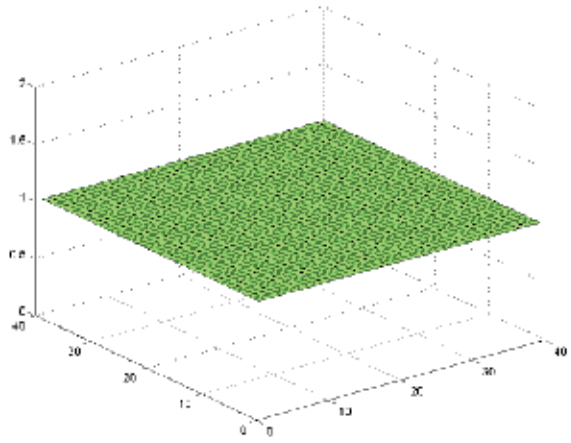


Fig. 50. Potential field degenerates in the formation case

A careful examination of the solution of (55) reveals that only the magnitude of the gradient field ($A(x_i)$) degenerates while the phase field ($Q(x_i)$) remains stable and computable. The component of the BVP in (55) that corresponds to the phase field may be derived as follows:

$$\begin{aligned} \nabla V(x_i) &= A(x_i)\mathbf{Q}(x_i)\,, \\ \nabla^2 V(x_i) &= \nabla \cdot \nabla V(x_i) \\ &= \nabla A(x_i)^T \mathbf{Q}(x_i) + A(x_i)\nabla \cdot \mathbf{Q}(x_i)\,. \end{aligned} \tag{56}$$

The gradient of the magnitude of ∇V in (4) drops to an infinitesimally small positive constant ε while A converges to unity. In this case the laplacian becomes:

$$\nabla^2 \mathbf{V}(\mathbf{x}_i) = \nabla \cdot Q(x_i)\,. \tag{57}$$

Since the potential is restricted to a constant value at S_i, Q will have no component tangent to S_i (i.e. $n \times Q(x_i)=0$, $x_i \in S_i$), where n is a unit vector normal to S_i. Therefore, the boundary value problem that may be used to generate Q is:

$$\begin{aligned} \nabla \cdot Q(x_i) &\equiv 0 & & x_i \in \Omega \\ n \times Q(x_i) = 0, \quad n^T G(x_i) &= 1. & & x_i \in S_i. \\ \dot{x}_i &= \alpha \cdot Q(x_i) \end{aligned} \tag{58}$$

where α is a positive constant. The field, Q, generated by solving the above BVP is observed to possess field lines that emanate normal to S_i and move into Ω meeting at

critical points inside the region (figure 51). Among other things these points show the tendency to form far from clutter and other agents occupying Ω. This makes it possible to utilize Q as the G-type separation control. As can be seen stable equilibrium points spontaneously form equally far from the obstacles in the environment. Figure 52 shows the separation field for another environment. The reason equilibrium points form inside Ω has to do with the fact that all the flows at the boundary are forced to be inside Ω, the continuity condition ($\nabla \cdot Q=0$) will fail at some areas in Ω. This results in stable and unstable equilibrium points being formed. A Quantitative study of these points in terms of how far from the closest object they will form is expected to be mathematically involved and will be kept for future work. However, a qualitative examination (figure 53) show that these points are comparable to maximizing the minimum distance from the obstacles.

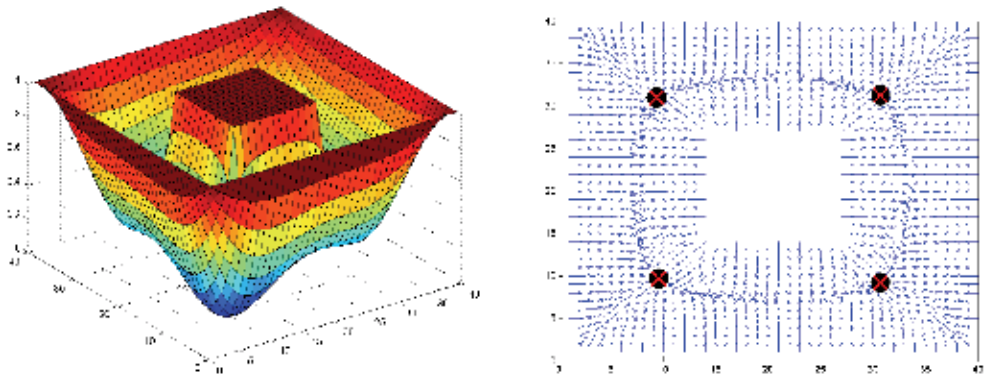


Fig. 51. Separation G-type control, harmonic phase field and gradient guidance field

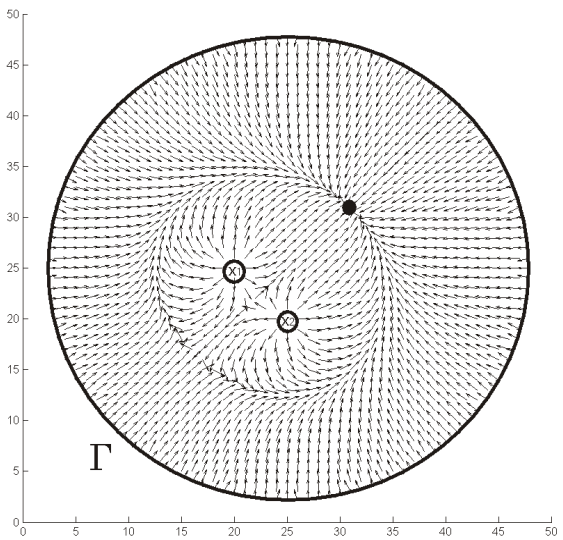


Fig. 52. Guidance field, G-type controller, separation mode

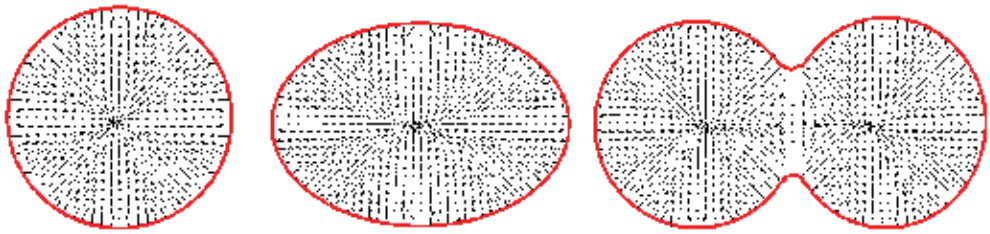


Fig. 53. Workspace geometry versus stable equilibrium points, separation mode

5.3.1 Results

The planner is tested in the separation mode for both full communication graph (figure 54) and nearest neighbor communication (figure 55). The five agents in the previous example were not provided with target points. As can be seen, in both cases, the agents managed to generate goal points that places them in a well-separated final configuration (better results were obtained in the case of the full communication graph). In addition to that, the decentralized controller was able to safely drive the agents from their initial positions to their respective target points, practically achieving a strictly increasing time - minimum separation distance profile.

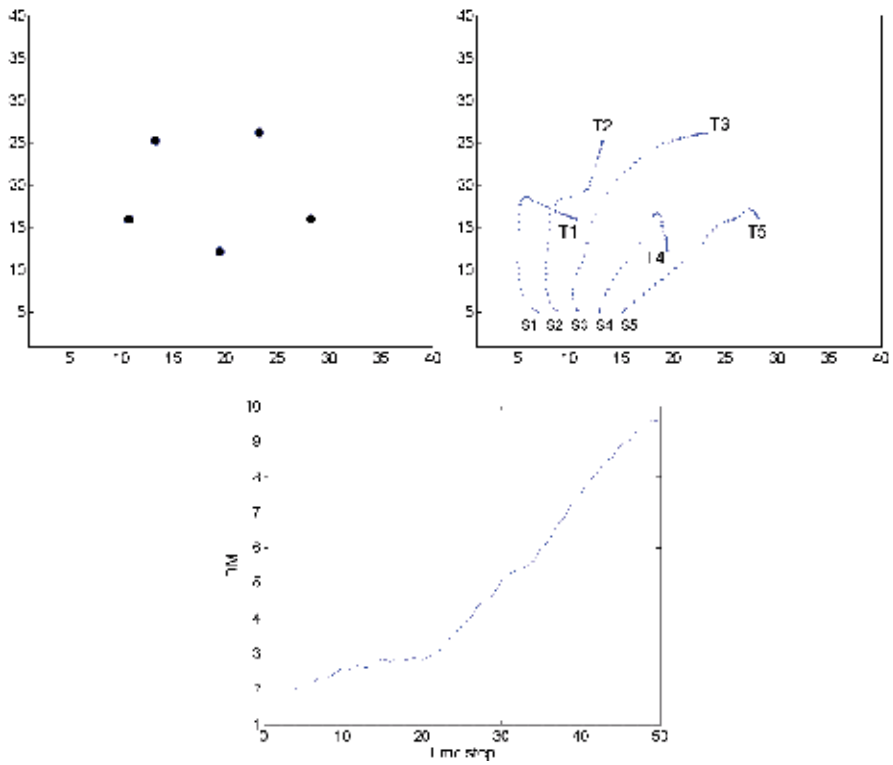


Fig. 54. Separation mode, full communication: final constellation, trajectories, minimum distance

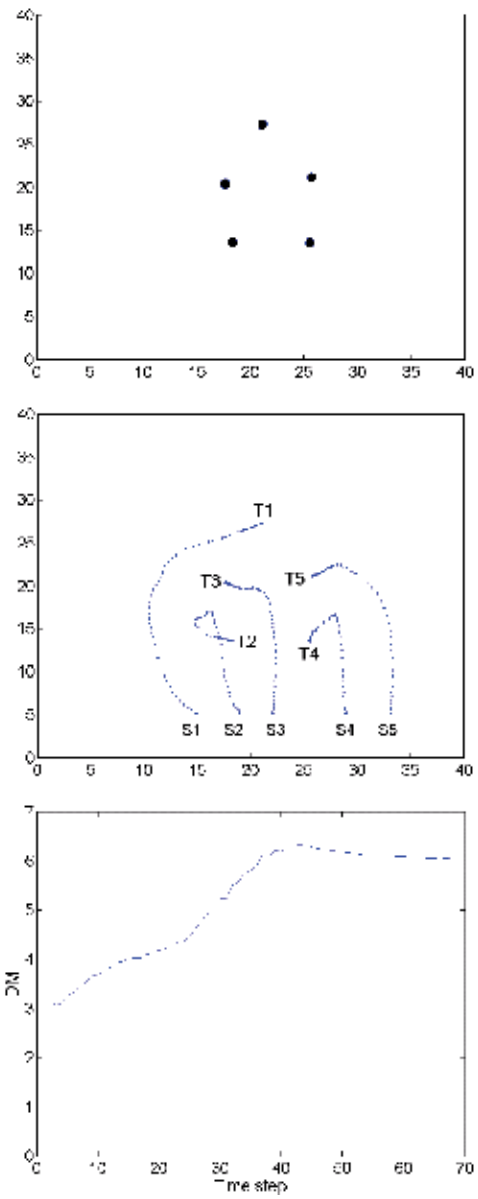


Fig. 55. Separation mode, nearest neighbour communication: final constellation, trajectories, minimum distance

In figure 56, the computational effort needed by the planner is examined in terms of the time needed to complete the steering process in the separation mode. The number of agents (Na) is varied from two to five and time needed to complete the steering process is recorded for each. Figure 16 shows the time needed to perform the steering process versus the number of agents. The time is normalized using that of the case Na=2. As can be seen, the computational time linearly grows with the number of agents. A full communication graph is used.

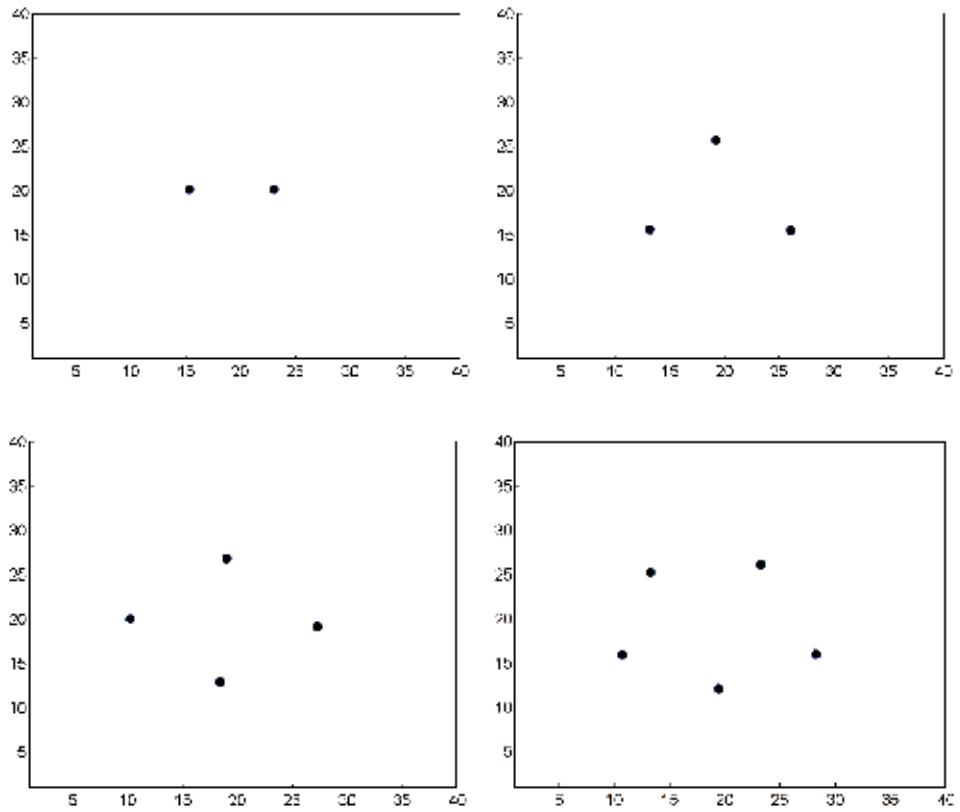


Fig. 56. Separation mode, final constellation versus number of agents

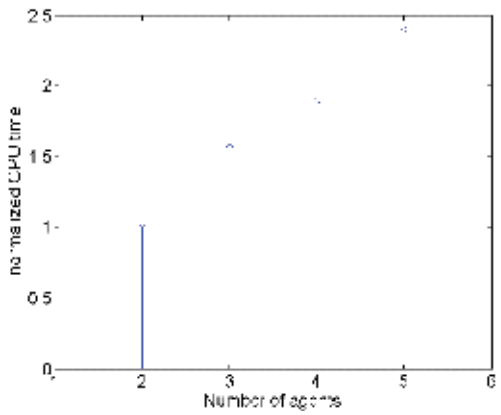


Fig. 57. Simulation time versus number of agents

In figure 58 the performance of the controller is examined in the presence of clutter for the separation mode. It is observed that all the attributes of the controller in the free space case were preserved when clutter is present. The agents distributed themselves in a final configuration that seems to maximize the minimum inter-agent distance as well as the

distance to the nearest obstacle. Also, a strictly increasing with time minimum separation distance profile is observed.

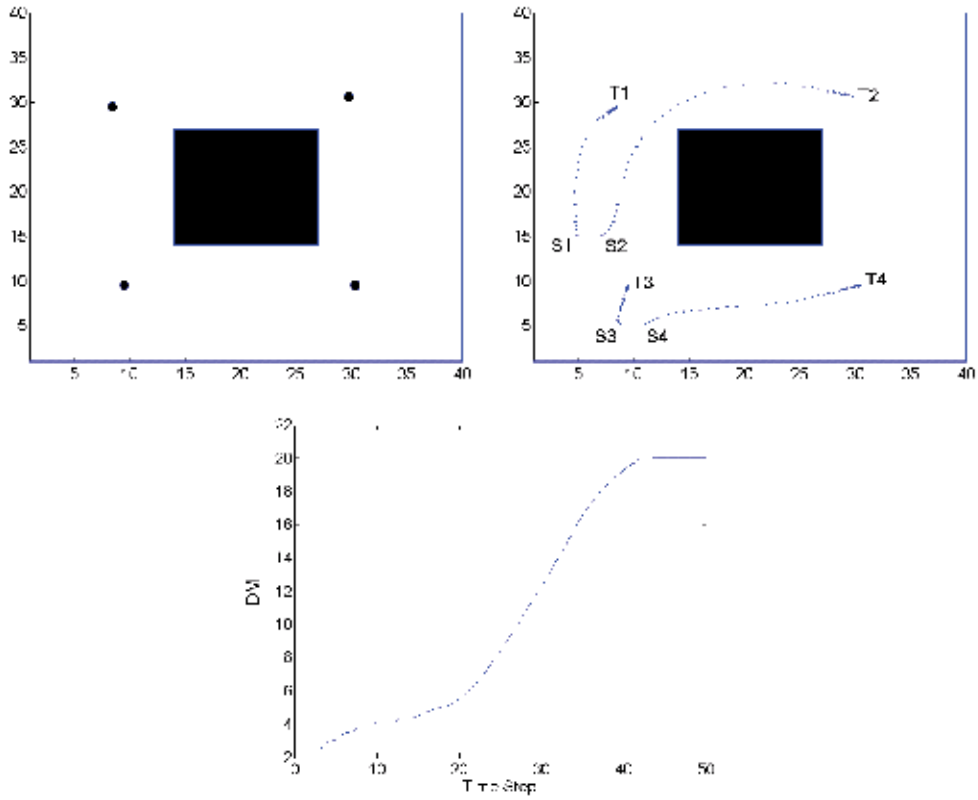


Fig. 58. Separation mode in the presence of clutter: final constellation, trajectories, minimum distance

6. Conclusions

This chapter demonstrates an important feature of harmonic potential field-based planners, that is: the social nature of such planners. This feature allows an agent steered by such a method to share, in a conflict-free manner, the same space with other agents using the same planner. Constructing a multi-agent controller in this manner has many advantages. While the system can operate in an asynchronous, decentralized mode, it can also operate in a centralized, synchronous mode that has a computational effort linear in the number of agents being controlled. The controller does exhibit an excellent ability to self-organize as well as a noncommittal planning action. This enables it to online generate the additional information needed to execute a successful action. It is also noted that the controller exhibits intelligent dispatching capabilities that enables it to redistribute the task of conflict evasion on the properly functioning agents. This property provides significant robustness in the case of sensor, or actuator failure. The controller employs an idea from the artificial life approach to behavior synthesis that is of central importance for the controller to achieve the above capabilities: i.e. the ability to project global useful activities through simple, local interacting

activities without the agents, necessarily, being aware of the generated global behavior. The artificial life G-type and P-type control modes do support such a behavior synthesis paradigm and may be considered as the backbone for building effective decentralized controllers. The HPF-based examples provided in this chapter are only a demonstration of the capabilities of this approach. The author believes that an HPF-based multi-agent controller does serve as a good basis for developing other multi-agent controllers that can effectively tackle challenging problems in many other areas such as decentralized routing in an ad hoc network (figure 59) that was suggested by (Masoud, 2008).

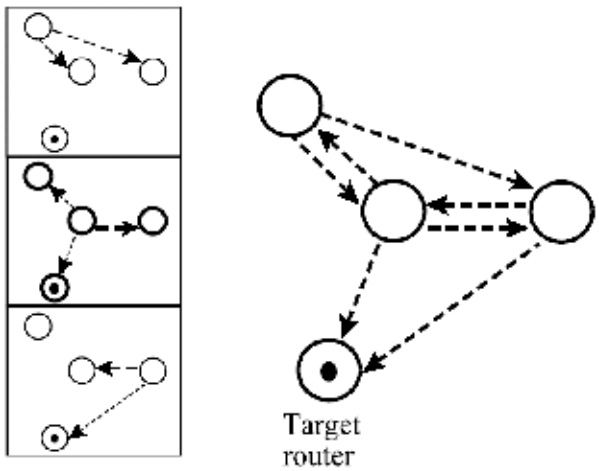


Fig. 59. Discrete harmonic potential field for decentralized routing on a graph

7. Acknowledgement

The author acknowledges the support of King Fahd University of Petroleum and minerals.

8. References

Akashita S., Kawamura S. & Hayashi K. (1990). New Navigation Function Utilizing Hydrodynamic Potential for Mobile Robots, *IEEE Int. Workshop Intelli. Motion Contr.*, Istanbul, Turkey, Aug 20-22, 1990, pp. 413-417.

Axler S., Bourdon P. & Ramey W. (1992*). Harmonic Function Theory*, Springer, 1992.

Buckley S. (1989). Fast Motion Planning for Multiple Moving Robots, 1989 *IEEE International Conference on Robotics and Automation*, May 18-21, Scottsdale, Arizona, pp. 322-326.

Bohringer K., Randall B.& MacDonald N. C. (1997). Programmable Vector Fields for Distributed Manipulation with MEMS Actuators Arrays and Vibratory Parts Feeders, in *Algorithms for Robotic Motion and Manipulation*, Ed. J. P. Laumond,M. Overmars, A. K. Petes, Wellesly MA., 1997, pp. 255-276.

Bazzan A. (1996). Evolution of Coordination as a Metaphor for Learning in Multi-agent Systems, in Distributed Artificial Intelligence Meets Machine Learning: Learning in Multi-agent Environments, *Lecture notes in Artificial Intelligence* 1221, Springer Verlag, 1996, pp. 117-136.

Burdun I. & Parfentyev O. (1999), AI Knowledge Model for Self-Organizing Conflict Prevention/Resolution in Close Free-Flight Air Sapce, *Proceedings 1999 IEEE Aerospace Conference*, March 6-13, 1999, vol. 2, pp. 409-428.

Connolly C., Weiss R. & Burns J. (1990). Path Planning Using Laplace Equation, *IEEE Int. Conf. Robotics Automat*., Cincinnati, OH, May 13-18, 1990, pp. 2102-2106.

Cohen W. (1996). Adaptive Mapping and Navigation by Teams of Simple Robots, *Robotics and Autonomous Systems*, 18, 1996, pp. 411-434.

Culberson J. (1999). Sokoban is PSPACE-Complete, *Proceedings in Informatics* 4, Fun With Algorithms, E. Lodi, L. Pagli, N. Santoro Eds., Carelton Scientific, Waterloo, 1999, pp. 65-76.

Chen Y. , Moore K. & Song Z. (2004). Diffusion boundary and zone control via mobile actuator-sensor networks (MAS-net): challenges and opportunities. INTELLIGENT COMPUTING: THEORY AND APPLICATIONS II (OR53). *SPIE Defense and Security Symposium 2004*. April 12-16, 2004, Gaylord Palms Resort and Convention Center, Orlando, FL, USA. SPIE5421-12

Decuyper J. & Keymeulen D. (1990). A Reactive Robot Navigation System Based on a Fluid Dynamics Metaphor, in *Proc. Parallel Problem Solving from Nature First Workshop*, H. Schwefel, R. Hartmanis Eds, Dortmund-Germany, Oct 1-3, 1990, pp. 356-362.

Dohner J. (1998). A Guidance and Control Algorithm for Scent Tracking Micro-Robotic Vehicle Swarms, *Sandia Report* SAND97-1354, UC-705, March 1998, Sandia National Laboratories.

De Weerdt M., Tonino H. & Wittevee C. (2001). Cooperative heuristic multi-agent planning, *Proceedings of the Thirteenth Belgium-Netherlands Artificial Intelligence Conference* (BNAIC-01), 2001, pp. 275-282.

Erdmann M. & Lozano-Perez T. (1986). On Multiple Moving Objects, 1986 *IEEE International Conference on Robotics and Automation*, April 7-10, San Francisco, California, pp. 1419-24.

Eby M. & Kelly-III W. (1999). Free Flight Separation Assurance Using Distributed Algorithms, *Proceedings 1999 IEEE Aerospace Conference*, March 6-13, 1999, vol. 2, pp.429-441.

Fujimura K. (1991). A Model of Reactive Planning for Multiple Mobile Agents, 1991 *IEEE International Conference on Robotics and Automation*, April 1991, Sacramento, California, pp. 1503-1509.

Findler N. &Lo R. (1991) , Distributed artificial intelligence approach to air traffic control, *IEE Proceedings*-D Vol. 138, No.6, Nov. 1991, pp. 515-524

Fiorini P. & Shiller Z. (1993). Motion Planning in Dynamic Environments Using the Relative Velocity Paradigm, 1993 *IEEE International Conference on Robotics and Automation*, May 2-6, Atlanta, Georgia, pp. 560-565.

Gupta S., Chakrabarty P. & Mukerjee A. (1998). Microscopic Simulation of Vehicular Traffic on Congested Roads, *Proceedings ISIRS-98*, ed. M. Vidyasagar, Bangalore, January, 1998, pp. 10-13.

Groen F., Spaan M., Kok J. & Pavlin G. (2005), Real World Multi-agent Systems: Information Sharing, Coordination and Planning, in *Logic, Language, and Computation: 6th International Tbilisi Symposium on Logic, Language, and Computation*,TbiLLC 2005 Batumi, Georgia, September 12-16, 2005. Lecture Notes In Artificial Intelligence archive, 2007, 154 - 165, ISBN:978-3-540-75143-4

Gupta R., Masoud A. & Chow M. (2006). A Network based, Delay-tolerant, Integrated Navigation System for a differential drive UGV using Harmonic Potential Field, *Proceedings of the 45th IEEE Conference on Decision & Control* Manchester Grand Hyatt Hotel San Diego, CA, USA, December 13-15, 2006, pp. 1870-1875.

Hopcroft J. E., Schwartz J. T. & Sharir M. (1984). On the Complexity of Motion Planning for Multiple Independent Objects, PSPACE-hardness of the warehouse-man's Problem, *International Journal of Robotics Research*, 3(4):76, 88, 1984.

Hasoun M., Demazeau Y. & Laugier C. (1992). Motion Control for a Car-like Robot: Potential Field and Multi-agent Approaches, 1992 *IEEE International Conference on Robotics and Automation*, July 7-10, Raleigh NC, pp. 1457-63.

Khatib O. (1985). Real-time Obstacle Avoidance for Manipulators and Mobile Robots, *The International Journal of Robotics Research*, Vol. 5, No. 1, Spring, 1985, pp. 90-98.

Kohonen T. (1997). *Self-Organizing Maps*. Berlin, Germany: Springer-Verlag, 1997.

Kuchar J. & Yang L. (2000). A Review of Conflict Detection and Resolution Modeling Methods, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, December 2000, pp. 179-189.

Krozel J. & Peters M. (2000). *Decentralized Control Techniques for Distributed Air/Ground Traffic Separation*, TR 99RTO36-03, NASA Ames Research Center Moffett Field, CA 94035, June 2000.

Korzel J., Penny S., Prete J. & Mitchell J. (2004). Comparison of Algorithms for Synthesizing Weather Avoidance Routes in Transion Airspace, *AIAA Guidance, Navigation and Confrol Conf*, Providence, RI, Aug. 2004, pp.1-16

Hao Z., Guo G. & Huang H. (2007). A particle swarm optimization algorithm with differential evolution, *Int. Conf. Machine Learning and Cybernetics*, vol. 2, pp. 1031 -1035, 2007.

LaSalle J. (1960). Some Extensions of Liapunov's Second Method, *IRE Transactions on Circuit Theory*, CT-7, No. 4, 1960, pp. 520-527.

Langton C. (1988). Artificial Life, *SFI studies in the sciences of complexity*, Addison-Wesley, 1988.

Lampinen J. & Storn R. (2004). Differential evolution, *New Optimization Techniques in Engineering,* G. Onwubolu and B. V. Babu, Eds. Germany: Springer-Verlag, pp. 123-166, 2004.

LaValle S. (2006). *Planning Algorithms*, Cambridge University Press, 2006

Milnor J. (1963). *Morse Theory*, Princeton University Press, 1963.

Mukerjee A. (1998). Neat vs. scruffy: A survey of computational models for spatial expressions.In P. Oliver and K.-P. Gapp, editors, *Representation and Processing of Spatial Expressions*. Kluwer, 1998.

Mackintosh M., Dunbar M., Lozito S., Cashion P., McGann A., Dulchinos V., Ruigrok R., Hoekstra J. & Van Gent R. (1998). Self-Separation from the Air and Ground Perspective, *2nd USA/EUROPE AIR TRAFFIC MANAGEMENT R&D SEMINAR* Orlando,1st - 4th December 1998, pp.1-16,

Malone S., Mermin J. & Neill D. (2000). Air Traffic Control, *The UMAP Journal* 21 (3) (2000) 241–256.

Marík V., Stepánková O., Krautwurmova H. & Luck M. (Eds.) (2001). Multi-Agent-Systems and Applications II", 9th ECCAI-ACAI/EASSS 2001, AEMAS 2001, HoloMAS 2001, Selected Revised Papers. *Lecture Notes in Computer Science* 2322 Springer 2002, ISBN 3-540-43377-5

Masoud A. & Masoud S. (1994). Robot Navigation Using a Pressure Generated Mechanical Stress Field, The Biharmonic Potential Approach, *The 1994 IEEE International Conference on Robotics and Automation*, May 8-13, 1994 San Diego, California, pp. 124-129.

Masoud A. (1995). Techniques in Potential-based Navigation, *Ph.D. Thesis*, Elec. & Comp. Eng. Dept., Queen's Univ., Kingston, Ontario, Canada, March 1995.

Masoud A. & Masoud S. (1997). Evolutionary Action Maps for Navigating a Robot in an Unknown, Multidimensional, Stationary Environment, Part II: Implementation and Results, *the 1997 IEEE International Conference on Robotics and Automation*, April 21-27, Albuquerque, New Mexico, USA, pp. 2090-2096.

Masoud A. & Masoud S. (1998). A Self-Organizing, Hybrid, PDE-ODE Structure for Motion Control in Informationally-deprived Situations, *The 37th IEEE Conference on Decision and Control*, Tampa Florida, Dec. 16-18, 1998, pp. 2535-2540.

Masoud A. & Masoud S. (2000a). Motion Planning in the Presence of Directional and Obstacle Avoidance Constraints Using Nonlinear Anisotropic, Harmonic Potential Fields, *the IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 24-28, 2000, pp. 2944-2951.

Masoud S. & Masoud A. (2000b). Constrained Motion Control Using Vector Potential Fields, *The IEEE Transactions on Systems, Man, and Cybernetics*, Part A: Systems and Humans. May 2000, Vol. 30, No.3, pp.251-272.

Masoud S. & Masoud A. (2002). Motion Planning in the Presence of Directional and Obstacle Avoidance Constraints Using Nonlinear, Anisotropic, Harmonic Potential Fields, *the IEEE Transactions on Systems, Man, and Cybernetics*, Part A: Systems and Humans, Vol 32, No. 6, November 2002, pp. 705-723.

Masoud A. (2003). An Informationally-Open, Organizationally-Closed Control Structure for Navigating a Robot in an Unknown, Stationary Environment, *2003 IEEE International Symposium on Intelligent Control*, October 5-8, 2003, Houston, Texas, USA, pp. 614-619.

Masoud A. (2007a). A Hybrid, PDE-ODE Control Strategy for Intercepting an Intelligent, Well-Informed Target in a Stationary, Cluttered Environment, *Applied Mathematical Sciences*, HIKARI Ltd, Vol. 1, 2007, no. 48, 2345-2371.

Masoud A. (2007b). Decentralized, Self-organizing, Potential field-based Control for Individually-motivated, Mobile Agents in a Cluttered Environment: A Vector-Harmonic Potential Field Approach, *IEEE Transactions on Systems, Man, & Cybernetics*, Part A: systems and humans, Vol. 37, No. 3, pp. 372-390, May 2007.

Masoud A. (2008). Planning with Discrete Harmonic Potential Fields, in: Xing-Jian Jing Ed., *Mobile Robots Motion Planning: New Challanges*, I-TECH, Vienna, Austeria, 2008, pp. 335-360.

Masoud A. (2009a). A Harmonic Potential Field Approach with a Probabilistic Space Descriptor for Planning in Non-divisible Environments., *2009 IEEE International Conference on Robotics and Automation*, May 12 - 17, 2009, Kobe, Japan, pp. 3774-3779.

Masoud A. (2009b), A Harmonic Potential Field Approach for Navigating a Rigid, Nonholonomic Robot in a Cluttered Environment, *2009 IEEE International Conference on Robotics and Automation*, May 12 - 17, 2009, Kobe, Japan, 3993-3999.

Masoud A. (2010). Kinodynamic motion planning: A novel type of nonlinear, passive damping forces and advantages, *IEEE Robotics and Automation Magazine*, v 17, no. 1, p 85-99, March 2010.

Ota J. & Arai T. (1993). Motion Planning of Multiple Mobile Robots Using Dynamic Groups, *1993 IEEE International Conference on Robotics and Automation*, May 2- 6, Atlanta, Georgia, pp. 28-33.

Parson D. & Canny J. (1990). A Motion Planner for Multiple Mobile Robots, 1990 *IEEE International Conference on Robotics and Automation*, May 13-18, Cincinnati, Ohio, pp. 8-13.

Parsons S. (2001). *Qualitative Methods for Reasoning Under Uncertainty,* MIT Press 2001.

Sato K. (1987). Collision avoidance in multi-dimensional space using laplace potential, in *Proc. 15th Conf. Robotics Soc. Jpn*., 1987, pp. 155–156.

Svestka P. & Overmars M. (1998). Coordinated Path Planning for Multiple Robots, *Robotics and Autonomous Systems*, 23, 1998, pp. 125-152.

Thom R. (1975).*Structural Stability and Morphogenesis*, W.A. Benjamin Inc., 1975.

Tournassoud P. (1986). A Strategy for Obstacle Avoidance and its Applications to Multi-Robot Systems, 1986, *IEEE International Conference on Robotics and Automation*, April 7-10, San Francisco, California, pp. 1224-1229.

Tomigana H. & Bavarian B. (1991). Solving the Moving Obstacle Path Planning Problem Using Embedded Variational Methods, *1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, pp. 450-455.

Tsubouchi T. & Arimoto S. (1994). Behavior of a Mobile Robot Navigated by an Iterated Forecast and Planning Scheme in the Presence of Multiple Moving Obstacles, *1994 IEEE International Conference on Robotics and Automation*, May 8-13, San Diego, California, pp. 2470-2475.

vidyasagar M. (1987). *Nonlinear System Analysis*, Prentice-Hall Inc., Englewood Cliffs, N. J. 1978

Woods W. (1983). What's Important About Knowledge Representation, *Computer*, Vol. 16, Issue 10, 1983, pp. 22-27

Warren C. (1990). Multiple Robot Path Coordination Using Artificial Potential Fields, *1990 IEEE International Conference on Robotics and Automation*, May 13-18, Cincinnati, Ohio, pp. 500-5.

Watanabe K. & Izumi K. (1999). A Survey of Robotic Control Systems Constructed by Using Evolutionary Computations, *1999 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE SMC'99, Vol. 2, pp. 740-745.

Wu S. & Chow T. (2007). Self-organizing and self-evolving neurons: a new neural network for optimization, *IEEE Trans. Neural Net*., vol. 18, no. 2, pp. 385-396, March 2007.

Ye G., Wang H. & Tanaka K. (2005). Coordinated Motion Control of Swarms with Dynamic Connectivity in Potential Flows, In *Proceedings of the 16th International Federation of Automatic Control World Congress*, Prague, Czech Republic, 2005.

Zhu Q (1981). Hidden Markov Model for Dynamic Obstacle Avoidance of Mobile Robot Navigation, *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, June 81, pp. 390-397.

Kohonen Maps Combined to Fuzzy C-means, a Two Level Clustering Approach. Application to Electricity Load Data

Khadir M. Tarek and Benabbas Farouk
Laboratoire de Gestion Electronique de Documents (LabGED)
Department of Computer Science, University Badji Mokhtar, Annaba, Algeria

1. Introduction

In a deregulated electricity market, load forecasting is nowadays of paramount importance to estimate next day load resulting in energy save and environment protection. Electricity demand is influenced (among other things) by the day of the week, the time of year and special periods and/or days such as religious and national events, all of which must be identified prior to modelling. This identification, known as day type identification, must be included in the design stages either by segmenting the data and modelling each day type separately or by including the day type as an input, which implies data classification and cluster creation.

Data classification consists in regrouping objects of a similar data set into homogenous classes. Two main types of classifications exist: supervised and unsupervised classification. Supervised classification is based on a set of objects L of known classes, called training set, with the main goal being to identify candidate objects into their belonging classes. Where, unsupervised classification consists in partitioning a set of data D into sub-sets of similar attributes called classes or clusters (Halgamuge, 2005). Unsupervised classification is termed *clustering*, and will be so in the remaining of the chapter.

For clustering means, conventional research usually employs multivariate analysis procedures. However, it was found that clustering the data directly, becomes computationally heavy using statistical method as the size of the data set increases (Jain & Dubes, 1988; Xu & Wunsch, 2005). Despite this fact, many linear approaches such as Principal Component Analysis (PCA) (Jolliffe, 2002) and K-means were and remain, extensively used for classification and clustering purposes.

Nonlinear classification and clustering approaches stand as a strong alternative in order to treat the complexity and visualisation problems issued from large multidimensional data sets. In recent years, due to their high performance in engineering, Artificial Neural Networks (ANN), more specifically Self Organising Maps (SOM), and fuzzy logic are now being used as alternate statistical tools. Combining both paradigms in a two-level approach may be profitable to reduce significantly the computational cost as shown in (Khadir et. al., 2010) where SOM and K-means were combined for time series clustering.

The idea behind the two level approach is to reduce the data dimensions using SOM finding the data quantization prototypes in the first stage. The second stage, then focuses on clustering the already obtained prototypes using, in this case, Fuzzy C-means (FCM) (Vesanto & Alhoniemi, 2000).

2. The Kohonen self-organizing map

The Kohonen self-organizing map (SOM) is an unsupervised classification method, which transforms a set of complex data to one or two dimensional vectors with a simple geometric relationships, and preserving the most important initial data metrics during the display, i.e. the close dataset of the input space will have close representations in the output space and thus will be classified in the same cluster or nearby clusters (Kohonen, 1990, Dreyfus *et al.*, 2004). The self organizing map is suitable for data survey because it has prominent visualization properties; it is also a very effective tool for visualizing and exploring multidimensional data (Himberg, 2000; Vesanto, 1999). SOM has two layers, the input and the Kohonen or output layer, Figure 1.

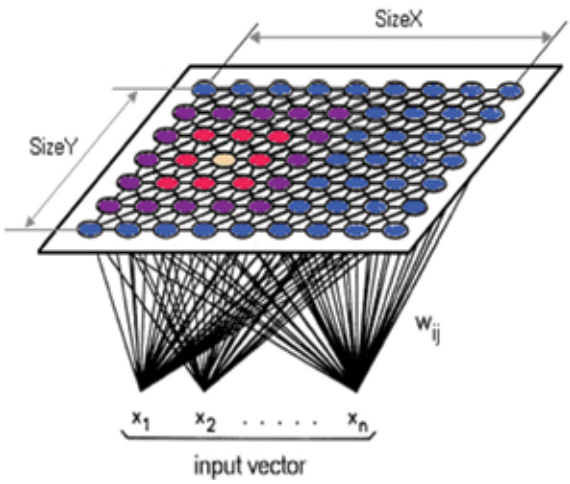


Fig. 1. Two dimensional Kohonen map

The network consists in a grid of output nodes connected to the inputs via a set of weights. When presented with the k^{th} input vector P_k ∈R1xn, the network calculates the activation of each node using P_k as:

$$a_{i,j,k} = W_{i,j} P_k \tag{1}$$

where $a_{i,j,k}$ and $W_{i,j}$ are the activation of, and weight ($\in R1 \times n$) connecting P_k to, node i, j respectively. P_k is said to be mapped onto the node with the highest activation. After several inputs have been presented, similar inputs are mapped to the same or adjacent nodes, i.e., within a small neighbourhood. A neighbourhood of size N_c around node i, j is defined as nodes $i \pm$ Nc to $j \pm N_c$. P_k for the current study is formed in two steps.
Each neuron of the topological layer is completely connected to the input layer neurons $Wi = (W_{1i} \ldots W_{ni})$, the weight vectors of these connections form the referent or prototype associated to each neuron, it has the same dimension as the input vectors. In each training

step, one sample vector **x** from the input data set is chosen and a similarity measure is calculated between it and all the weight vectors of the map. The Best-Matching Unit (BMU), is the unit whose weight vector has the greatest similarity with the input sample *P*. The similarity is usually defined by means of a distance measure; typically Euclidian distance. The use of neighbourhood concept introduces the topological constraints in the final SOM geometry.
The weights may or may be not, initialised randomly. In some cases they are initialised around the mean of the inputs as the inputs are all similar and thus restricted to a small portion of the space.
The neurons of the Kohonen map learn to recognize groups of similar input vectors. Thus, the neuron whose weight vector is closer to the input vector is then updated to be even closer. The result is that the winning neuron is more likely to win the competition next time if similar vector is presented, and less likely to win when a very different input vector is presented. The training stage stops when any of the following conditions are met: the maximum number of epochs is reached, the performance has been minimized to the goal, or maximum amount of time has been exceeded.
During training the inputs are presented one by one and the weights of the triggered node (the node to which the inputs is mapped) and nodes in its neighbourhood are updated as in equation (2).

$$W_{i,j}(m+1) = W_{i,j}(m) + \alpha(m)\left[P_k - W_{i,j}(m)\right] \tag{2}$$

Where α is the adaptation gain, with $0 < \alpha < 1$, and *m* is the iteration number. This has the effect of increasing the activation of the triggered node and its neighbours. In a single iteration all the inputs are presented and the weights adapted. After several iterations, the neighbourhood size is reduced by one and so on until zero, i.e., the triggered node only is adapted.

3. Fuzzy C-means clustering

K-means and conventional clustering techniques are referred to as hard or crisp clustering, which means that each object is assigned to only one cluster. For fuzzy clustering, this restriction is relaxed, and the object can belong to all identified clusters with a certain degree of membership (Bezdek, 1981). This is particularly useful when the boundaries among clusters are not well separated and ambiguous. FCM is one of the most popular fuzzy clustering algorithms (Szilágyi, 2009), it attempts to find the most characteristic point in each cluster, which can be considered as the "centroid" of the cluster and then, the grade of membership for each object in the clusters. Such aim is achieved by minimizing an objective function. A commonly used objective function is: membership weighted within cluster error defined as follows:

$$J_m(U,V) = \sum_{i=1}^{n}\sum_{j=1}^{c}(u_{i,j})^m \left\|x_j - v_i\right\|^2 \tag{3}$$

Where *n* is the total number of patterns in a given data set and *c* is the number of clusters; $X = \left\{x_1, x_2, ..x_n \subset \boldsymbol{R}^s\right\}$ and $V = \left\{v_1, v_2, ..v_n \subset \boldsymbol{R}^s\right\}$ are the feature data and $U = \left[u_{ij}\right]_{cxn}$ is a fuzzy

partition matrix composed of the membership grade of pattern x_j to each cluster *i*. $\|x_j - v_i\|$ is the Euclidean norm between x_j and v_i .
The weighting exponent m is called the fuzzifier which influences the clustering performances of the FCM (Cannon, 1986; Bezdek and Pal, 1988; Yu et al., 2005). The cluster centroids and the respective membership functions that solve the constrained optimization problem in (3) are given by the following equations:

$$v_i = \frac{\sum_{j=1}^{n} (u_{ij})^m x_j}{\sum_{j=1}^{n} (u_{ij})^m}, \; 1 \le i \le c, \tag{4}$$

$$u_{ij} = \left[\sum_{k=1}^{c} \left(\frac{\|x_j - v_i\|^2}{\|x_j - v_k\|^2} \right)^{1/(m-1)} \right]^{-1}, \; 1 \le i \le c, 1 \le j \le n. \tag{5}$$

Equations (4) and (5) constitute an iterative optimization procedure. The goal is to iteratively improve a sequence of sets of fuzzy clusters until no further improvement in $J_m(U,V)$ is possible.

3.1 The fuzzy C-means clustering algorithm

The FCM algorithm is executed in the following steps:
Step 1: Given a pre-selected number of cluster *c*, a chosen value of *m*, initialize memberships u_{ij} of x_j belonging to cluster *i* such that

$$\sum_{i=1}^{c} u_{ij} = 1. \tag{6}$$

Step 2: Calculate the fuzzy cluster centroid v_i for $i = 1, 2, \ldots, c$ using Eq. (4).

Step 3: Employ Eq. (5) to update the fuzzy membership u_{ij} .

Step 4: If the improvement in $J_m(U,V)$ is less than a certain threshold (ε), then halt; otherwise go to step 2.
Numerous FCM variants and other fuzzy clustering algorithms have appeared as a result of the intensive investigation on the distance measure function, the effect of weighting exponent on fuzziness control (Hoppner, 1999; Eschrich et al., 2003).

3.2 Cluster validity and validity indices in fuzzy environment

The FCM clustering algorithm has been widely used to obtain a fuzzy c-partition. This algorithm requires the user to predefine the number of cluster (*c*) and the fuzzier parameter *m*. Since it is not always possible to know these parameters in advance, different fuzzy partitions are obtained for different values of *c*. An evaluation methodology is required to validate each of the fuzzy c-partitions and, once the *c*-partitions are established, an optimal partition (or optimal number of clusters) may be considered. This quantitative evaluation is the subject of cluster validity. The cluster validity may be an indicator on partition quality.

Consequently, the cluster validity index can also be used to search for the optimal number of clusters when the number of clusters in data set or prototypes is not known in advance. Among the most used validity indices, and the one used in the remainder of the chapter, one may cite the following:
The Partition Coefficient (PC): measures the amount of overlapping between clusters. It is defined by (Bezdek, 1981) as follows:

$$PC = \frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{\ 2} \tag{7}$$

The Partition Entropy (PE): measures only the fuzziness of the Partition, similarly to the Partition Coefficient (PC) (Bezdek, 1981).

$$PE = -\frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}\ \log u_{ij} \tag{8}$$

The Partition Index (SC): It is the sum of the individual cluster validity measures normalized through division by the fuzzy cardinality of each cluster (Bensaid, *et al.*, 1996).

$$SC = \sum_{i=1}^{c} \frac{\sum_{j=1}^{N} \left(u_{ij}\right)^{m} \left\|x_j - v_i\right\|^2}{N_i \sum_{k=1}^{c} \left\|x_j - v_i\right\|^2} \tag{9}$$

The Xie and Beni's index: aims to quantify the ratio of the total variation within clusters and the separation of clusters (Xie and Beni, 1991).

$$XB = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{\ m} \left\|x_j - v_j\right\|^2}{n \min_{i \neq j} \left\|x_j - v_j\right\|^2} \tag{10}$$

Both *PC* and *PE* possess monotonic evolution tendencies with *c,* and involve only the membership values. This may have some drawbacks cited as: Their monotonous dependency on the number of clusters, their sensitivity to the fuzzier parameter, *m* and the lack of direct connection to the geometry of the data or prototypes, since they do not use the data itself. *SC* is useful when comparing different partitions having equal number of clusters. A lower value of *SC* indicates a better partition. Xie and Beni's validity function involves the membership values and the data or prototypes. The validity XB index focuses on two properties: compactness and separation.

4. A two level clustering approach

The number of prototype vectors resulting from SOM clustering may be large, especially when dealing with highly multidimensional time series applications. Only one classification level can then be revealing. A high level is interesting because it provides more detailed quality analysis and less compresses the dataset if we summarize all days by representatives

of a small class's number (Rousset, 1999). It also can be very difficult to attribute some units of the input vector to a given cluster given by the map. The problem lies in the selection of some clusters border, where a clear distinction between two clusters is impossible. A second clustering stage becomes then useful to remove ambiguity and validate the SOM results.

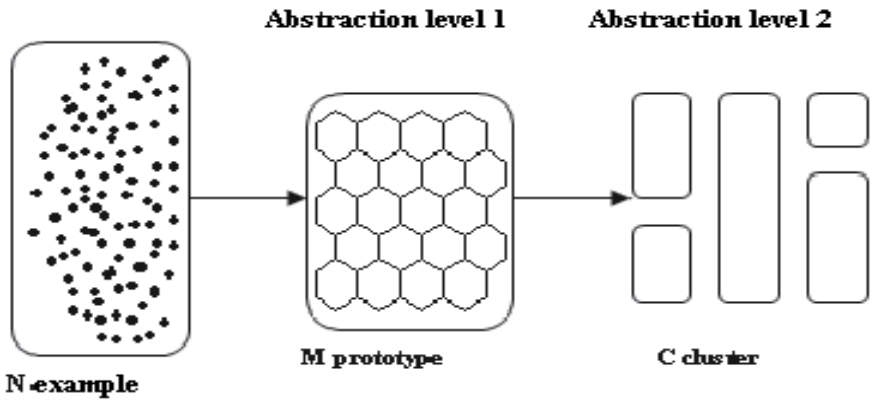


Fig. 2. First abstraction level is obtained by creating a set of prototypes vectors using the SOM. Clustering of the SOM creates the second abstraction level

The approach used in this chapter, is depicted in Fig. 2, the first abstraction level is achieved by creating a set of prototypes using SOM. These prototypes are then clustered in the second abstraction level using the fuzzy c-means clustering algorithm (Section 3). It was noticed that clustering a large multidimensional time series data using only fuzzy c-means or k-means is computationally heavier than the two-level clustering approach. Another advantage of this approach is noise reduction (Vesanto & Alhoniemi, 2000), as the prototypes are local averages of the data and therefore less sensitive to random variations than the original data.

5. Day type identification of electricity load

As explained in Section 1, it is proven that the day types or daily consumer's habits for different periods of time, such as working days, weekends, special holidays, etc affect heavily the load shape (Fay, 2004). Different prediction models may then be designed for each day type. A rigorous study of the load data is, therefore of paramount importance prior to any modeling stages.

5.1 Overview of Algerian electricity load

Electrical demand in Algeria from 01/01/2000 to 31/12/2004 is shown in Fig. 3. As can be seen there is an upward trend in the data reflecting increasing economic activity over this period, also the seasonal aspect of the time series is clearly highlighted.

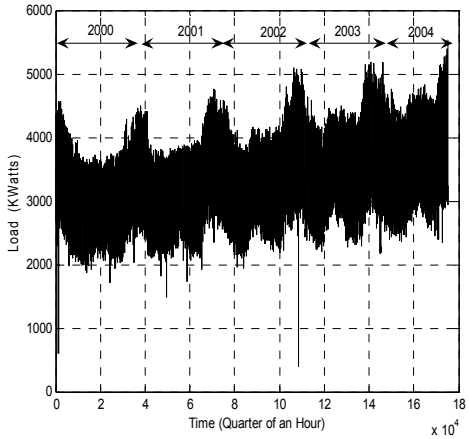


Fig. 3. Algerian electricity load 2000-2004

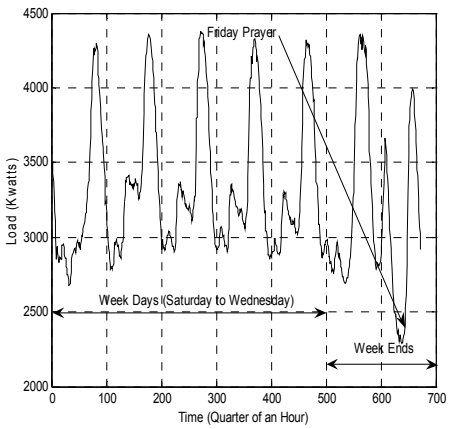


Fig. 4. Weekly load

Daily load data can be disaggregated into distinct groups (called day-types) each of which has common characteristics. As can be seen in (Fig. 4.) there is, for example, an obvious difference between the shapes of the load on a typical weekend day, such as Friday and a working day such as Saturday or Sunday due to decreased economic activity and the weekly religious prayer on Friday. Note that in Algeria the weekend was on Thursdays and Fridays in those years. Furthermore, there is a distinct difference between the shape of a typical winter day and summer day.

In addition to time, seasonal and economic activity, the regional aspect may strongly affect load characteristics. As such, the identification study has to address concerns for the load of four distinctive regions: Algiers (the capital), Oran, Hassi Messaoud and In Amenas. The two first regions are located in the north-west and north-center of the country. Their climate is Mediterranean with a complete seasonal cycle. The size of the population and the economic development are increasing significantly. The last two regions are located in the south of the country (Sahara). Their climate is very hot all year long; and the population

density is less than one inhabitant per Km² which decreases considerably any economic activity. Fig. 5 shows the Saturday 18th of March 2000, normalized load for all four regions. As it can be seen, the differences in characteristics, concerning the early, midway and late peaks are clearly highlighted.

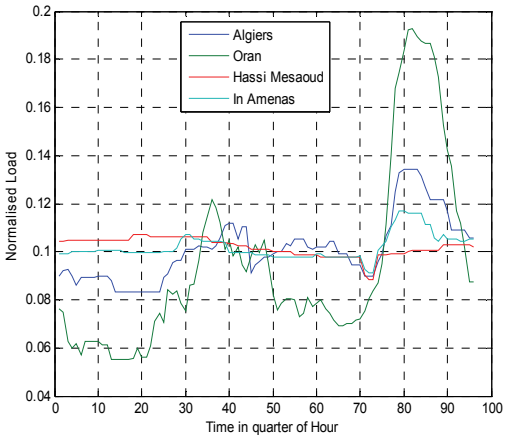


Fig. 5. Regional load differences

5.2 Day type identification using Kohonen maps

The existence of several different day types has been shown by several researchers (Bretschneider *et al.*, 1999; Hsu and yang, 1991; Muller and Petrisch, 1998). However, the level of desegregation in day-type selection is, to a large extent, subjective and dependant on the judgment of the forecaster. As pointed out by (Hubele and Cheng 1990), the application of a separate load forecasting model for different seasons (for example summer, autumn, winter and spring) has the advantage that the models do not need to incorporate seasonal information.

Further desegregation of the load by day of the week (for example Summer Sunday, Winter Sunday, Summer Monday etc.) reduces further the amount of information that the model needs to incorporate. Such approaches have been implemented successfully by (Srinivasan *et al.*, 1999) and (Mastorocostas *et al.*, 1999), to mention but a few. Where a single model is used for all the data, the day-type information is often incorporated as an additional input (two examples are (Chen *et al.*, 1992) and (Lertpalangsunti and Chan, 1998). In either case the day-types must, however, be identified. The selection of day-types can be guided by analytical techniques. The self-organising feature map or Kohonen map (Kohonen, 1990) would appear ideal for day-type identification as the number and similarity between day-types is not known *a priori*. The Kohonen map can be implemented for day-type identification in several different ways (Fay and Ringwood, 2003; Hsu and yang, 1991; Muller and Petrisch, 1998) however differences in the results are insignificant in most cases thus the algorithm used by Hsu and Yang (Hsu and yang, 1991) was chosen.

For the present trials, the full years of data from 2003 and 2004 for two characteristic regions were used. In order to investigate the regional aspect of electricity load, data from Oran and Algiers representing northern large populated cities and Hassi Messaoud and In -Amenas southern ones were chosen. The Kohonen map was trained using the following parameters,

an initial neighborhood size of *Nc=1*, adaptation gain equal to 0.003, a total number of iteration m=10 and a grid size 18*18 (324) in total.
Initially, the daily load curve is extracted from each day to give a set of load curves that have a minimum value of zero and a maximum value of one (Hsu and yang, 1991).

$$Y'(i)_k = \frac{Y(i)_k - \min Y_k}{\max Y_k - \min Y_k} \quad i = 1,..,24 \tag{11}$$

where $Y'(i)_k$ and $Y(i)_k$ are the i^{th} elements (hour) of the load curve $Y'_k \in R^{1\times 24}$, and actual load $Y_k \in R^{1\times 24}$ of day k respectively. The load curves are then normalised to give them unity length:

$$P(i)_k = \frac{Y^i(i)_k}{\left(\sum_{j=1}^{24} Y_k'^2\right)^{1/2}} \quad i = 1,..,24 \tag{12}$$

where $P(i)_k$ is the i^{th} element of P_k. The weights are initialised as:

$$W_{i,j} = \left\| \left[(\mu_p(1)),,(\mu_p(24)) \right] + 5\mu \left[(\rho_p(1)),,(\rho_p(24)) \right] \right\| \tag{13}$$

where $\mu_p(1)$ and $\rho_p(1)$ are the sample mean and standard deviation of P(i) over all k, u is a uniformly distributed random number between -0.5 to 0.5 and $W_{i,j}$ is normalised to unit length as in (Hsu and yang, 1991). Weight update is then done following equation (14) repeated below for clarity:

$$W_{i,j}(m+1) = W_{i,j}(m) + \alpha(m)\left[P_k - W_{i,j}(m) \right] \tag{14}$$

Fig. 6 shows the triggered nodes identified for the years stating from 2000 until 2004. We notice that they are located in the map at the coordinates i, j with: *i* between 0 and 17 and *j* between 10 and 20.

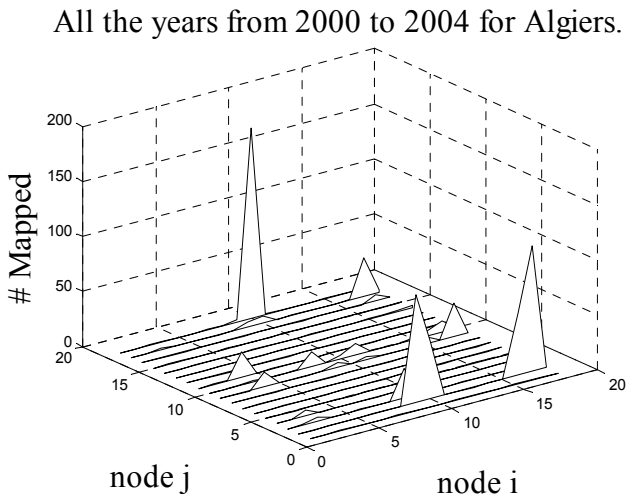


Fig. 6. Kohonen map results for Algiers load

Fig. 7 shows respectively the triggered nodes identified for 2003 until 2004 for Algiers, Oran, Hassi Messaoud and In Amenas. Clearly, according to their regional characteristics, northern cities exhibit similar behaviour.
It can be seen, Fig. 8, that week days activate roughly the same map nodes where, the weekend activate different nodes for Algiers load. This is also true for Fridays which is the weekly prayer occurring from 12 to 2:30. Thursday and Friday are the day of the weekend in Algeria.

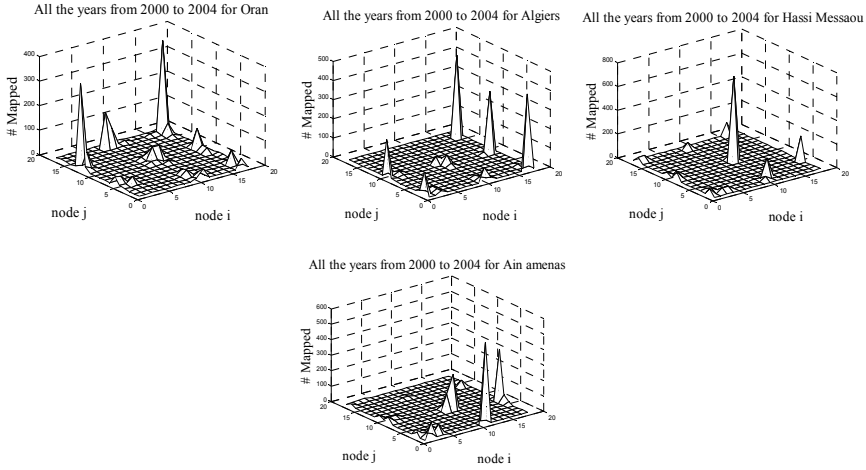


Fig. 7. Nodes triggered for All the years from 2003 to 2004 for (Algiers, Oran, Hassi Messaoud and In-Amenas)

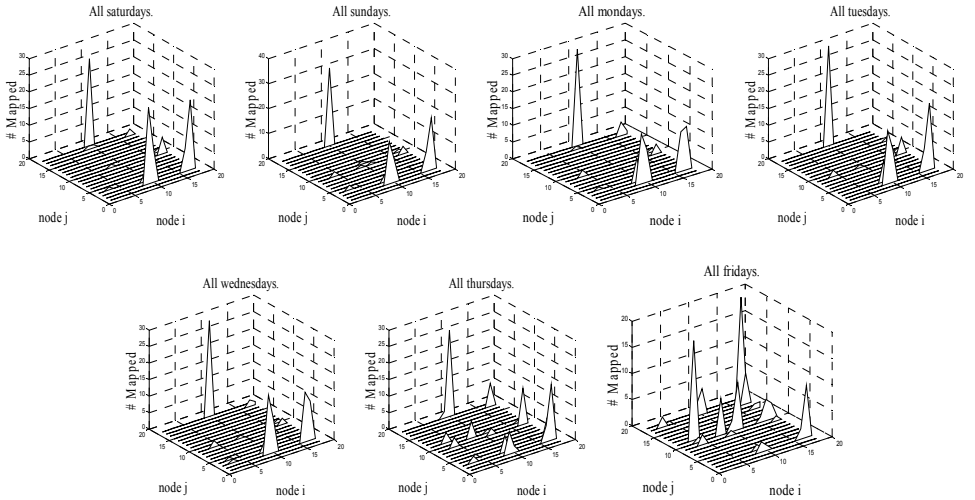


Fig. 8. Nodes triggered for working days (Saturday to Wednesday) and Week days (Thursday and Friday) loads fort he region of Algiers

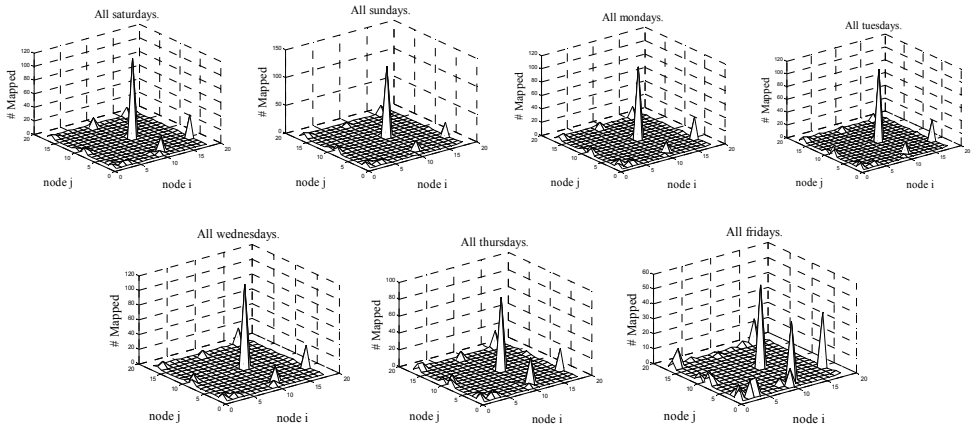


Fig. 9. Nodes triggered for working days (Saturday to Wednesday) and Week days(Thursday and Friday) loads for Hassi Messaoud

Weekdays and weekends however appear differently on the map. The nodes that are triggered from Saturday to Wednesday occupy the same parts of the grid but Thursday and Friday (weekends) loads; trigger different parts of the grid showing the difference between these day types. The figure shows the difference between these days for Algiers load, where the disparate distribution of Fridays appears clearly, and is heavily dependant on seasonal effect.

For southern regions, nodes triggered from Saturdays to Fridays for Hassi Messaoud and In-Amenas are roughly the same, showing no difference between weekdays and weekend, characteristic of southern cities, with small economic activity and population (Fig. 8).

The seasonal effect is clearly shown for northern cities, Fig. 10 for Algiers where peaks appear along the longitudinal axe of the SOM with respect to monthly (seasonal) load. Fig. 11 shows seasonal day-type identification for a southern region. The number of visually seen clusters may be identified as 4 clusters for Algiers and Oran and 3 clusters for Hassi Messaoud and two clusters for In-Amenas (Benabbas *et al.,* 2008).

In the experiments, the cluster validity indices where tested for Algiers and Hassi Messaoud prototypes in Section 5.3.

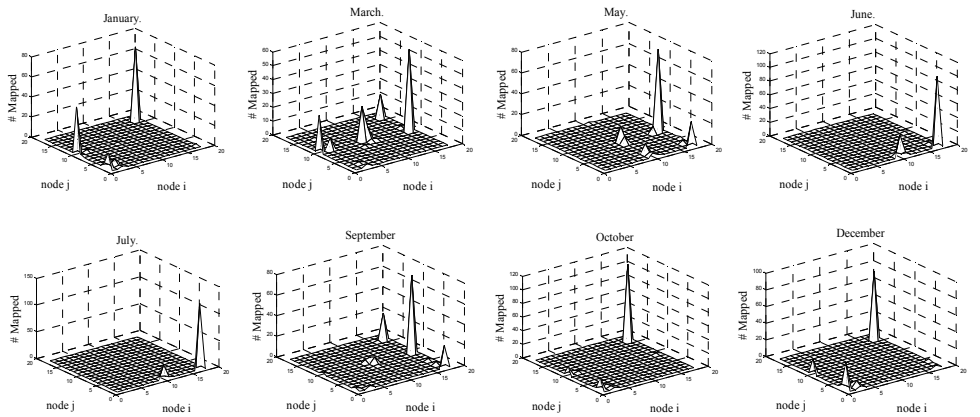


Fig. 10. Seasonal day-type identification for Algiers

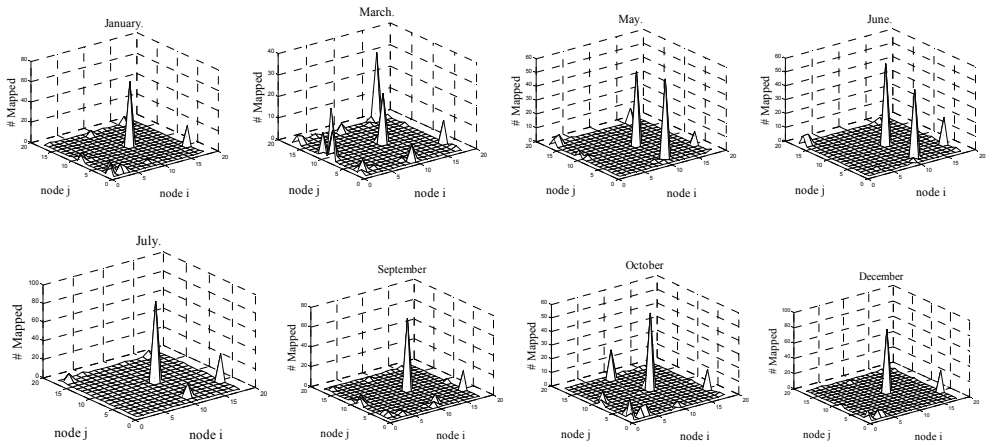


Fig. 11. Seasonal day-type identification for Hassi Messaoud

5.3 Fuzzy C-means clustering of the SOM prototypes

Table 1 lists the results of validity index for c=2,3,…..,13. For each $c \geq 2$, index values were computed for each of the four validity index considered (PC, PE, SC and XB given in equations 7, 8, 9 and 10 respectively). The optimal number of clusters is found to be, 4 for Algiers and 3 for Hassi Messaoud.

As a second stage clustering technique, FCM is used to assign the already identified items to different clusters of electricity load (Jang et al., 1997). The fuzziness index *m* was set to a value of 2; the maximum number of iteration was set to 100. These values were selected after a number of simulations and were found most suited for clustering the data obtained from the first clustering stage. The minimal amount of improvement was initially set to 1e-5. The membership's value in each group or cluster indicates the probability for a day to belong to that specific cluster. A sample of the membership's values of each of the 731 days for each of the 4 group is presented in Table 2. The group which has the highest membership's value among the 4 groups is the representative group for that day.

DataSet	*PC*	*PE*	*SC*	*XB*
Algiers	2	2	4	4
Hassi Messaoud	3	3	3	3

Table 1. Optimal cluster number according to each validity indices

For the first day corresponding to the 1st January 2003, membership's values for the 4 groups are 0.0812, 0.0353, 0.2641 and 0.6192 respectively for groups 1 to 4. The sum of theses values should be equal to 1. The representative group for the day represented by item 1, is group 4 (having the maximum membership's value of 0.6192). Similarly, all other days were analysed and grouped. It is observed that none of the 4 groups is empty. This may be due to the advantage of Fuzzy cluster analysis which allows each data set to have partial membership in all clusters. Table 3 shows the results of fuzzy clustering in three clusters for Hassi Messaoud.

Item number	C1	C2	C3	C4
1	0.0812	0.0353	0.2641	**0.6192**
2	0.1012	0.0388	0.3516	**0.5083**
3	0.0691	0.0341	0.2005	**0.6960**
4	0.2288	0.0535	**0.5690**	0.1485
5	**0.7004**	0.0675	0.1813	0.0505
……				
……				
727	0.3198	**0.4956**	0.1209	0.0635
728	**0.4382**	0.4055	0.1080	0.0481
729	**0.5729**	0.3079	0.0864	0.0326
730	**0.4382**	0.4055	0.1080	0.0481
731	**0.6134**	0.2451	0.1013	0.0401

Table 2. Obtained results for Algiers (four clusters)

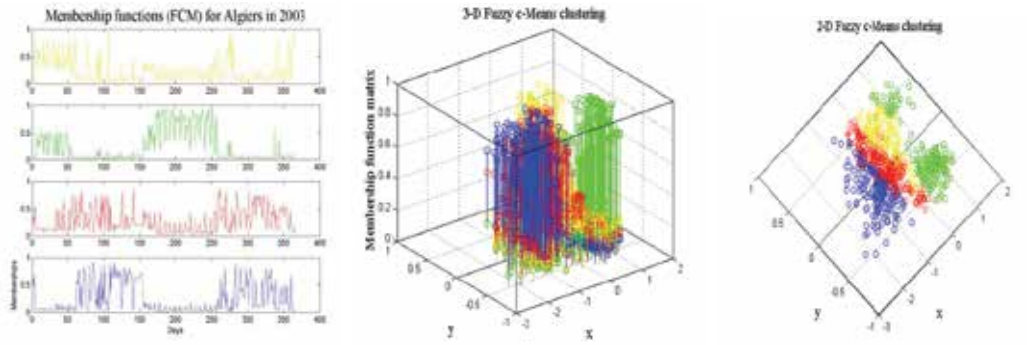
Item number	C1	C2	C3
1	0.0695	0.1092	**0.8211**
2	0.0412	0.0476	**0.9110**
3	0.2916	0.1350	**0.5733**
4	0.0319	0.0658	**0.9022**
5	0.1279	0.1038	**0.7682**
6	0.0379	0.0938	**0.8681**
……			
……			
726	0.0700	0.2241	**0.7058**
727	0.0522	**0.8123**	0.1353
728	0.0737	0.3281	**0.5980**
729	0.0438	0.1481	**0.8079**
730	0.0379	0.0938	**0.8681**
731	0.0773	**0.7364**	0.1861

Table 3. Obtained results for the southern city of Hassi Messaoud (three clusters)

In Fig. 12.a and Fig.13.a the items data for Algiers and Hassi Messaoud (northern and southern city respectively) are presented according to their membership degrees in a graphical interpretation. The horizontal axis in each subfigure represents the prototypes obtained by SOM, and the vertical axis in each subfigure represents their associated memberships.

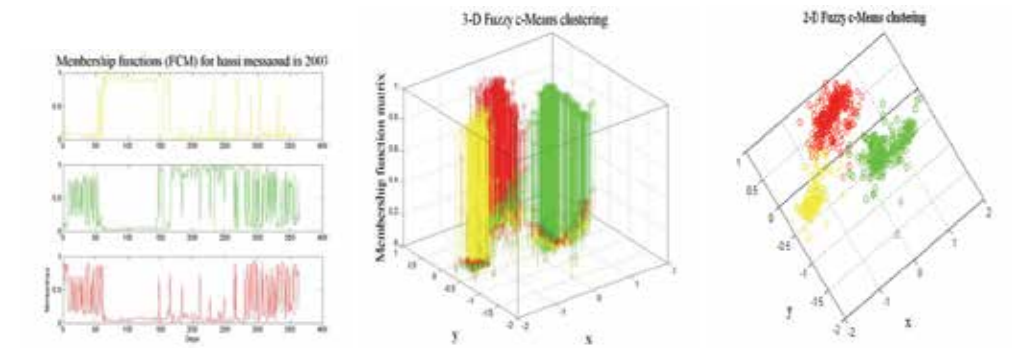
The visualization is obtained using PCA (Principal Component Analysis). The results presented in 3D and 2D are given for Algiers (Fig. 12.b) and (Fig. 12.c) and Hassi Messaoud (Fig.13.b) and (Fig. 13.c.), respectively. X and Y, being respectively, the first and the second component.

The number of identified clusters is clearly shown, in 3-D Fig. 12(b) and 13(b) respectively for Algiers and Hassi Messaoud. For a clearer interpretation, Fig. 12(c) and 13(c) shows the clusters in a 2-D representation. Fig. 12(a) and 13(a), show the membership values for each identified cluster.



(a) (b) (c)

Fig. 12. Graphical results of Fuzzy Clustering for Algiers. (a) Four cluster from top to bottom (C1 to C4). The horizontal axis in each subfigure represents the prototypes obtained by SOM and the vertical axis in each subfigure represents the membership value. (b) and (c) are the visualisation of clusters using PCA respectively in 3D and 2D dimensions. X and Y are respectively the first and second component



(a) (b) (c)

Fig. 13. Graphical results of Fuzzy Clustering for Hassi Messaoud. (a) Three cluster from top to bottom (C1 to C3). The horizontal axis in each subfigure represents the prototypes obtained by SOM and the vertical axis in each subfigure represents the membership value. (b) and (c) are the visualisation of clusters using PCA respectively in 3D and 2D dimensions. X and Y are respectively the first and second component

The day with the highest membership value in a group is the representative day for that group. The representative day for each class for Algiers and Hassi Messaoud are summarised respectively in Table 4 and Table 5.

Table 6 and Table 7 shows respectively the period and classes associated with each cluster coloured and visualized in figure 12 and 13.

Table 8 and 9 indicate for each region and for each day type the level of memberships to the clusters. For example for Algiers, The day corresponding to 01 of July 2003 belongs with a confidence degree of 90,55% to cluster C2 (Summer season) , 5,55 % to cluster C1(Winter season) 2,6 % to cluster C3 (Autumn season) and 1,3 % to cluster C4 (Spring season).

The day corresponding to the 31th January, a winter day belongs with a confidence degree of 84,14 % to cluster C1 (Winter season), 8,87 % to cluster C2 (Summer season), 5,26 % to cluster C3 (Autumn season) and 1,73 % to cluster C4 (Spring season).

Representative Day	30 December 2003	27 march 2003	21 May 2003	25 june 2003
Membership value	**0.9099**	**0.926**	**0.8741**	**0.9352**

Table 4. Representative day per class for Algiers

Representative Day	01 April 2003	27 May 2003	21 October 2003
Membership value	**0.9899**	**0.9881**	**0.9462**

Table 5. Representative day per class for Hassi Messaoud

Season	Classes	Overlapping	Period
winter	C1 (yellow)	C2,C3,C4	Dec, Jan, Fev
Summer	C2 (green)	C1,C3,C4	July et August
Autumn	C3 (red)	C1,C2,C4	September, October and November
spring	C4 (blue)	C1,C2,C3	March April May June

Table 6. Classes identified for Algiers

Season	Classes	Overlapping	Period
Autumn	C1 (yellow)	C2,C3	Feb March April
Summer	C2 (green)	C1,C3	May June July August
Spring	C3 (red)	C1,C2	Sept Oct Nov Dec, Jan

Table 7. Classes identified for Hassi Messaoud

Day	C1	C2	C3	C4
01/07/2003	0.0555	**0.9055**	0.026	0.013
31/01/2003	**0.8414**	0.0887	0.0526	0.0173
22/01/2003	**0.7928**	0.0799	0.0953	0.032
21/10/2003	0.1645	0.056	**0.5202**	0.2594
10/08/2003	0.208	**0.6406**	0.1072	0.0442
10/05/2003	0.102	0.0497	0.3682	**0.4802**

Table 8. Membership values per class per day for Algiers

Day	C1	C2	C3
31/01/2003	0.07	0.2373	**0.6927**
20/12/2003	0.0717	0.2314	**0.6969**

Table 9. Membership values per class per day for Hassi Messaoud

6. Conclusion

This chapter presents day type identification for regional Algerian electricity load using a combination of two of the most popular artificial intelligence paradigms, for instance Kohonen Maps and Fuzzy logic. The Kohonen map analysis allows a rough visual identification of the different existing classes, while the C-means clustering approach identifies existing clusters as well as providing each new day to identify, with a membership value. The choice of the optimal number of clusters is based on four fuzzy cluster validation measures (PC, PE, SC and XB). Four clusters are clearly identified for Algiers and three clusters for Hassi Messaoud, two large representative cities, respectively from northern and southern part of the country.

Fuzzy clustering methods allow objects to belong to several clusters simultaneously, with different degrees of membership. This is more natural than hard clustering, as objects on the boundaries between several classes are not forced to fully belong to one of the classes, but rather are assigned membership degrees between 0 and 1 indicating their partial memberships.

This is an important feature for day type identification to increase sensitivity, allowing a Neuro-Fuzzy forecasting approach, as existing clusters are already identified as potential models for each studied region (city). The subsequent combination between forecasting models may be given by the already determined membership functions.

7. References

Benabbas, F., Khadir, M.T., Fay, D., Boughrira, A. (2008). Kohonen Map Combined to the K-Means Algorithm for the Identification of Day Types of Algerian Electricity Load, *In Proceedings of The International Conference of Computer Information Systems and Industrial Management Applications*, Ostrava, Czech Republic.

Bensaid, A.M., Hall, L.O Bezdek, J.C., Clarke, L.P., Silbiger M.L. and Arrington J.A. (1996). Validity-guided (Re)clustering with applications to image segmentation, *IEEE Transactions on fuzzy systems*, 4(2) , 112-123.

Bezdek, J.C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum, NY.

Bezdek, J.C. and Pal, N.R. (1988). Some new indexes of cluster validity, *IEEE Transactions on systems, Man and Cybernetics* – Part B, 28, 301-315.

Bretschneider, P., Rauschenbach, T., and Wernstedt, J., (1999). Forecast using an adaptive fuzzy classification algorithm for load, *6th European Congress on Intelligent Techniques and Soft Computing*, 3, *1916-1919.*

Cannon, R.L., Dave J.V. and Bezdek J.C. (1986). Efficient implementation of the fuzzy c-means clustering algorithm. *IEEE Trans. Pattern Anal. Machine Intell*. 8, 248-255.

Chen, S.T. Yu, D.C. Moghaddamjo, A.R. (1992). Weather sensitive short-term load forecasting using non-fully connected artificial neural network, *IEEE Transactions on Power Systems,* 7 (3), 1098-1104.

Dreyfus G. , Martinez, J.M., Samuelides, M., Gordon, M.B., Badran, F., Thiria, S. and Hérault, L. (2004). *Reseaux De Neurones : Méthodologie Et Application*, Eyrolles ISBN : 2-212-11464-8, France.

Eschrich, S., Ke, J., Hall, L., and Goldof, D. (2003). Fast accurate fuzzy clustering through data reduction, *IEEE Transactions on Fuzzy syst*, 11(2), 262-270.

Fay, D., Ringwood, J.V., Condon, M., and Kelly, M. (2003). 24-hour electrical load data -a sequential or partitioned time series? *Journal of Neurocomputing,* 55(3-4), 469-498.

Fay, D. (2004). *A strategy for short-term load forecasting in Ireland*, Ph.D Thesis, Dept. of Electronic Engineering, Dublin City University, Ireland.

Halgamuge, S.K. (2005*). Classification and Clustering for Knowledge Discovery (Studies in Computational Intelligence),* Wang, L. (Eds), Springer, 3540260730, The Nederlands.

Hoppner, F., Klawonn, F., and Kruse, R. (1999). *Fuzzy Cluster Analysis : Methods for Classification, Data Analysis, and Image Recognition*, New York, Wiley.

Himberg, A. (2000). SOM Based Cluster Visualization and Its Application for False Coloring, *Proceedings of International Joint Conference on Neural Networks* (IJCNN), 3, 587-592.

Hsu, Y.Y. and Yang, C.C. (1991). Design of artificial neural networks for short-term load forecasting Part I: Self-organising feature maps for day type identification, *IEEE Proceedings-C*, 138(5), pp 407-413.

Hubele, N.F. and Cheng C.S., (1990). Identification of seasonal short-term forecasting models using statistical decision functions*, IEEE Transactions on Power Systems,* 5 (1), 40-45.

Jain, K. and Dubes, R.C. (1988). *Algorithms for clustering data*, Prentice-Hall, Englewoods Cliffs.

Jang J. S. R., Sun C. T. and Mizutani, E. (1997). Neuro-fuzzy and soft computing, p (426-427), Prentice Hall.

Jolliffe, I.T., (2002). *Principal Component Analysis, Springer*, 0387954422, New York.

Khadir M. Tarek, Khedairia Sofiane and Benabbas Farouk, *Kohonen Maps Combined to K-means in a Two Level Strategy for Time Series Clustering Application to Meteorological and Electricity Load data*, Book Title: Self-Organizing Maps, edited by George K Matsopoulos, InTech, Austria, pp 63-80, 2010.

Kohonen , T. (1990). The self-organising map, *Proceedings IEEE, 78 (9). 1464-1480.*

Lertpalangsunti N., and Chan C.W., (1998). An architectural framework for the construction of hybrid intelligent forecasting systems: application for electricity demand prediction, *Engineering Applications of Artificial Intelligence*, 11, pp 549-565.

Muller, H. Petrisch, G. (1998*). Energy and load forecasting by fuzzy neural networks*. In: H. Jurgen, H.J. Zimmermann eds., Proceedings, European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, September. Aachen: Elite foundation, 1925-1929.

Rousset, P. (1999). *Applications des algorithmes d'auto-organisation à la classification et à la prévision.* Ph.D thesis, University of Paris I, France.

Szilágyi, S., Szilágyi, L., Iclănzan, D., Dávid, L., Frigy, A. and Benyó Z. (2009). Intensity in homogeneity correction and segmentation of magnetic resonance images using a multi-stage fuzzy clustering approach, *International Journal on non-standard computing and artificial intelligence,* 19(5), 513-528.

Srinivasan D., Tan S.S., and Chang, E.K. (1999). Parallel neural network-fuzzy expert system for short-term load forecasting: system implementation and performance evaluation, *IEEE Transactions on Power Systems,* 14(3), 1100-1106.

Vesanto, J. (1999). SOM-Based Data Visualization Methods, *Intelligent Data Analysis,* 3(2), 111-126.

Vesanto J. and Alhoniemi , E. (2000). Clustering of the Self-Organizing Map, *IEEE Transactions on Neural Networks, special issue on data mining,* 11(3), 586−600.

Xie, X. L. and Beni, G. A. (1991). Validity measure for fuzzy clustering, *IEEE Transactions on Pattern analysis and Machine intelligence*, 13(8), 841-847.

Xu, k.C.R. and Haynes, L. (2001). A new data clustering and its applications .*Proceeding of SPIE-the international society for optical engineering*, 4384, 1-5.

Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3), 645-678.

Yu, J., Cheng, Q. and Huang, H. (2004). Analysis of the weighting exponent in the FCM and its application to parameter selection. *IEEE Transactions on Fuzzy systems,* 13(1), 164 - 176.

28

Fault Localization Upon Non-Supervised Neural Networks and Unknown Input Observers for Bounded Faults

Benítez-Pérez H.[1] and Ortega-Arjona J. L.[2]
[1]*Departamento de Ingeniería de Sistemas Computacionales y Automatización, Instituto de Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Apdo. Postal 20-726., Admón. No. 20, Del. A. Obregón, México D. F., CP. 01000*
[2]*Departamento de Matemáticas, Facultad de Ciencias, UNAM, Ciudad Universitaria, CP. 04510, México City, México*

1. Introduction

The task of fault diagnosis consists of determining the type, size and location of the fault as well as its time of detection. The use of knowledge-based techniques for fault localization and diagnosis allows on-line recognition of abnormal scenarios. These are based upon data treatment (Nelles, 2001), albeit these techniques require large amounts of data in order to obtain a valid representation of different scenarios. Alternatively, analytical redundancy allows a highly accurate detection of faults, based on a model of the observed system. Nevertheless, analytical redundancy requires a very accurate model of the system in order to locate a fault. Both, knowledge-based techniques and analytical redundancy, allow localization and classification of unknown scenarios as abnormal situations. The advantages of both methods depend on the type of information obtained, such as heuristic knowledge or model-based implementation. However, for abnormal situations, they have the disadvantage of not providing accurate results. In general, both methods require two important features: *(a)* the capability to determine faults, and *(b)* its sources of information.
Several different approaches attempt to combine knowledge-based techniques and analytical redundancy. (Venkatasubramanian V., et al., 2003a) (Venkatasubramanian V., et al., 2003b) (Venkatasubramanian V., et al., 2003c) present an extended overview of fault localization and diagnosis based on model- and knowledge-based techniques. In general, the combination of both methods is feasible, although presenting undesirable glitches when used simultaneously, as discussed by (Liling et al., 2002).
Several approximations have reviewed this constraint like (Su T., et al., 2008) where function approximation is pursued using a hybrid artificial neural network where data analysis becomes crutial for this purpose. In a similar manner (Zhong & Wang, 2008) presents a support vector regression where data uncertainty is studied, giving a good idea about the inherent characteristics of the data in order to by analysed. Several issues need to be addressed in order to study data analysis for system identification and representation, most of there are out of the scope of this paper. The goal of the approach followed here is to

enhance fault localization by local model identification (Billings & Wei, 2005) (Wang X., et al., 1999) (Li D., et al., 2005) and classification to unknown scenarios through a self organizing map as shown in Fig. 1. Similar approaches haven been studied by Benitez-Perez (Benítez & García, 2005) (Benítez P., et al., 2007).

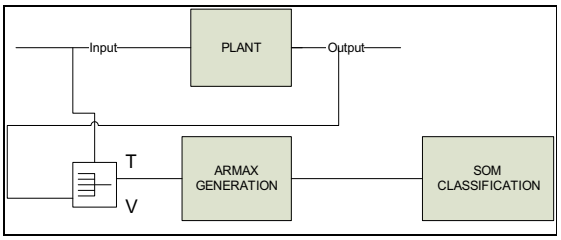


Fig. 1. Schematic approximation.

Furthermore (Benítez P. H. & Benítez P. A., 2009) present a novel strategy based upon the combination of parameter modeling (to extract several characteristics of the observed system) and the self organizing map (to classify the type of parameters extracted from the parametrization stage) in order to detect and classify online unknown behaviour.

For the purpose of this work it is necessary to define that an autonomous element is a device that is able to communicate, self-diagnose, and make decisions. The main goal of this device is to obtain as much information as possible in order to produce self-calibration and compensation. By monitoring autonomous elements of a system, several approaches can be followed for fault detection, identification and localization. Moreover, the use of local control within autonomous elements is expected to attenuate the effects of disturbances and non-linearities inherent to local model. (Lee D., et al., 2000) proposes the use of parameter estimation in order to self-tune a PID control where the response of the self-tuning procedure is restricted to a fairly linear model in order to response on time. Other approaches, like (Wang W., et al., 2002) present a strategy based on robust control, which is feasible for highly non-linear models, although having a drawback of computational cost.

On the other hand, the use of feature extraction by neural networks only presents the disadvantage of robustness in terms of inherent data uncertainties and large quantity of necessary data. Different authors have explored similar strategies previously, like (Abe S., 2001) where feature extraction using local parametric models are proposed giving valuable results, however, with the drawback of bounded system response. A strategy for fault diagnosis that integrates an ART2A network and a Kohonen neural network can be feasible since the objective is to combine both strategies in order to generate two subsystems capable to overcome glitches and redundant data representations (Abe S., 2001). The attention and oriented subsystems, following the ART2A topology and the Kohonen Neural Network, are used to perform a learning strategy from the evaluation result of the Kohonen Network. This strategy allows diagnosing faults on-line with the inherent uncertainty to SOM variation due to the plasticity-stability dilemma. A fundamental work has been introduced by (Kiviluoto, 1995) where an extended review is given about topics related to sensors patterns and stability-plasticity trade-off inherent to the ART2A network. Interesting comments are included in here on how time window data can be monitored in order to identify abnormal situations, as well as, how data should be treated in terms of normalization, time scaling and filtering and their comparison prior to declare a winner selection. Further developments are addressed (García H. & Machón G. I. 2004) focusing on

the use of a parallel ART2A network approach based on wavelet decomposition where clustering is defined on wavelet domain although it is not proposed for a dynamical system.
A complementary strategy for fault diagnosis is based on feature extraction following wavelets decomposition. Feature extraction based upon wavelets for dynamical systems presents the advantage of scales decomposition allowing several possibilities of fault detection depending on the scale of the fault. Similarly, fault detection can be easily engage if a source of information is decomposed in several fruitful components. These components can be taken as parameter vectors where several signal conditions are highlighted depending on the resolution. Moreover, these need to be combined in a fair strategy in order to classify similar behaviors. To do so, these are classified using a Self Organizing Map (SOM) where each vector is processed as consecutive input. The result of this classification would give a number of selected patterns depending on the learning rate with respect to a time window. Nevertheless, the plasticity-stability is not overcome by this technique (Benítez P. H. & Benítez P. A. 2010).
Following this review this work is focused on load systems modeling and nonlinear classifiers. The objective of this work is to define a different approach to combine knowledge-based methods and analytical redundancy for on-line classification, using non-supervised neural networks and a bank of unknown input observers (UIO's) for self-diagnosis of autonomous elements. The novelty of this approach is the classification of non-well defined fault scenarios during on-line performance of an autonomous element. In order to locate unknown scenarios, input and output data is periodically sampled from the autonomous element, using a bank of unknown input observers, which generate residual data. Two neural networks are used to process data, input output and residual, in order to determine the autonomous element's behaviour. As this is inherently time varying, the signature of its faults may also change over time. Hence, one neural network is proposed to cope with changes in the signature of autonomous element's faults, within certain boundary restrictions. The other neural network is used to classify autonomous element's behaviour, according with a number of defined scenarios. Nevertheless, for detecting time varying faults, sampling time plays a key issue.
This work is organized as follows: Section 2 describes the actual proposed approach for fault localization for an autonomous element, based on the integration of neural networks and unknown input observers. Section 3 presents a case study for testing the approach. Section 4 presents some of the most valuable results, as well as the correspondent analysis. Finally, Section 5 presents the concluding remarks.

2. Fault localization for an autonomous element

2.1 General description of the approach

The actual approach proposes an integration of two neural networks and a bank of unknown input observers for fault localization, as presented in Fig. 2.1. A non-supervised neural network samples the data from an element, processing it in order to obtain a pattern. Then, a second non-supervised neural network, using the winning weight vector (related to the winning pattern) classifies any abnormal situation.
The idea of using two consecutive neural networks is to avoid miss-classification during the presence of unknown scenarios, using a self-organizing map (SOM) and adaptive resonance theory algorithms. SOM categorizes the behaviour of the monitored element. Then, the results are evaluated by a second neural network (an ART2A) in order to avoid glitches between similar categories.

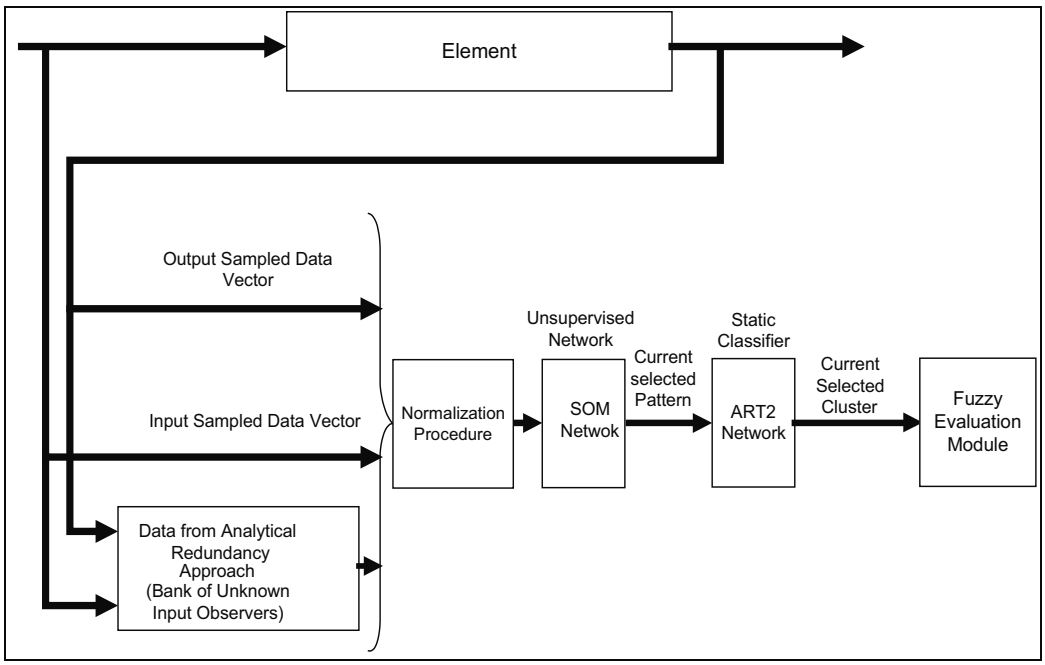


Fig. 2.1. Process diagram for Fault Localization

The data is divided into three types: input, output and residual data, this last one obtained from analytical redundancy. Data is used in two stages: an off-line stage in order to train both neural networks, and an on-line stage for testing this approach. During the first stage, a training matrix is build. Such a training matrix consists of three types of variables input, output and residual data, normalized between 0 and 1. In terms of scenarios, this matrix is divided into three areas. Each type of variable has M samples, organized as rows. The whole bunch of variables are integrated by three scenarios, organized as columns.
During training stage, each sample time window is composed of M samples directly related to a time window (Δt). The frequency of the fault has a bottom boundary, shown in eqn. 2.1.

$$\mathrm{frq}_{\mathrm{fault}} \geq \frac{4}{\Delta \mathrm{t}} \tag{2.1}$$

where frq_{fault} is the frequency of the monitored fault, and Δt is the sampled time window.
Experimentally, a quarter of Δt has been chosen as the bottom boundary since this frequency is fast enough to distinguish sampled fault information between patterns. Therefore the frequency of the fault can be larger than this quarter of Δt. Alternatively, the top limit in terms of fault sampling is unlimited, although, the approach proposed here would be useless to classify a fault much faster than a Δt sampling window. At the time that this fault localization approach produces a result, it is highly possible that the current fault can be in another stage. This top bound is still open for further research and, in principle, is based on the relation between the frequency of case study and the Δt time window. Thus during on-line stage, sampling time is reduced to one sample evaluated every time.
Both neural networks are trained in cascade as shown in Fig. 2.1. Each has its own weight matrix, which are initialized randomly.

For UIO design, formal knowledge of the element behaviour during fault scenarios is crucial, since these scenarios are defined in terms of element response during the presence of certain unknown input. Hence, it is necessary to have access to several sources of information from the monitored element. Notice that any fault localization approach relays on the dynamic characteristics of the monitored element.

2.2 Integration of non-supervised neural networks

Non-supervised Neural Networks are able to implement cluster algorithms. The main idea behind any cluster algorithm is to define centers as points within a data space. Centers serve as focal points for initial data representation. They are used for classifying non-linear behavior within non-supervised neural networks, such as SOM and ART2A. These networks present a fast response for non-linear and abnormal scenarios, although there is no guarantee for glitch presence in case of transitions. Therefore, the integration following a sequential mode allows the elimination of non-desirable transitions between scenarios due to "cluster" classification performed by SOM and pattern integration performed by ART2A.
There are various methodologies to build clusters (Billings & Wei, 2005). Proposals such as an entropy-based fuzzy clustering method defines cluster based on the entropy of each point with respect to a center. In the case of SOM and ART2A, these have the peculiarity to classify unknown scenarios in a predictable behavior. In fact, the defined clusters are the representation of several scenarios (ART2-A results) whereas those classified patterns (SOM classified patterns) are the representation of the local behaviour of the element. The integration of UIO, SOM and ART2-A allows several advantages, such as availability of measured states and the capability to classify abnormal situation, avoiding undesired glitches during on-line performance.
During the off-line stage, SOM is trained using fault and fault-free scenarios with certain frequency, using different parameters, which are tuned in order to produce a valid and unique response. An important assumption, which impacts on the structure of SOM, is the use of a rectangular grid for data classification. This has been chosen due to its regularity when comparing between patterns, even in case of unknown faults. The regular grid allows a distribution of winner patterns. However, when a scenario is classified between the winner pattern and other devious patterns, miss-classification is present. This sort of case is defined as "glitch", and it is related as a transition from one scenario to another. There are various ways to avoid this behaviour like a better training procedure, or defining winner patterns during transitions. However, glitches are not completely and certainly classified by SOM. Hence, glitches are classified using an ART2A. This network is trained to identify the response of SOM during the evaluation of one scenario with one particular pattern. This means that one particular pattern (from ART2A network) represents those patterns from SOM related to the same scenario.
The objective of this cascade is to eliminate miss-classification of time variant faults and transitions between fault-free scenarios. This approach relays on certain boundary with respect to the similitude between patterns from SOM and those from ART2A.

2.3 The fuzzy evaluation module

After defining the use of two neural networks as an approach to classify unknown scenarios, a heuristic measure is required as a final step to determine how a particular scenario has been degraded. This measure, known as confidence value, is generated by a fuzzy logic

module. This module evaluates the winning weight vector related to the classified pattern from ART2A, in order to produce a percentage representation of current behaviour.
The confidence value classifies the behaviour of peripheral element under the presence of a fault. It shows the degradation of the element with respect to the output, input and residuals. The procedure by which the fuzzy logic acquires knowledge is a key issue. Different methodologies can be followed. The confidence value has a continuous range from zero (catastrophic situation) to one (fault-free scenario) (Fig. 2.2).

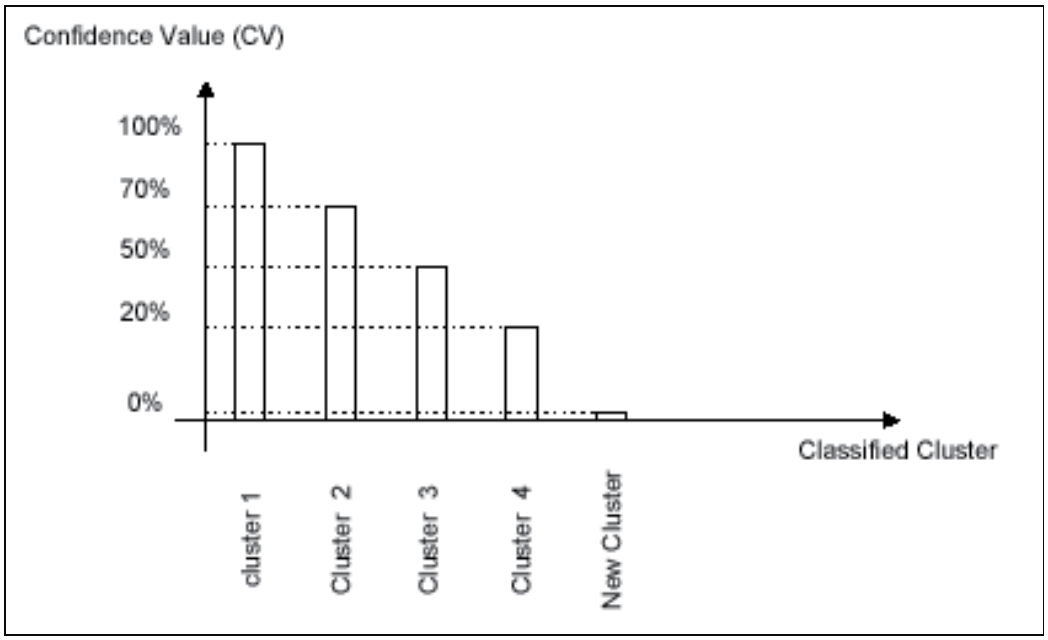


Fig. 2.2. Confidence Value Representation

2.4 Evaluation of the approach

The evaluation of this approach is carried out using two scenarios: the first scenario is composed of four similar signals with different frequencies (Fig. 2.3).
These four signals have different frequencies: 0.005 Hz (continuous line), 0.01 Hz (dotted line), 0.06 Hz (dash-dotted line), and 0.1 Hz (dashed line). This scenario has a time window of 1000 seconds. The evaluation of SOM+ART2A approach is performed every sample during this time window. First, a learning stage is accomplished by training both neural networks, using this scenario during 100 seconds (Fig. 2.3). During this learning stage, the parameters η and ρ are 0.02 and 0.021, respectively. These two parameters, η and ρ, correspond to SOM and ART2A networks respectively. During the next stage (classification stage) the SOM+ART2A approach is tested using the rest of the time window. In this case, the parameters η and ρ are changed, resulting in different numbers of patterns for the same evaluated scenario.
In the classification stage, these patterns are considered as *fail* patterns (as extra patterns) because SOM and ART2A have failed to classify them as similar to the originally recognized patterns, as shown in Table 2.1.

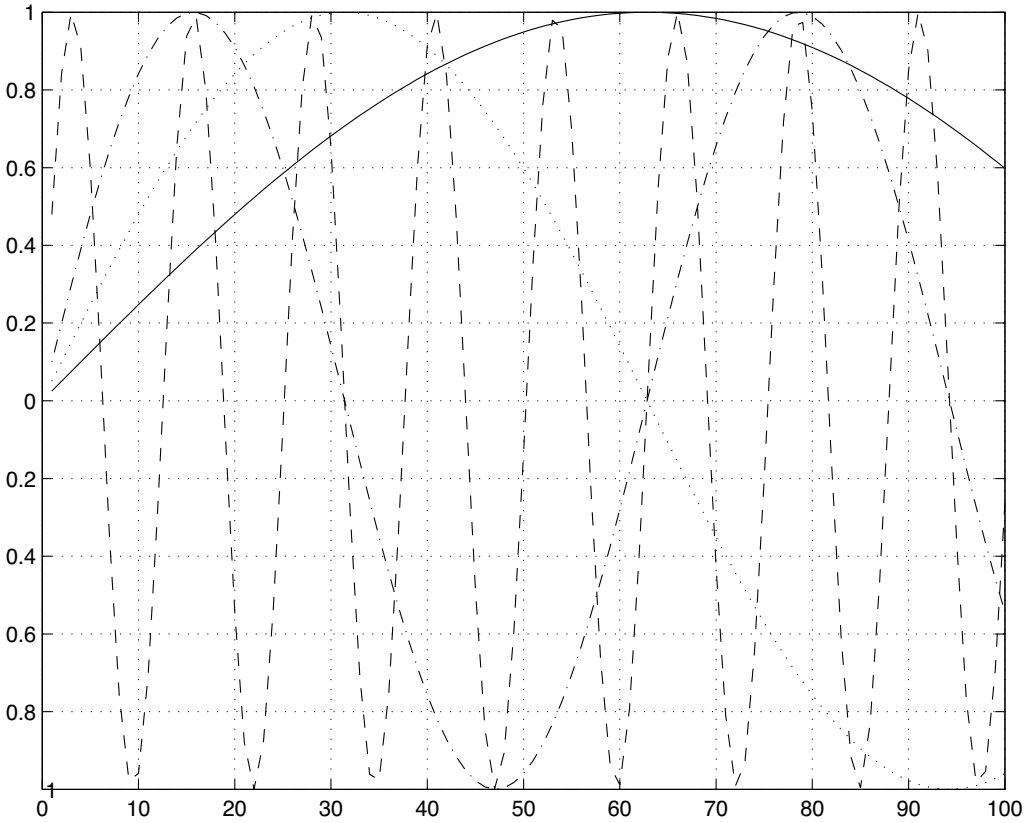


Fig. 2.3. First scenario used to evaluate the approach (a section of first 100 seconds)

		SOM		ART2A	
η (Parameter related to SOM)	ρ (Parameter related to ART2A)	New Patterns	Number of Fail Patterns	New Patterns	Number of Fail Patterns
0.011	0.011	424	-	135	-
0.015	0.015	424	-	135	-
0.02	0.02	432	-	135	-
0.07	0.07	450	378	135	369
0.1	0.1	480	477	140	463
0.12	0.12	463	480	142	463
0.15	0.15	450	378	140	380
0.18	0.18	455	380	141	385

Table 2.1. Evaluation using the First Scenario

These results suggest that the increment on both parameters permit the increment of *fail* patterns. However, the number of patterns from both neural networks does not suffer a substantial increase. The meaning of this *failure* is that some patterns are miss-classified within different scenarios.

In the second scenario, the approach is evaluated keeping both weight matrices. In this case, signals are conformed by the element's response during different situations, such as transitions from different operating points. The case study, presented in Section 4, is used to generate these signals. Fig. 2.4 shows the initial 1000 seconds of the second scenario. The continuous line is referred to the output temperature, the dashed-dotted line is the response of pressure, and the dotted line is the residuals.

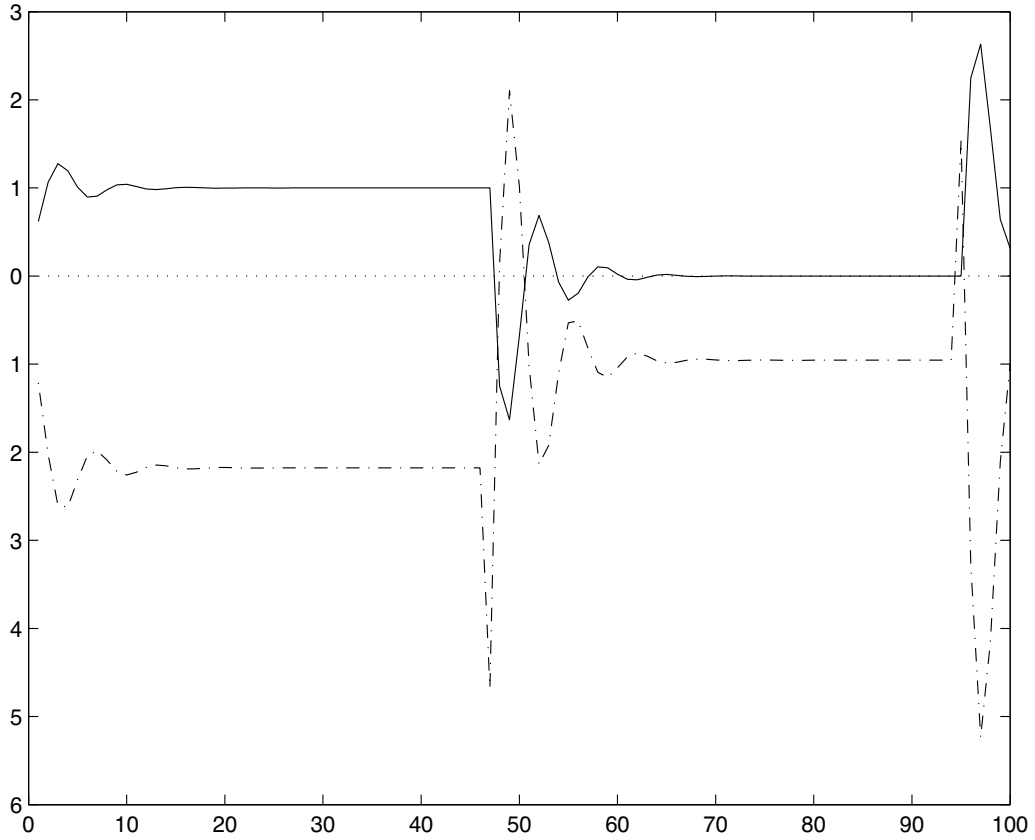


Fig. 2.4. Second scenario used to evaluate the approach

The response of this evaluation is shown in Table 2.2. During this scenario the number of new patterns is increased by SOM. Nevertheless, this behaviour is not presented in ART2A. Thus, this result confirms one of the goals of this paper, which is defining a strategy capable to cope with unknown scenarios without further appearance of new patterns. However, the number of *fails* patterns considerably increases.

For this scenario, the best η and ρ for unknown scenarios are 0.015 and 0.014 respectively. In order to confirm this result, a validation measure (Wang X., et al., 1999) (Li D., et al., 2005) has been performed, obtaining a topographic error calculated as follows (eqn. 2.2).

			SOM		ART2A	
Number of Scenarios	η (Parameter related to SOM)	ρ (Parameter related to ART2A)	New Patterns	Number of Fail Patterns	New Patterns	Number of Fail Patterns
1	0.011	0.011	457	-	135	-
2	0.015	0.014	465	451	135	462
3	0.02	0.02	465	451	135	462
4	0.07	0.07	617	472	135	479
5	0.1	0.1	658	470	139	483
6	0.12	0.12	658	470	139	483
7	0.15	0.15	618	472	138	479

Table 2.2. Evaluation using the Second Scenario

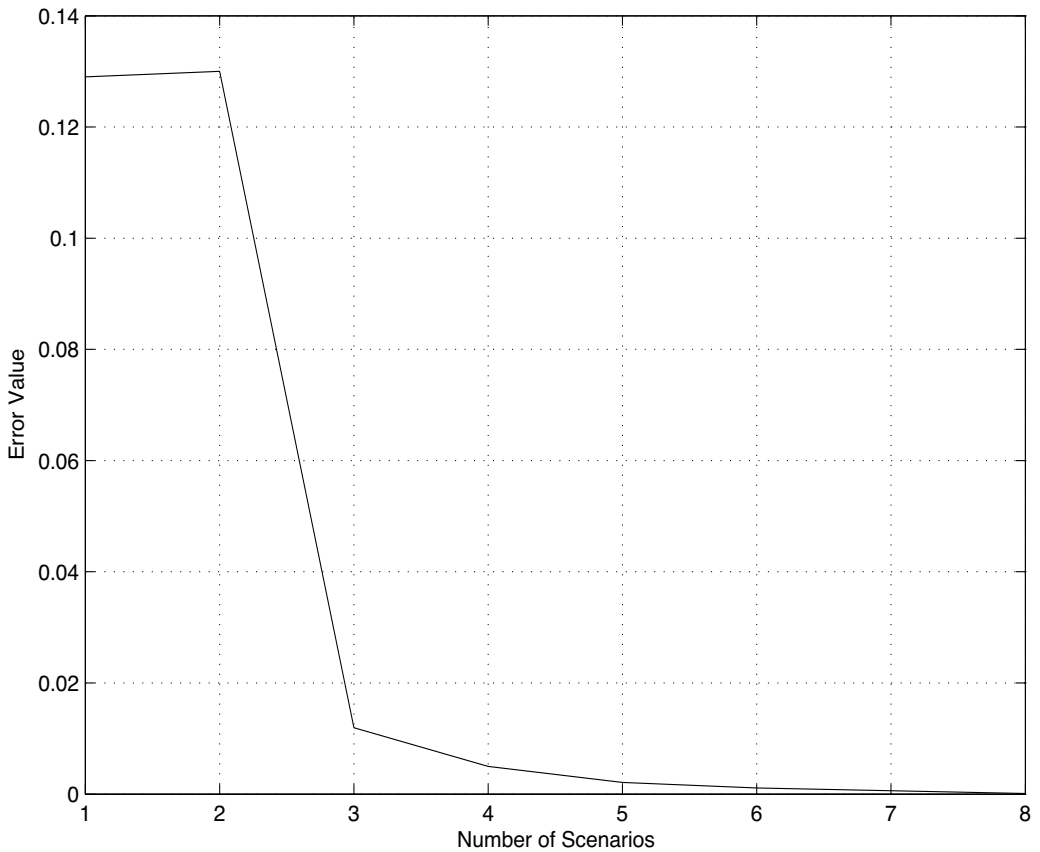


Fig. 2.5. Error Measure Performance with respect to η and ρ numbers

$$e_t = \frac{1}{N}\sum_{k=1}^{N} u(x_k) \tag{2.2}$$

where N is the number of samples, x_k is the *kth* sample of the data set, and $u(x_k)$ is 1 if the first and second best matching patterns are not adjacent units, otherwise zero. The error is evaluated with respect to classified patterns from SOM. This error shows how separate are classified patterns between each other (Fig. 2.5). Every scenario presented has a very low error performance, such as scenarios 6 and 7, where error is neglected. However, the number of patterns during these scenarios considerably increases. This is an undesirable response for on-line performance, due to the increase of time consumption during the classification stage. Alternatively, second scenario has the largest error, but the number of patterns has not presented the previously referred increase. The conditions presented in second scenario for η and ρ are preferable for classification stage, rather than any other respective value.

3. Case-study

In order to validate the present fault localization approach, this section introduces an example related to a basic implementation of the autonomous element (Fig. 3.1). This case study is based on a pressure sensor composed of three similar transducers, which have been linearised to a nominal value. The dynamic model is presented in eqn. 3.1. It consists of a bank of UIO's, an Intelligent Fault Localization Module, a local control law, and a Fuzzy Evaluation Module.

$$\begin{aligned} A &= \begin{bmatrix} 1.1 & 0.0 \\ 0.01 & 2.1 \end{bmatrix} \\ B &= \begin{bmatrix} 1.8 & -2.1 \\ 0.9 & 0.86 \end{bmatrix} \\ C &= \begin{bmatrix} 0 & 1 \end{bmatrix} \end{aligned} \tag{3.1}$$

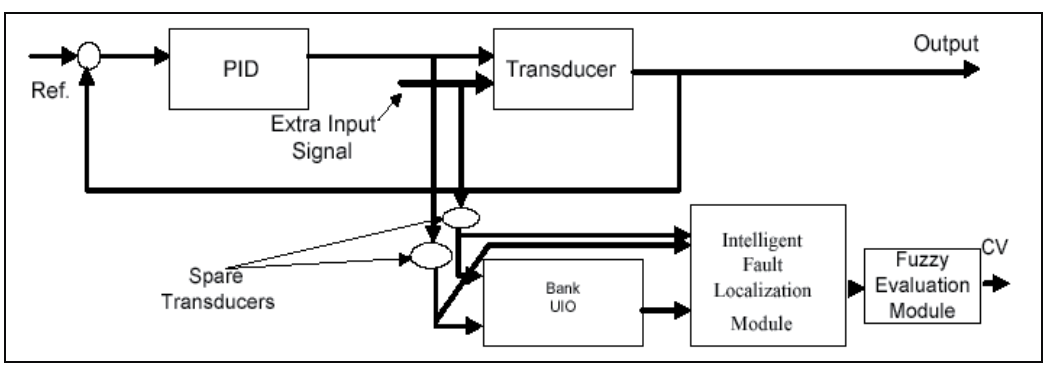


Fig. 3.1. Pressure Sensor divided in Three Modules

The input vector is composed of data from pressure demand and temperature. The output signal presents the delivery pressure. Based upon these available measures, the feedback relation is proposed in order to attenuate some disturbances. The proposed control is based upon the PI architecture (Su T., et al., 2008). The dynamics of spare transducers are not modeled. Two additive faults are considered. Both injected faults are related to a backlash, variable time delays, and a dead zone (Table 3.1).

Fault I	Backlash=0.01, Dead Zone= (-0.01, 0.01), Time Delay = 0.001
Fault II	Backlash=0.09, Dead zone= (-0.051, 0.032), Time Delay = 0.012

Table 3.1. Fault Scenarios

The presence of the faults is established in two injection points at the output of case study. These are present during specific times in order to demonstrate the proposed approach. Hence, it is necessary to implement two UIO's, sensible to each fault. Both observers conform their matrices as follows (Eqn 3.2).

$$\text{First Observer}\quad H=\begin{bmatrix}0\\1\end{bmatrix},\ F=\begin{bmatrix}1.1 & -0.1\\ 0.0 & 20.0\end{bmatrix},\ K=\begin{bmatrix}0.0001\\0.0001\end{bmatrix},\ G=\begin{bmatrix}1.8 & -2.1\\ 0.0 & 0.0\end{bmatrix}$$

$$\text{Second Observer}\quad H=\begin{bmatrix}0\\1\end{bmatrix},\ F=\begin{bmatrix}1.21 & -0.1\\ 0.0 & 20.0\end{bmatrix},\ K=\begin{bmatrix}0.0001\\0.0001\end{bmatrix},\ G=\begin{bmatrix}1.91 & -2.1\\ 0.0 & 0.0\end{bmatrix} \tag{3.2}$$

Each observer responds to a particular fault. Furthermore, fault scenarios not considered for both observers are classified as different patterns by the fault localization module. In the case of glitches and transitions, SOM classifies this behaviour as "weak" patterns, meaning patterns that belong to a certain cluster in a distance manner. If this behaviour keeps its presence, a new cluster is declared. Current values of local PID control law are k_1=*0.91*, k_2=*0.05* with respect to following equation (3.3).

$$pid = k_1 * e + k_2 * \int_{t=0}^{t=0.1} edt \tag{3.3}$$

where *e* corresponds to current error, and *pid* current control output.
The characteristics for both neural networks are selected as shown in Table 3.2. Specifically, the sampling window is equal to 100 samples. Therefore detectable fault have a minimum frequency equal to 100 Hz.

SOM	Size of Sampling data (M samples)	100 samples
	Size of Initial Output vector	4 data
	Input Vector	4 data
	Initial Population of Neurons	76 neurons
	Learning Value	0.015
ART2A	Input Vector	4 data
	Size of Initial Output vector	4 data
	Initial Number of Neurons	100
	Vigilance Parameters	0.014
	Learning value	0.02

Table 3.2. Technical Characteristics of Neural Networks

4. Results and analysis

This section presents the results related to fault and fault-free scenarios. Three different scenarios are considered, two known scenarios (Fault and Fault-Free) and one unknown scenario (Unknown Fault). For the fault-free scenario both neural networks and UIO have been already trained and designed. The element response is presented in Table 4.1, where time delay gives an approximation of how long it takes to obtain a trustable response.

Name of Scenario	Number of Selected Patterns from ART2A (New Patterns)	Response Time Delay
Known Fault-Free Scenario	5	Immediate Response

Table 4.1. Fault Free Scenario

For the case of second known scenario, where a fault is present, the selected patterns and time delay response are shown in Table 4.2

Name of Scenario	Number of Selected Patterns from ART2A (New Patterns)	Response Time Delay
Known Fault Scenario	7	100 seconds

Table 4.2. Known Fault Scenario

Alternatively, an unknown scenario is used for fault localization procedure. This scenario consists of saturation at the output of case study. Therefore it is expected an increment in the number of patterns and time delay. Table 4.3 shows this behaviour, taking into account starting time and detection time.

Name of Scenario	Number of Selected Patterns from ART2A (New Patterns)	Response Time Delay
Unknown Fault Scenario	60	200 seconds

Table 4.3. Unknown Fault Scenario

The graphical representations of these results are presented in Fig. 4.1 for fault-free scenario. Four different graphs are shown: current input, its respective output, the injected fault according to the decision making module, and the number of patterns selected as result of this evaluation. This fault-free scenario has a time variance of *sin(0.5*t)*. This time variance behaviour is depicted as current output of case study. The selected patterns are presented in a consecutive manner with respect to the horizontal axis. Although, the number of patterns increased to 70 (vertical axis), those selected are no more than 10. The first 20 patterns have been selected as part of the setting of both neural networks. From the final 50 patterns, two are predominant. These are pattern number 70 and pattern number 45. Both patterns are the representation of this fault-free scenario with a limited time variance. The number of patterns is related with the final position within the weight matrix from ART2A. An important issue with respect to the number of patterns is the very low number of *fail* patterns.

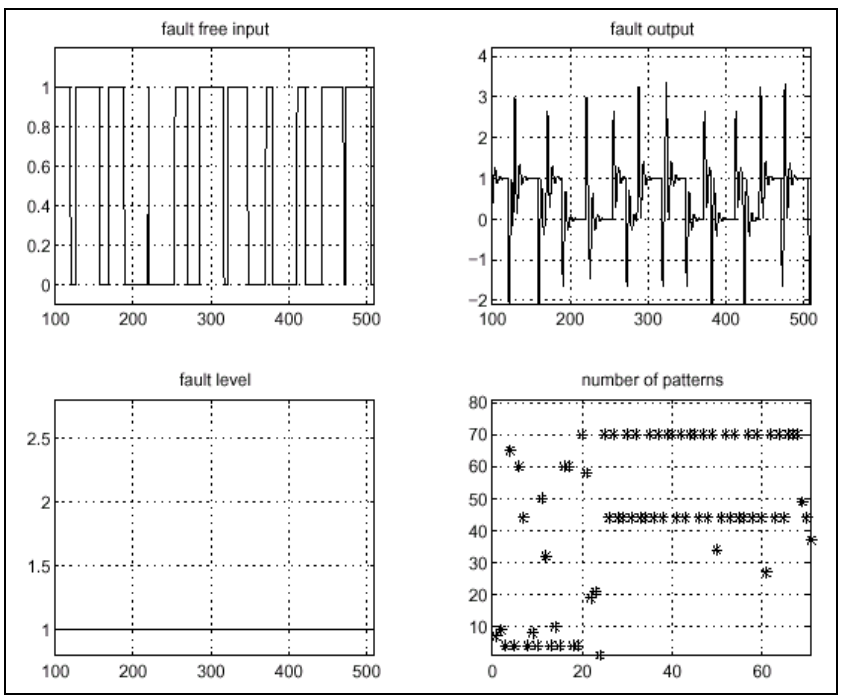


Fig. 4.1 Fault-Free Scenario with Time Variance Behaviour

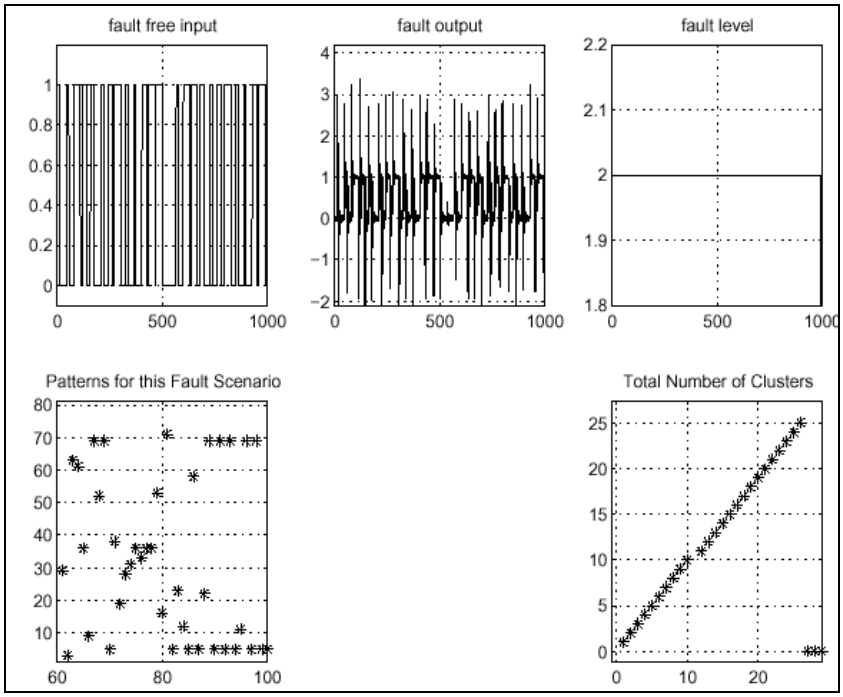


Fig. 4.2. Fault Scenario without Residual Evaluation

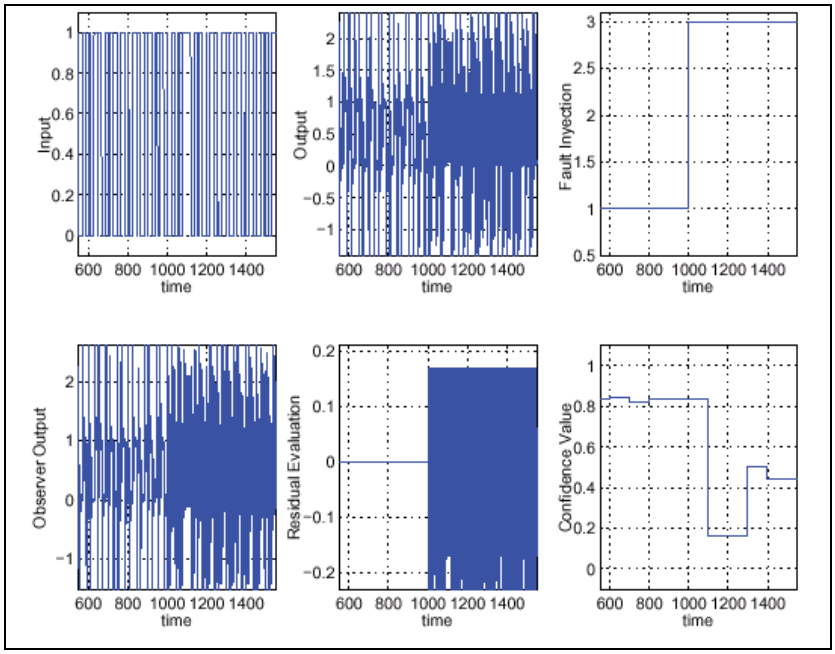


Fig. 4.3. Fault and Fault-Free Scenarios with Residual Evaluation and Time Invariant Behaviour

In the case of a fault scenario (Fig. 4.2), the response of the element presents a small perturbation due to an increment of time delay (Table 3.1, Fault II). The number of patterns increases to 25. There is no predominant pattern during this test. However, some patterns have been already selected during fault-free scenario.

The fault scenario takes into account a time variance of 0.12s as well as confidence value responses (Fig. 4.3). In this case, the fault is a time delay at the output of case study. This fault modifies the residual value at the output of UIO, therefore the behaviour of selected patterns is modified. This results in a decrement of the confidence value, keeping a response of 82% during fault free scenarios, and a response of 19% and 41% during fault scenario.

In the fault-free scenario (Fig. 4.4), several types of patterns are classified. However, confidence value keeps a regular result, around 80%. In this case, 100% trust has not been achieved due to inherent time variant. As expected, residual value remains null during this scenario. Although, element response has not been accurately controlled by the local control law.

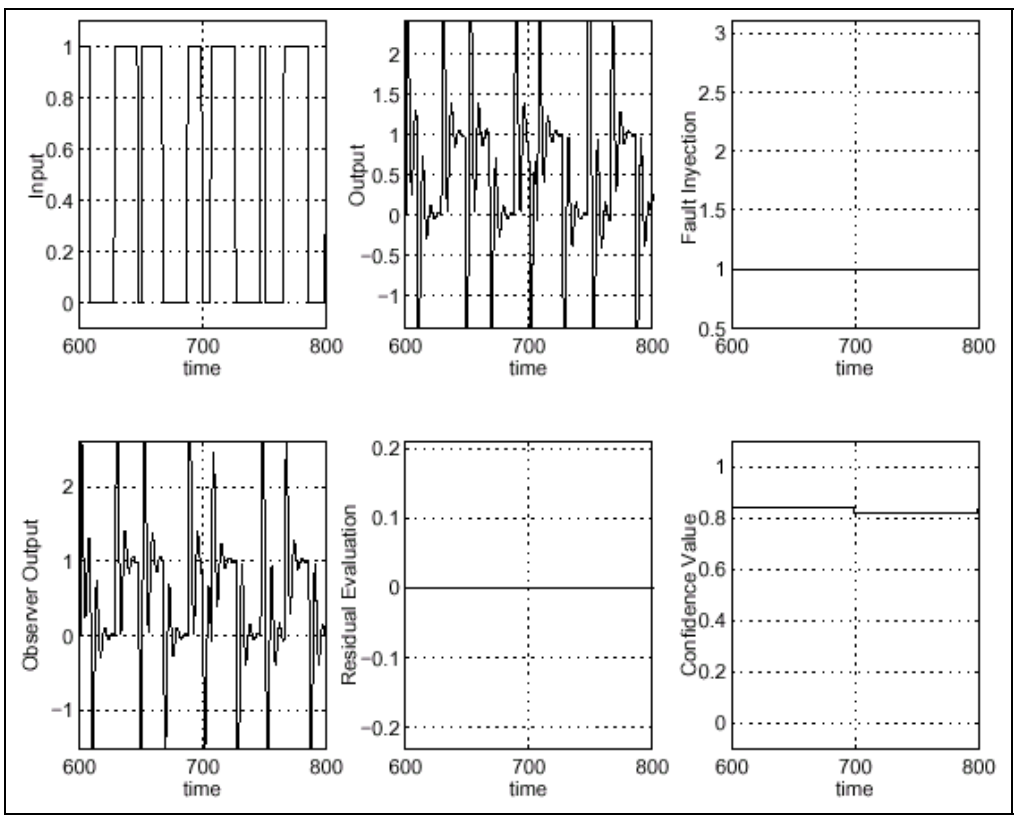


Fig. 4.4. Fault-Free Scenario with Time Variance Input Behaviour with Residual Evaluation

5. Conclusions

The combination of neural networks and analytical redundancy enhances the capabilities for fault localization. The key issue here is how data tends to be processed by the neural networks, in order to classify patterns. The integration of two neural networks in cascade allows the classification of time variant behaviour even during fault presence. This is possible due to an ART2A is used to determine the boundary between clusters from SOM output.

An important restriction of this approach is the sampling time window. This is inherent to the sampling technique. It is necessary to define a lower bound in terms of the number of samples, which is stated here as M. This value has a direct effect over Δt and frq_{fault}. However, in terms of sampling, there is no restriction regarding an upper bound. Its only practical restriction has to do with the response time. In such a case, faults that occur faster than this fault localization approach give a useless classification. Moreover, there is another clear restriction regarding to the possible *explosion* of the number of patterns.

The use of a bank of observers presents a formal approach in order to determine an isolated fault. This allows the isolation of fault-free and fault-specific scenarios, with very low time consumption for on-line performance. However, when an unknown scenario appears, it cannot declare a specific performance. Thus, neural networks represent an advantage as geometric classifiers. The integration of both, bank of observers and non-supervised neural networks, enhances the classification of abnormal scenarios such as unknown faults even with time variation.

6. Acknowledgments

The authors wish to thank projects PAPIIT-UNAM Num. IN101307 and Num. IN103310-3 for their financial support. Additionally authors would like to thank Mr. Duran-Chavesti for his fruitful support.

7. References

Nelles, 0. (2001). *Non-Linear Systems Identification*, Springer-Verlag, Berlin, Germany.

Venkatasubramanian V., Rengaswamy R., Kavuri, S., and Yin K.. (2003a). Computers and Chemical Engineering. *A review of process fault detection and diagnosis Part I: Quantitative model-based methods*, vol. 27, pp. 293-311.

Venkatasubramanian V., Rengaswamy R., Kavuri, S., and Yin K.. (2003b)., Computers and Chemical Engineering. *A review of process fault detection and diagnosis Part II: Qualitative models and search strategies*, vol. 27, pp. 313-326.

Venkatasubramanian V., Rengaswamy R., Kavuri, S., and Yin K. (2003c). Computers and Chemical Engineering. *A review of process fault detection and diagnosis Part III: Process history based methods*, vol. 27, pp. 327-346.

Liling Ma Yinghua Y., Fuli W. and Ningyun Lu. (2002). 15th Triennial World Congress, IFAC, Session slot T-We-M10. *A Neural Network Observer Approach for Actuator Fault Detection and Diagnosis in Non-linear Systems*, Barcelona, Spain.

Su T, Jhang J., and Hou C. (2008). International Journal of Innovative Computing Information and Control. *A Hybrid Artificial Neural Networks and Particle Swarm Optimization for Function Approximation*, Vol. 4, No. 9, pp. 2363 -2374.

Zhong P. and Wang L. (2008). International Journal of Innovative Computing Information and Control. *Support Vector Regression with Input Data Uncertainty*, Vol. 4, No. 9, pp- 2325-2332.

Billings S., and Wei H. (2005). IEEE Transaction on Neural Networks. *A New Class of Wavelet Networks for Nonlinear System Identification*, Vol. 14, No. 4, pp. 862-874.

Wang X., Chen B., Yang S. and McGreavy C. (1999). Computers and Chemical Engineering. *Application of Wavelets and Neural Networks to Diagnostic Systems Development, 2, an Integrated Framework and its Application*, Vol. 23, pp. 945-954.

Li D., Pedrycz W. and Pizzu N. (2005). IEEE Transaction on Biomedical Engineering. *Fuzzy Wavelet Packet based Feature Extraction Method and Its Application to Biomedical Signal Classification*, Vol. 52, No. 6, pp. 1132 -1139.

Benítez-Pérez, H. and García-Nocetti, F. (2005). *Reconfigurable Distributed Control*, Springer Verlag.

Benítez-Pérez H., García-Nocetti F. and Thompson H., International Journal of Innovative Computing, Information and Control. *Fault Classification Based upon Self Organizing Feature Maps and Principal Component Analysis for Inertial Sensor Drift*, Vol. 3, April 2007, Issue 2.

Benítez-Pérez H. and Alma Benítez-Pérez. (2009). International Journal of Innovative Computing, Information and Control, IJICIC. *The use of ARMAX strategy and Self Organizing Maps for Feature Extraction and Classification for Fault Diagnosis*, Vol. 5, No. 12, Diciembre, pp. 4787-4796.

Lee D., Thompson H. A., and Bennett S. (2000). *PID control for a distributed system with a smart actuator*, Digital Control: Past, Present and Future of PID Control (PID'00). Proceedings IFAC Workshop, pp.499-504.

Wang W. H., Liu K. D., Zhou D. H. and Wang C. H. (2002) 15th Triennial World Congress, IFAC. *A Fuzzy and Rough sets Integrated Approach to Fault Diagnosis*, pp. 1031-1037, Barcelona, Spain.

Abe S. (2001). *Pattern Classification: Neuro- Fuzzy Methods and their Comparison*, Springer-Verlag, UK.

Kiviluoto K.. (1995) *Topology Preservation in Self-organizing Maps*, Technical Report A29, Helsinki University of Technology, Laboratory of Computer and Information Science.

Lopez-García, H., and Machón González I. (2004). Engineering Applications of Artificial Intelligence. *Self-Organizing Map and Clustering for Wastewater treatment monitoring*, Vol. 17, pp. 215-225.

Benítez-Pérez H. and Alma Benitez-Pérez. (2010). Accepted en International Journal of Innovative Computing, Information and Control, IJICIC. *The use of WAVELET strategy and Self Organizing Maps for Feature Extraction and Classification for Fault Diagnosis*.

29

Use of SOM to Study Cotton Growing and Spinning

Josphat Igadwa Mwasiagi
School of Engineering, Moi University, Eldoret, Kenya

1. Introduction

From time immemorial cotton has maintained its position as one of the most important textile fibers. The world cotton production has continued to hit new records year after year due to the increase in cotton demand. The demand for cotton has also maintained a steady rise. This has resulted in increasing cotton prices (ICAC, 2004). Globally, cotton growing and processing has remained one of the profitable industries, which employs large sections of the populations and also earns foreign exchange. The Government of Kenya adopted Sessional Paper No. 2 of 1996 on the Industrial transformation of Kenya to become a Newly Industrialized Country by the year 2020, where the Textile Industry was identified as one of the key industries which could spur industrial growth (Kenyan Government, 1996a; Kenyan Government 1996b). This was due to the fact that the Kenyan textile industry grew rapidly in the 1970's and 1980's. By 1983, the textile industry was the second largest manufacturing industry after food processing.

The problems bedeviling the Kenyan cotton growing and processing industry can be traced back to the introduction of trade liberalization policies in 1982, continual deterioration of the Kenyan infrastructure, rising cost of energy and corruption practices (Kenya Government, 1982). The above mentioned problems left the industry without finances to keep up with the global changes in technology hence the Kenyan textile industry found itself using inefficient technology, which rendered its goods less competitive (Ikiara and Ndiragu, 2004). Currently Kenya is a net importer of textile products, a far cry from its state in the 1980's when Kenya was a net exporter of textile products (Mwangi, 2004). The need for Kenya to improve its cotton growing and processing industry can not be overemphasized. In 2000, for example the textile industry had an installed capacity to process 90,000 bales of cotton lint. Unfortunately the country produced only 20,000 bales and the short fall had to be imported, leading to a strain in the already meager foreign exchange (Ikiara and Ndirangu, 2004). A research carried out by the Kenyan Ministry of Agriculture to study Kenyan's potential to produce more cotton revealed that cotton can be grown on 2 million hectares. However in the last ten years the acreage of cotton has stagnant at around 50,000 hectares with very low yields (Estur, 2004). The reduced cotton acreage has brought doom to the cotton ginning industry, with the spinning industry having to import cotton lint while the farmers remain poor due to lack of employment (Kimenyi, 2001). The Government of Kenya has taken several steps to try and revive the ailing cotton growing and processing industry. Some of the key steps include; the removal of taxes related to ginning and textile machinery, the

supply of free cotton seeds to farmers and the drafting of a bill to manage the textile sector. The problems of the cotton farmers are however further complicated by the trends in the world cotton market and the end of Multi-Fiber Arrangement (Hussein, 2004; Panagariya et al, 2001; Cling et al, 2005). These global changes are likely to have a negative impact on the cotton growing industry in Kenya. In view of the changing trends of the Kenyan cotton growing sub-sector, a study which emphasized on the seed cotton production, yield per hectare and the cost of cotton growing was undertaken. Self Organizing Map (SOM) was used to study the relationship between cotton yield and farming practices. This will form the first section of this chapter.

Apart from considering the quantity of cotton grown the quality of the cotton is another key factor which will affect the competitiveness of the Kenyan cotton growing and processing industry. Cotton is a natural fiber and is therefore affected by the environment in which it is grown, ginned and marketed. This environment, which is responsible for the high variability of cotton lint characteristics, is as a result of complex interaction between several factors which include; climatic conditions, crop husbandry methods, government policies, spinners' requirements and the seed genetic inheritance (Rico, 2002; Joubert et al, 2002).

Cotton lint characteristics such as: fiber length, length uniformity, spinning consistency index, elongation, micronaire, strength, color, short fiber index and trash measurements, are used in fiber selection and marketing and can be measured by a variety of instruments. When the measurement of single fiber characteristics is done using a stand alone measuring instrument, it takes a lot of time and manpower to measure many samples. Consequently only representative measurements were taken and used in cotton grading and spinning before the introduction of the High Volume Instrument (HVI). The HVI system which incorporates all fiber measuring instruments into one testing unit was developed due to a concerted effort by researchers to provide an accurate, reliable and timely system which can measure important cotton quality characteristics speedily. In fact the HVI system can measure most of the known cotton quality characteristics in about forty seconds per sample The introduction of High Volume Instrument (HVI) system has however changed the measurement of cotton lint (Jones & Yankey, 1999). The HVI system incorporates several cotton properties measuring equipments which include; length/strength, moisture content, micronaire and color/trash equipments. The HVI equipment is fully automated and is supplied with relevant computer software and hardware which makes it possible to test the following cotton properties;

- Micronaire (measured by relating airflow resistance to the specific surface of fibers),
- Maturity (maturity ratio is calculated using a sophisticated algorithm based on several HVI measurements),
- length (upper half mean length, uniformity index and short fiber index; are measured optically in a tapered fiber beard which is automatically prepared, carded, and brushed),
- Strength (fiber strength (g/tex) and elongation at break (%) are measured physically by clamping a fiber bundle between 2 pairs of clamps at known distance. The second pair of clamps pulls away from the first pair at a constant speed until the fiber bundle breaks, hence the strength and elongation at break are measured),
- color (Rd (whiteness) and +b (yellowness) are measured optically by different color filters and then converted to customized color chart),
- trash (trash particle count (%) and surface area covered by trash are measured optically by utilizing a digital camera, and converted to trash standards).

The popularity of HVI can be attested to by the fact that its results are being used to grade cotton in many countries. The high testing speed of the HVI system enables the spinning mill to test each individual bale of cotton. While HVI system provides a lot of data for cotton lint the utilization of the data has however been limited to a use of one or two important characteristics such as Spinning Consistency Index (SCI) and micronaire for bale selection (Kothari, 1999; Majumdar et al, 2004). According to Kothari (Kothari, 1999) cotton bale selection can be done by first considering SCI followed by other HVI characteristics such as micronaire, fiber length or length uniformity depending on the spinning system. The aim of this procedure is to gain between mix long-term and short term consistency of all properties from all of the cotton bales available for yarn spinning. The bale selection procedure can be simplified and made more objective by using clustering techniques together with other statistical tools used for measuring central tendency and measure of dispersion for multivariate data. Clustering techniques group items into sets of similar objects based on given attributes (Jain & Dubes, 1988). The second section of this chapter will consider the classification of Kenyan cotton lint using data analysis techniques such as SOM, k-means clustering technique and Probabilistic Neural Network (PNN). The aforementioned techniques can be used to classify cotton lint based on its characteristics. Since the samples obtained for the classification of Kenyan cotton lint was less than 200, it was deemed necessary to consider the classification of a much large sample. In this connection HVI data for over 2000 bales was obtained from a testing institution in China. The collected data was used to classify the cotton bales using a model consisting of Kohonen Self Organizing Maps (SOM) to visualize the high dimensional cotton lint HVI data and K-means technique for clustering.

Having considered cotton growing and the classification of cotton lint using the SOM technique the final part of this chapter will consider the study of cotton yarn quality using SOM. The manufacture of cotton yarn involves assembling fibers in a set format and introducing some strength imparting twist. Consequently yarn may be defined as a linear assemblage of fibers formed into a continuous strand having textile-like characteristics. The textile-like characteristics referred to include good tensile strength and high flexibility (McCreight et al, 1997). The ring spinning process (Fig. 1) can be summarized into six stages; cotton mixing, blowroom, carding, drawing, speedframe and ringframe (Klein, 1987; Oxtoby, 1987; McCreight et al, 1997). From mixing to carding stages the cotton lint received from the ginnery undergo a series of processes which include opening, cleaning and fiber individualization and parallization. The cotton material is then gradually drafted (reduction in the diameter) in drawframe, speedframe and ringframe stages until the final yarn diameter is achieved. Twisting, which is also done at the ringframe stage is the final procedure in the yarn forming process. To ensure that the yarn manufactured in a spinning process meets specified standards, quality control measures are carried out. Quality control in the cotton spinning process involves the inspection of;

- Raw materials prior to entering the production line,
- Products at every stage in the spinning process, and
- Final products prior to dispatching.

The quality characteristics of the ring spun yarn include: yarn count, tensile properties, evenness and imperfections. Yarn count is a measure of the linear density of the yarn. It defines the yarn fineness and is highly correlated with other yarn quality characteristics like yarn strength, evenness and imperfections. The important yarn tensile properties are yarn

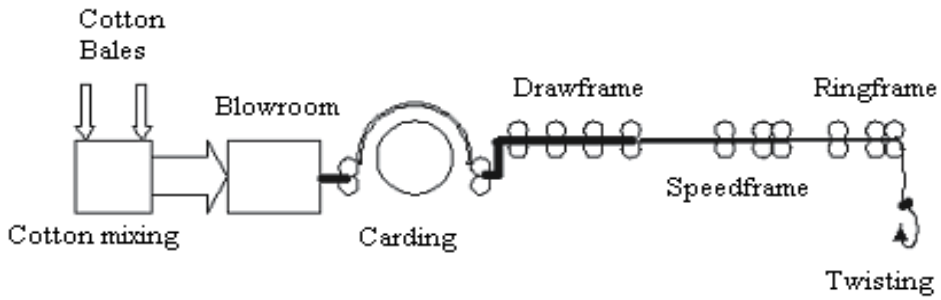


Fig. 1. The cotton Spinning process

breaking strength and breaking elongation. Yarn breaking strength (strength) is the force required to break the yarn when a tensile force is applied along its length. The percentage increase in length at break is referred to as breaking elongation (elongation). Yarn evenness (Kothari, 1999) refers to the variation in mass per unit length and can be measured using U%, which is the percentage mass deviation of unit length of material. CVm, which is the coefficient of variation of mass per unit length, has recently gained popularity as a yarn evenness unit. The relationship between the two factors is given in equation 1.

$$CVm = 1.25 \times U\% \tag{1}$$

Another important yarn quality characteristic is yarn imperfections, which is defined as frequently occurring yarn faults and can be subdivided into three groups: Thin places/km, thick places/km and neps/km. Yarn imperfections are caused by defects in raw material (fiber) or improper machine settings. The standard sensitivity levels for ring spun yarn imperfections are:

- Thin place (-50% : yarn cross section is only 50% of the mean value),
- Thick place (+50% : the cross section at thick place is 150% of yarn mean value),
- Neps (200%: the cross section at the nep is 200% of the yarn mean value)

Yarn irregularity occurs when there is a variation in any of the above yarn quality characteristics along the length of the yarn. Since many of the yarn quality characteristics are highly correlated with each other, a variation in one characteristic will adversely affect many other yarn properties. A case study of the Egyptian cotton spinning industry revealed that the main reason for quality problems were (Azzam & Mohamed, 2005):

- Unsuitable quality levels,
- Large quality variations, and
- Unexplained quality exceptions.

Unsuitable quality levels arise due to individual factories coming up with their own quality standards. This makes it difficult to compare products from different factories. This problem can however be solved by adapting internationally acceptable standards like Uster Quality Standards (Uster, 2007). Uster Statistics are quality reference figures which permit a classification of the quality of fibers, slivers, rovings and yarns with regard to world wide production. The samples which were used to formulate the latest Uster statistics were collected from different parts of the world. Uster standards can be used for quality benchmarking, product specification and comparison of spinning factories and products.

In view of the potential of the latent in the cotton growing and processing industry it is goes without saying that any effort however small it may be, taken to improve the profitability of

the industry is worthy the effort. The cotton growing industry generates high dimensional data, where many factors that affect the productivity of the industry can be linked to the yield of cotton. In the cotton spinning the quality of the lint which can be represented by over ten characteristics can be used to grade, hence improve its marketability and processing. Yarn spinning can also be improved if the high dimensional data generated from the quality characteristics of the yarn are analyzed accurately and timely. Given that, Self Organizing maps (SOM) can handle high dimensional data effectively enabling data visualization and clustering, this chapter will concentrate on the use of SOM, in the study of the cotton growing and processing industry.

2. Data processing algorithms

2.1 Kohonen Self Organizing Maps

Kohonen Self Organizing Maps (SOM) is an unsupervised competitive learning clustering network, in which only one neuron (or only one neuron in a group) is "on" at a time. SOM learns to recognize groups of similar input vectors in such a way that those neurons that are physically close together in the neuron layer respond to similar input vectors (Kohonen, 1997). The SOM algorithm has been applied in many areas due to its ability to provide an effective platform for visualization of high-dimensional data (Si et al, 2003). Kohonen Self Organizing Maps (SOM) learns to recognize groups of similar input vectors in such a way that neurons physically close together in the neuron layer respond to similar input vectors. The neurons in the layer of a SOM are arranged in physical positions according to a given topology and distance functions (Hagan et al, 2002). SOM learning algorithm exploits two fundamental interactions between neurons, namely, competition and cooperation. Competition during learning refers to the characteristics of a neuron, where each input vector is compared with each weight vector of the weight matrix. The winning neuron which is also referred to as the Best Matching Unit (BMU) is selected based on the least distance between the vector and the neuron. Cooperation learning on the other hand refers to the situation whereby all neurons located in a topological neighborhood of the winning neuron k have their weights updated in accordance with a neighborhood function φ(j) according to equation 2 (Jang & Mizutani, 1997).

$$\phi(j) = \exp\left(-\frac{\rho(j)^2}{2\sigma^2}\right) \tag{2}$$

where $\rho(j) = ||\mathbf{V}(j,:) - \mathbf{V}(k,:)||$ is the distance between the j^{th} and k^{th} neurons in the l-D (l =1,2 or 3) feature space, and σ^2 is a variance (or spread) of the Gaussian distribution.

When a vector is presented, the vectors of the winning neuron and its close neighbors will move towards it. Consequently after many presentations, the neighboring neurons will have learnt vectors similar to each other. The architecture for SOM is given in Fig. 2. The ||dist|| box accepts the input vector p and the input weight matrix $IW_{1,1}$, and produces a vector having S^1 elements.

The competitive transfer function accepts a net input vector for a layer and returns neuron outputs of 0 for all neurons except for the winner. The quality of the SOM maps can be checked by using two factors: data representation accuracy and data set topology representation accuracy. Data representation accuracy can be measured using average distance between each data vector and its BMU. Data set topology representation accuracy

can be expressed as the topographic error which is the percentage of data vectors for which the first and second BMUs are not adjacent units.

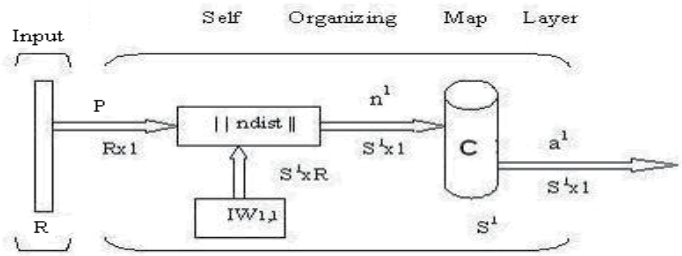


Fig. 2. The Architecture of SOM Network

2.2 Probabilistic Neural Network

Probabilistic Neural Network (PNN) is a type of a classification Artificial Neural Network (ANN) that provides a general solution to pattern classification problems by following the probabilistic approach based on the Bayes Decision Theory (Specht, 1996). In comparison to other multilayer feedforward neural networks it is not trained iteratively, thus it trains faster and involves less parameters for its optimization. Other advantages of PNN over standard neural networks and traditional statistical techniques include their robustness to noisy data (with outliers), which can harm many types of ANN and can severely hamper most traditional statistical methods. The architecture of a PNN is given in Fig. 3, where R is the number of inputs, Q is the number of input/target pairs and k is the number of classes of input data. The first radial basis layer produces a vector whose elements indicate how close the training input is to the training output. The competitive layer forms a vector of probabilities for each class of the input–output pair and picks the maximum of these probabilities to produce a "1" for that class and a "0" for the other classes. PNN can be used for classification problems. Their design is straightforward and does not depend on training, because it typically involves choosing the optimum value for the spread in the first layer. A PNN is guaranteed to converge to a Bayesian Classifier provided it is given enough training data. The efficiency of a PNN can be improved by adapting an appropriate training data set selection method (Bolat & Yildirium, 2003).

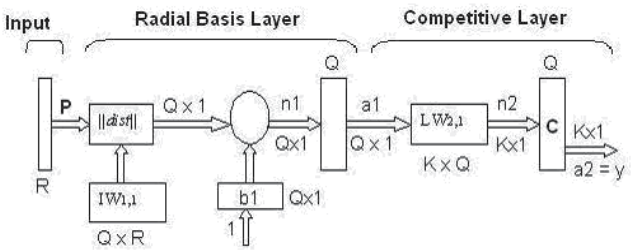


Fig. 3. The architecture of probabilistic network

2.3 k-means Clustering Technique (for bale classification)

Clustering algorithms attempt to organize unlabeled feature vectors into clusters or natural groupings such that vectors within a cluster/group are more similar to each other than

other vectors belonging to different clusters. K means is a partitioning nonhierarchical clustering method that constructs k (k fixed a *priori*) clusters for a given data set. The K-means algorithm defines k centers one for each cluster and hence has k groups. The grouping is done by minimizing the sum of squares of distances between the data members and the corresponding cluster centers. This is achieved by minimizing a squared error objective function;

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \tag{3}$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is the chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centers. The algorithm is composed of the following steps;

i. Place k points into the space represented by the objects that are being clustered
ii. Assign each object to the group whose center is closest
iii. When all objects have been assigned, recalculate the positions of the k centers
iv. Repeat the second and third steps until the centers no longer move. This produces a separation of the objects into groups from which the factor to be minimized can be calculated. These points represent initial group centers.

The quality of the data partitioning using K-means clustering techniques can be analyzed by using cluster validity methods such as Davies Bouldin (DB) index and silhouette means. The DB index is a function of the ratio of the sum of within-cluster scatter to between-cluster separation and can be computed as shown in equation 4. In Eq. (4) *Sc* and *dce* denotes the centroid intra-cluster and inter-cluster distances respectively. The intra-cluster distance for a given cluster is obtained as the average of all pair wise distances from points in the cluster to the cluster centroid.

$$DB = \frac{1}{N_c} \sum_{j=1}^{N_c} \frac{\max_{j \neq k}(Sc(k) + Sc(j))}{dce(k, j)} \tag{4}$$

where $Sc = \dfrac{\sum_i | \, | x_i - c_k \, | \, |}{N_k}$ and $dce = | \, | c_k - c_j \, | \, |$.

The inter-cluster distance between two clusters is computed as the distance between their centroids. N_k is the number of objects belonging to cluster k, given that a total of N_c clusters are found to exist in the data. The DB index can be plotted for several numbers of clusters, whereby the cluster value with the lowest DB index will be deemed to be the optimum number of clusters for the data. The Silhouettes method on the other hand uses width plots. The silhouette width S_i for the i^{th} feature vector in a cluster q is defined by equation 5 (Jain & Dubes, 1988).

$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)} \tag{5}$$

In Eq. 5, a_i is the average distance from the i^{th} feature vector to all other feature vectors in the cluster k; b_i is the minimum average distance from the i^{th} feature vector to all the feature

vectors in another cluster j (j =1,….,q; j ≠ q). When S_i is close to one, this implies that a_i is small with respect to b_i. This is an indication that the vector is correctly classified. If S_i is close to zero, this implies that a_i and b_i are approximately equal, thus making it unclear which cluster i should belong to, while negative S_i implies that i is assigned to the wrong cluster. For given k clusters, the overall average silhouette width is the average of the silhouette widths for all the feature vectors in the data set. The partition with the maximum overall average silhouette width is taken as the optimal partition. As stated earlier, in k-means clustering, the value of k is unknown a *priori*. There is need therefore to use another algorithm to determine the value of k. SOM is good at pattern recognition, and hence can be used to determine the nature of clustering in a given data. This information can be used to determine the value of k, which can then be passed on to the K-means clustering algorithm, for the clustering of the data (Vesanto & Alhoniemi, 2000).

2.4 Statistical techniques

There are many methods in statistical analysis which can be used to organize and summarize data. These can be studied in descriptive statistics. There are also many descriptive measures such as measure of tendency and measure of dispersion. Given a sample of data set for one attribute the measure of central tendency such as means, mode etc and the measure of variation such as standard deviation and coefficient of variation (CV) can be used to describe the data. (Bluman, 2004; Weiss, 1997). The results of the above analysis can be displayed using two dimensional graphs. In real life situation there are many occasions when the data to be analyzed is multivariate (i.e. it has more than one attribute). A good example of a multivariate data is HVI data, which may contain say 100 samples of cotton lint, samples each having over characteristics producing over 1000 data, having 100 rows and over ten columns. This type of data will be hard to visualize. Graphical display of data can only be done for at most three variables displayed in a three dimensional plots. For data with more than three variables it is hard to visualize their relationship. Boxplots (box-and –whisker diagram) together with principal components analysis technique can be used to study the characteristics of such high dimensional data.

A common characteristic of many multivariate data is that the variables within a given data set often move together. This may be partly due to the fact that several variables may be measuring the same driving principle governing the behavior of the system. Another common characteristic of data set with many variables is that in many systems there are only a few such driving forces. There are however an abundance of instrumentation which can be used to variables. Taking advantage of this redundancy of information, simplifies the problem by replacing a group of variables with a single new variable. The above principles are referred to as Principal components analysis. This method generates a new set of variables, called principal components. Each principal component is a linear combination of the original variables. All the principal components are orthogonal to each other so there is no redundant information.

3. Application of SOM in the cotton growing and processing industries

3.1 Use of SOM to study Cotton growing

3.1.1 Data and sample collection

The cotton growing process involves farming management techniques which may involve handling a lot of data. In Kenya the District Agricultural officers (DAO) collect data

pertaining to the growing conditions of all crops within a district. This data is archived at the Ministry Headquarters, which handles data from all the districts in the country. The data has been collected for many years and it has become too voluminous for the traditional data mining techniques to give a meaningful interpretation. An attempt to access the data so as to analyze it using SOM was not well received since the Agricultural officers were not very familiar with the advantages of the SOM techniques. The aim of this section was to collect data from the farmers in selected districts for the 2003/4 and 2004/05 cotton growing seasons and to use SOM techniques so as to demonstrate its advantages. Data on cotton growing was collected from District Agricultural officers (DAO) and cotton farmers in all the cotton growing regions. Data was also collected from the Ministry of Agriculture (MOA), which included districts, provincial and national levels. Interaction with all the stake holders in the cotton growing industry like farmers, ginners, DAO's, fiber research institutions, cotton processing mills and MOA personnel was done through direct interviews and questionnaires. Cotton lint samples were collected from all the ginneries which were operating during the study period. The collected samples were analyzed using the internationally acceptable High Volume Instrument (HVI) system.

3.1.2 Results and discussions

3.1.2.1 The trend of cotton production

The trend of the cotton growing industry in Kenya was analyzed using the production of seed cotton data obtained from the Ministry of Agriculture in Nairobi. As shown in Fig. 4, the production of seed cotton from 1966 to 2004 showed a steady increase up to 1978, when it stated to decline. The decline in cotton production in the Early 1980's can be attributed to high levels of mismanagement in the government ministries, parastatals and co-operatives as reported by Ndegwa (Kenyan Government, 1982). The government of Kenya tried to revive the cotton growing industry in the mid 1980's and this lead to an increase in cotton growing in 1986. However subsequent reduction in cotton production could be attributed to the liberalization of the Kenyan economy and the decrease in the world cotton prices.

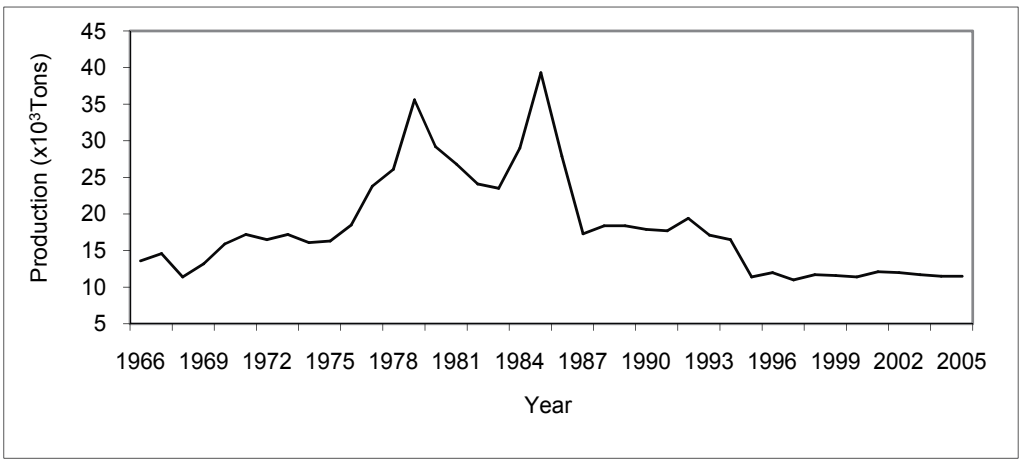


Fig. 4. Cotton production in Kenya

Having established the trend of the cotton growing industry data pertaining to cotton growing was collected from the DAO's from five years before 2004. A mean of this data was taken and designated as "mean for 5 years" (Table 1).
Data for the 2004/05 cotton growing season was collected from farmers and DAO's from all the cotton growing regions in Kenya. The mean of the collected data was designated as "2004/05 season" (Table 2). The collected data shown in Table 2 indicated that the cost of producing seed cotton has been declining in the last five years.
While the average break-even cost for the last five years was Kes 25 (US$ 0.33), the break-even cost in the 2004/05 season was Kes 23 (US$ 0.31). The price offered to the farmers during the 2004-2005 cotton growing season was Kes 22 (US$ 0.29) thus the farmer lost an average of Kes 1 for every kg of seed cotton grown.
The price of cotton during the 2004-2005 cotton season showed that over half of the cost (51%) were spend on land preparation and weeding. Since most of land preparation and weeding is done manually the Kenyan farmer could reduce his cost by adapting mechanized farming systems. There is need for research to be done in the so as to reduce the Kenyan farmers cost of land preparation. The amount of fertilizers and pesticides showed a decreasing trend. This could be attributed to rising poverty levels among the rural folks in Kenya (Kemenyi, 2001).

Description	Cost (Kes/ha)	
	Mean for 5 years	2004/2005
Ploughing	3129	3600
Harrowing	1496	1268
Seed	232	253
Fertilizers	969	630
Planting	1223	554
Weeding	3299	1720
Pesticides	2659	1776
Harvesting	3508	1507
Transport	875	1077
Others	456	507
Total cost	17847	12891
Yield (Kg/ha)	717	571
Break-even cost	25	23

Note: 1 ha = $10^4 m^2$
1 US$ = 70.67 kes

Table 1. Cost of growing cotton in Kenya

3.1.2.2 Cost analysis

Having established the factors related to cotton growing analysis of the cost of cotton growing in Kenya was done using the SOM technique. The analysis of cluster formation for the factors affecting the cost of cotton growing identified six cluster groups (Table 2).
During the clustering process it was evident that cotton yields is positively correlated with the cost of pesticides and weeding. This is expected. Pesticides control pests and hence boost

cotton yields. Proper control of weeds also leads to higher yields because the cotton crop is able to grow unhindered. Due to high poverty levels which are prevalent in the Kenyan cotton growing areas the farmers use lower than the required amounts of pesticides and sometimes do not weed properly.
From Table 2, it is clear that good yields come from a combination of optimum usage of pesticides, fertilizers and proper land preparations. There are other factors that need further consideration. Harrowing for example is a mandatory cost in some areas having black cotton soils, like in the Western regions, while it is not necessary at all in the Eastern regions. While there are possibilities of planting without ploughing, as reported in other parts of the world, this may need more research before it ca be applied in Kenya.

	Cost Distribution (%)					
Cluster	One	Two	Three	Four	Five	Six
Pesticides	27	2	34	17	7	10
Harvest	18	27	18	26	18	13
Fertilizer	17	14	3	2	2	2
Weeding	16	23	14	21	22	16
Ploughing	10	14	14	13	21	26
Plant	5	7	6	7	8	7
Transport	3	4	3	8	4	5
Others	3	4	3	2	2	2
Seed	1	1	1	1	1	2
Harrowing	1	4	5	3	13	18
Yield(Kg/ha)	1696	1211	1105	851	542	403

Table 2. Grouping of regions based on costs

3.2 Classification of cotton lint using SOM techniques

3.2.1 Sample collection and data analysis

As mentioned in section 3.1 the cotton lint samples used in this research work were obtained from the 2003/2004 and 2004/2005 cotton growing seasons in Kenya. During cotton lint sample collection it was necessary to ensure that the samples collected could be deemed to be representative of the cotton lint grown in Kenyan's three cotton growing regions; coastal, western and eastern/central. Consequently the samples collected were 30, 66 and 96 which correlated to the percentages of cotton lint produced in the coastal, western and eastern/central regions respectively. The cotton lint samples were measured using the Uster HVI spectrum instrument and 13 cotton characteristics were used as inputs for SOM. The total data had 192 cotton samples and 13 cotton HVI characteristics, namely micronaire, maturity, length, elongation, strength, short fiber index, length uniformity, spinning consistency index, reflectance, yellowness, trash count, trash area and trash grade.
The high dimensional HVI data was first visualized using SOM toolbox (software freely available under GNU license) (Vesanto et al, 2000). The quality of the SOM visualization process was checked using data representation accuracy and data set topology representation accuracy. The visualization of the high dimensional data using SOM gave an idea of the nature of clustering in the data and hence the value of k was determined. This

information was thereafter used to partition the data using k-means clustering technique. Having partitioned the data PNN was designed to classify the cotton lint clusters. The HVI data was divided into a ratio of 4:1 for training and simulation (unseen) data sets respectively. First the PNN was trained using the training data set and then tested with the unseen data set. Different levels of spread ranging from 1 to 20 were tried in order to study the effects of spread on the efficiency of the PNN, and hence be able to find the optimum operational design for PNN. The performance of the PNN algorithm was measured by taking a regression between the predicted and target value.

3.2.2 Results and discussions

3.2.2.1 Data Visualization

SOM reduced the 192x13 HVI data into a 12x6 grid, with a final quantization error of 1.647 and final topographic error of 0.010. The pattern codes and number of hits for the nodes (Figure 5) indicate that the cotton lint samples can be sub-divided into four clusters.

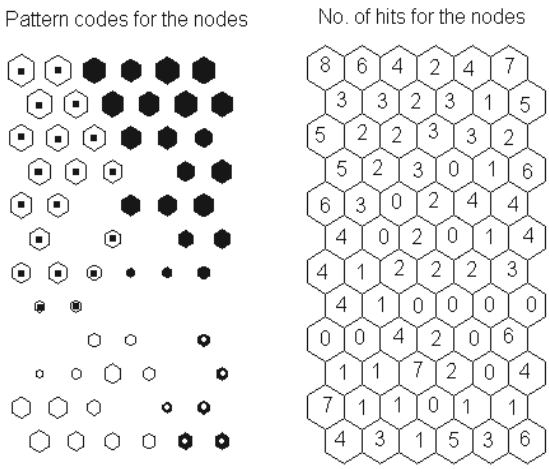


Fig. 5. Pattern codes and No. of hits for SOM

3.2.2.2 Clustering of Cotton Lint using k-means Clustering Technique

Having established from the SOM maps that the HVI data for the cotton lint can be sub-divided into 4 clusters, k-means clustering technique was used to partition the cotton lint data into four clusters. The means of the four clusters are given in Table 3. The clustering show that the longer Kenyan cottons tend to have higher values of micronaire, maturity, spinning consistency index, strength, uniformity, elongation and reflectance, but lower values for yellowness, short fiber index and trash measurements values (trash cent, trash area and trash grade).

3.2.2.3 Classification of Cotton lint clusters using PNN

After clustering the cotton lint HVI data into four clusters (classes), PNN was used to predict the class number.

The PNN algorithm was designed using four fifth (4/5) of the data and the other one fifth (1/5) was used for cluster testing. The variation of R-value at different levels of spread during the design of the algorithm is given in Fig. 6.

The PNN attained its optimum performance level (R-value =1.00) of being able to classify the clusters of the HVI data at a spread value of 6 and 7.

HVI factor	Clusters			
	1	2	3	4
Micronaire	4.03	3.95	3.31	3.06
Length	30.38	29.96	29.27	29.23
Maturity	0.88	0.86	0.82	0.81
SCI	164	157	135	143
Strength	32.57	30.25	26.85	27.78
Uniformity	85	84	82	83
Short fiber Index	7.73	8.13	9.56	9.73
Elongation	6.18	6.16	5.88	5.86
Reflectance (Rd)	76.45	79.53	70.68	66.43
Yellowness (+b)	10.95	11.25	13.64	14.79
Trash Cent	27.54	11.93	33.27	69.96
Trash Area	0.40	0.15	0.48	0.91
Trash Grade	4	2	4	6
No. of Cases	26	90	49	27

Table 3. The means for cotton lint

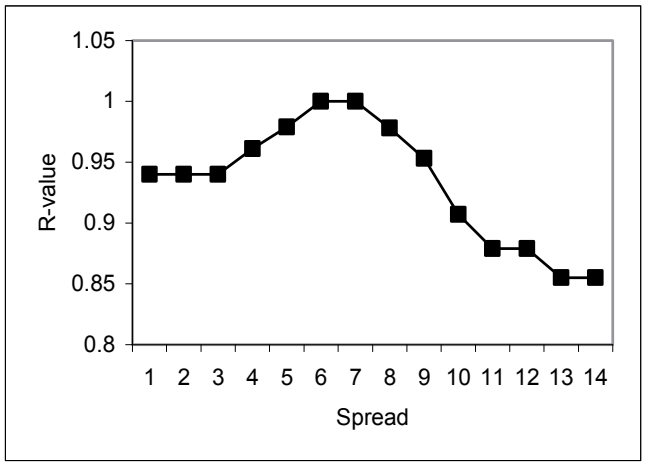


Fig. 6. Performance of PNN algorithm

3.3 The use of SOM to classify cotton bales

3.3.1 Data collection and analysis

The HVI data used in this study was obtained from the Shanghai Inspection Center of Industrial Products and Materials (SICIPM). SICIPM is mandated to inspect all industrial products and material entering or leaving China, through Shanghai port. SICIPM has a well equipped laboratory for testing the properties of cotton fiber. HVI data from the SICIPM

HVI data bank for cotton bales samples meant for importation into China through Shanghai port between October 2005 and July 2006 was selected. The total data sample had 2421 cotton bales and 13 cotton HVI characteristics, namely micronaire (Mic), maturity (Mat), length (Len), elongation (Elg), strength (Str), Short Fiber Index (Sfi), length uniformity (Unf), Spinning Consistency Index (Sci), reflectance (Rd), yellowness (+b), trash cent (Trc), trash area (Tra) and trash grade (Trg). The high dimensional HVI data containing 2421x13 inputs was first partitioned using K-means algorithm. DB index was run for all clusters partitioning starting from 2 to $\sqrt{2421}$ and the cluster with the lowest DB index was selected. Since DB index algorithm in the SOM toolbox starts out with random centers, the optimum cluster could vary from time to time depending on the random centers selected. The cluster validity using DB index was run for a 100 times and the least number of cluster was selected as the optimum cluster number and hence designated as k.
The HVI data was then partitioned into k groups and the Coefficient of variation for each group was checked for all the HVI attributes. Any group which was found to have a CV higher than that of the main group (MG) for any of its attributes was declared to have failed. The failed groups were further analyzed using other cluster validity methods and statistical techniques, such silhouette means and principal component analysis. The bale classification model can be summarized as follows;

i. Use SOM data visualization technique to get an idea of the nature of clustering within HVI data,
ii. Partition the HVI data using K-means technique. The value of k should be obtained from (i) above, and
iii. Check for data group compactness using other methods and techniques such as silhouette means, coefficient of variation and principal component analysis.

3.3.2 Result and discussion

3.3.2.1 Data Visualization

Data visualization was done using the SOM techniques and K-means technique. The SOM technique reduced the high dimensional of 2421x13 to a much smaller data 18x13. The SOM quantization error is 1.879 and topographic error of 0.083. These are acceptable as per the standards of SOM algorithm and can therefore be used to study cotton bale classification.
The partition of the HVI data was done using K-means clustering technique, and the occurrence of optimum cluster was obtained by using DB index as described in the methodology section. The optimum number of clusters (groups) as shown in figure 7 is 19.
The number of hits for each node of the SOM map is given in Fig. 7, which shows that there are multiple hits in 231 out of the 234 nodes of the visualization grids.
The means and the Coefficient of variation (CV) for the groups are given in Tables 4 and 5.
The CVs for the 2nd, 7th and 17th group are higher than that of the main group (MG) for some attributes. These three classes are therefore considered not to be properly partitioned. The other 16 classes are considered to well partitioned, producing classes which are compact and different from one another.

3.3.2.3 The Re-combined Subgroup (RSG)

The visualization and the hits for the re-combined sub group (RSG), made of the members of 2nd, 7th and 17th groups is given in Fig. 8, which shows that the level of grouping is very low with only five out of the 66 nodes having hits of more than 5, and about 44% of all the nodes have one (1) or zero hits.

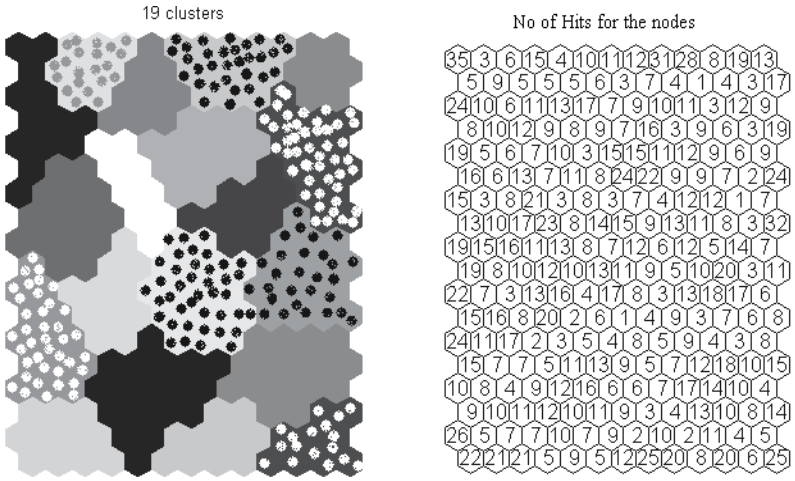


Fig. 7. Pattern codes and No. of Hits for Bale classification

Group No.	Sci	Len	Unf	Str	Elg	Sfi	Mic	Mat	Rd	+b	Trc	Tra	Trg	No. of Data
1	99	1.05	78.59	24.04	5.79	12.71	4.36	0.86	77.70	9.14	21.94	0.28	2.56	34
2	108	1.05	79.44	25.30	5.94	12.39	4.38	0.86	78.96	10.37	24.12	0.34	2.93	57
3	110	1.07	79.78	25.01	5.88	11.80	4.28	0.86	79.46	9.23	13.35	0.17	1.58	84
4	110	1.07	79.79	25.01	5.89	11.81	4.28	0.86	79.49	9.23	13.31	0.16	1.55	83
5	111	1.07	80.18	25.93	5.79	11.68	4.44	0.87	75.38	8.62	34.66	0.43	3.54	71
6	116	1.07	80.27	26.16	6.05	11.18	4.31	0.86	79.19	9.69	22.00	0.26	2.47	123
7	117	1.09	80.86	26.44	5.90	11.11	4.28	0.86	74.28	9.03	52.20	0.66	4.68	25
8	122	1.09	81.49	27.27	6.11	10.51	4.37	0.87	74.98	8.96	47.47	0.55	4.20	55
9	123	1.09	81.42	27.18	6.01	10.34	4.42	0.87	77.81	9.40	30.56	0.33	3.06	125
10	124	1.09	80.82	26.70	6.07	10.70	4.18	0.86	80.89	9.63	11.71	0.16	1.56	186
11	129	1.10	81.57	27.34	6.04	10.21	4.15	0.86	80.36	9.70	19.75	0.25	2.32	310
12	132	1.11	81.49	27.77	5.94	10.55	3.72	0.84	76.51	8.97	45.21	0.48	3.82	82
13	136	1.12	80.91	27.58	6.12	10.98	3.16	0.82	78.60	8.98	40.27	0.39	3.16	49
14	137	1.12	82.78	28.62	6.03	9.50	4.25	0.87	78.27	9.32	33.26	0.43	3.57	176
15	138	1.13	82.45	28.07	6.17	9.55	4.10	0.86	81.15	9.18	8.93	0.12	1.22	306
16	142	1.14	82.66	28.56	5.98	9.45	3.98	0.86	80.64	9.54	18.51	0.26	2.32	279
17	149	1.17	84.04	29.48	6.80	8.78	4.45	0.89	81.46	9.89	12.76	0.19	1.95	62
18	151	1.17	83.86	29.57	6.13	8.68	4.16	0.87	81.48	9.48	11.07	0.15	1.49	205
19	162	1.19	84.83	31.70	6.15	8.46	4.17	0.88	80.80	9.54	19.03	0.26	2.39	109

Table 4. Means for the 19 groups of cotton bales

Group No.	Sci	Len	Unf	Str	Elg	Sfi	Mic	Mat	Rd	+b	Trc	Tra	Trg
1	3.67	2.76	1.50	5.69	8.65	10.41	10.29	3.04	3.44	12.29	31.32	50.20	45.65
2	16.45	5.13	2.77	9.87	12.44	18.52	12.86	3.80	5.59	16.04	43.53	83.85	56.41
3	5.45	3.19	1.50	6.24	8.49	10.13	12.30	3.16	2.69	11.70	34.37	59.40	54.44
4	5.48	3.13	1.50	6.28	8.50	10.14	12.37	3.18	2.69	11.77	34.59	56.50	53.09
5	3.58	3.31	1.57	5.81	8.24	10.15	11.85	3.21	3.47	12.13	13.21	43.42	27.61
6	2.89	3.43	1.60	5.61	9.07	9.67	12.98	3.35	3.39	11.53	14.56	47.19	42.33
7	12.50	4.25	2.55	10.52	6.17	17.47	13.88	4.19	5.41	9.56	41.41	30.44	19.79
8	2.98	3.16	1.54	4.64	8.25	9.28	13.04	3.38	3.79	12.04	7.00	41.78	24.20
9	2.88	3.03	1.49	5.44	8.11	10.75	11.94	3.19	3.81	11.53	9.63	36.36	24.59
10	2.73	3.53	1.50	5.95	9.66	10.27	11.83	3.19	2.81	11.49	26.20	57.21	53.38
11	2.90	3.20	1.58	5.36	9.70	10.71	13.04	3.25	3.09	10.92	17.43	65.09	46.51
12	5.10	3.86	1.78	6.56	8.64	9.74	12.34	2.88	3.88	10.23	15.06	29.52	21.70
13	8.59	3.09	1.77	5.41	6.39	9.51	10.33	1.92	3.65	10.10	46.79	58.70	40.06
14	5.00	3.60	1.43	6.12	9.19	9.55	13.02	3.26	3.95	12.10	16.48	49.27	28.22
15	3.02	3.21	1.31	5.52	9.32	8.78	9.81	2.59	3.07	11.58	35.15	73.38	50.13
16	2.78	3.41	1.67	6.15	9.11	9.59	12.92	3.28	2.85	12.21	16.59	68.71	45.65
17	12.18	6.52	2.42	11.91	15.33	15.45	15.16	4.46	2.87	16.43	60.64	100.92	64.70
18	2.35	2.94	1.41	6.63	9.41	8.87	10.28	2.84	2.93	11.57	43.48	60.55	51.94
19	4.57	3.68	1.77	8.01	8.34	8.82	11.33	2.94	3.12	11.58	37.92	57.64	47.34
MG	11.71	4.78	2.36	8.55	9.73	14.70	13.14	3.39	3.96	12.31	58.70	74.12	55.48

Table 5. CV for groups of cotton bales

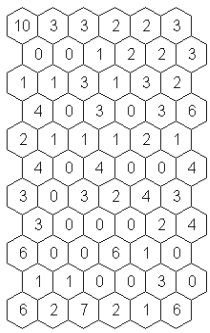


Fig. 8. Visualization for the re-combined 2, 7 and 17 groups

The silhouette means for the RSG is given in Fig. 9. The optimum clustering is 3, which was the initial number of groups in RSG before re-combining. Checking out the principal components of the attributes indicted that 97. 87% of the variance is accounted for by the first two principal components.

A scatter plot of the scores of the first two principal components (Fig. 10), shows that there are three distinct regions. These regions marked A, B and C are widely scattered but region B and C seem to be more scattered than A. B and C can be subdivided into two groups each. Thus RSG will now have a total of five subgroups.

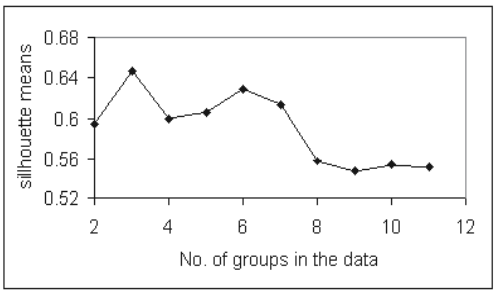


Fig. 9. Clustering the RSG

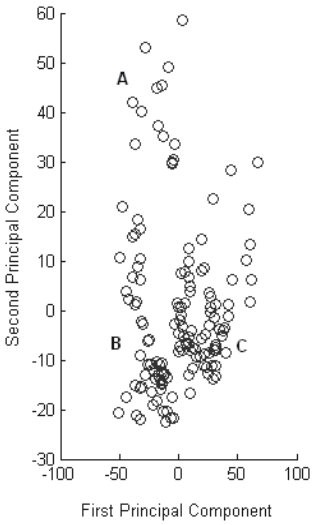


Fig. 10. Principal Component scatter for RSG

The means and CVs of the five groups of RSG are given in Tables 7. These subgroups still do not show a sense of being more compact than the MG, since all the groups still have attributes with CVs which are higher than the original MG. Further subdivision of RSG will produces very small subgroups having less than ten members, so it can be concluded that RSG is actually a collection of outliers which can be subdivided into five sets.

3.4 Study of cotton yarn quality

3.4.1 Sample collection and data analysis

For the purpose of studying the quality of cotton yarn, cotton lint and carded ring spun yarn samples were collected from textile factories in Kenya. For every yarn sample collected, a sample of the corresponding cotton lint mixture used to spin the yarn was also collected. Cotton lint samples from three regions, Voi, Kitui and Western Tanzania (WT) were collected. Table 8 gives the details of the cotton lint and yarn samples collected.

Attributes	Group Means						Group CV					
	RSG 1	RSG 2	RSG 3	RSG 4	RSG 5	MG	RSG 1	RSG 2	RSG 3	RSG 4	RSG 5	MG
Length	1.042	1.118	1.109	1.212	1.047	1.115	4.97	4.00	3.96	5.77	3.43	4.78
Uniformity	78.96	82.65	82.19	85.23	78.46	81.88	2.19	1.57	2.00	2.11	1.93	2.36
Strength	24.33	27.82	28.63	31.22	24.62	27.70	7.71	6.74	5.02	12.38	7.65	8.55
Elongation	6.03	6.41	6.03	6.88	5.78	6.06	13.59	17.89	5.01	12.33	5.83	9.73
Sfi	12.88	9.69	10.18	7.99	12.85	10.12	16.25	11.80	13.59	11.48	16.26	14.70
Micronaire	4.27	4.42	4.53	4.43	4.45	4.16	12.91	18.23	8.41	9.58	15.09	13.14
Maturity	0.86	0.88	0.88	0.89	0.86	0.86	3.89	4.84	2.45	3.49	4.42	3.39
Rd	78.78	81.43	73.66	82.26	73.36	79.70	4.18	3.45	4.58	2.26	5.30	3.96
+b	10.52	9.52	8.96	10.49	9.23	9.43	17.28	12.70	10.00	16.45	11.89	12.31
Trash cent	17.60	17.88	66.77	11.38	46.12	20.70	30.30	46.07	14.70	75.55	19.56	58.70
Trash area	0.35	0.23	0.67	0.20	0.53	0.26	101.12	60.23	15.45	123.68	41.71	74.12
Trash grade	2.88	2.29	4.77	2.03	3.94	2.32	68.77	49.87	12.07	72.27	29.48	55.48
Sci	103	135	128	163	96	132	9.28	5.37	8.39	8.64	5.48	11.71
No. of data	40	42	11	32	17	2421	40	42	11	32	17	2421

Table 7. Means and CV for the 5 groups in the RSG

Yarn Type	Cotton Source	Mill Code	Cotton Code	Yarn Tex	Spindle Speed (rpm)	Draft	Ring dia (mm)
a	WT	A	T20	20	12000	26	42
b	Kitui	A	K20	20	12000	26	42
c	Voi	B	V20	20	11000	25	42
d	Kitui	A	KA2	25	11,000	21	42
e	Kitui	C	KC2	25	8000	30	50
f	Voi	B	V30	30	10000	20	42

Table 8. Details of cotton lint and Yarn samples

The cotton lint samples were tested using the Uster Spectra 1000 HVI system, and the results, which include fiber length, length uniformity index (Unf), micronaire (mic), maturity, spinning consistency index (SCI), strength (Fstrength), elongation at break (Fe%), short fiber index (SFI), reflexion (Rd), yellowness (+b), trash cent (TrC), trash area (TrA) and trash Grade (TrG) were taken for the model.

Each yarn sample collected consisted of 25 yarn cops randomly selected from a group of ringframes running a given count. Thus a total of 150 ringframe cops spun according to the machine settings given in Table 11 were collected. Various yarn testing equipments were used to test the yarn quality characteristics, the yarn quality characteristics were analyzed using Somtoolbox (Vesanto et al, 2000).

3.4.2 Results and discussions

3.4.2.1 Quality of the yarn according to Uster Standards

The Uster standard levels for six yarn quality characteristics (evenness (CVm), strength, elongation (E%), Thin places/km (Thin), Thick places/km (Thick) and Neps/km (Neps)) are given Fig. 11. The Uster Standards for evenness (CVm) for *a*, *c* and *d* were at 75%. All the yarn samples showed average (50%) Uster standard level for yarn strength, while the Uster standards for yarn elongation and imperfections (thin, thick and neps) varied from 5% to 50%.

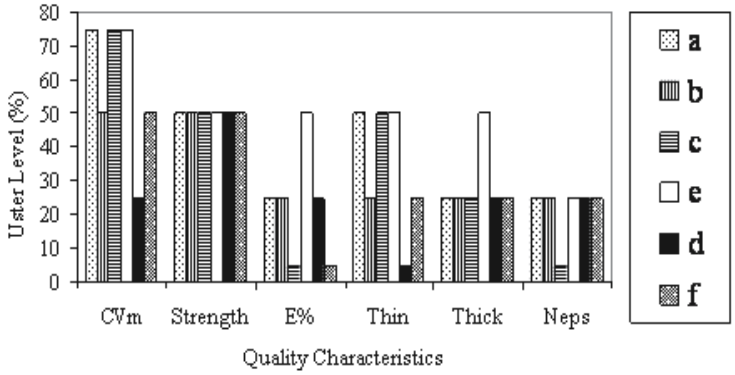


Fig. 11. Uster Standards for Kenya Cotton Yarn

In Fig. 12 the coefficient of variation (CV) for the yarn quality characteristics is given. Yarn evenness (CVm) and tensile properties (strength and elongation) showed low CV, while the CV for imperfections (thin places/km, thick places/km and Neps/km) was high. This implies that imperfections were widely scattered along the length of the yarns, leading to high levels of irregularities in the yarns.

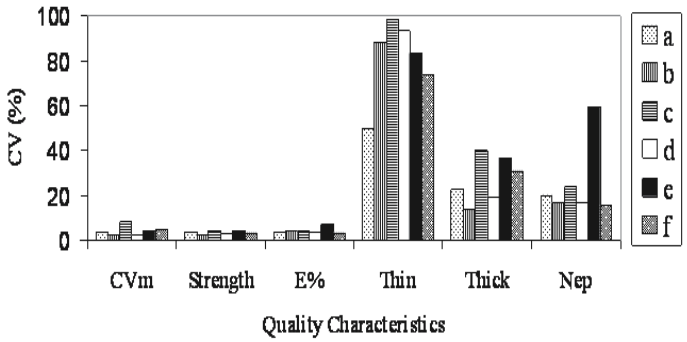


Fig. 12. Quality of Kenyan Yarn

This could be responsible for the high mass variation levels (CVm) observed in Fig. 11. Since twist tends to be higher at the thin places as compared to the thicker places, a yarn with high irregularity will lead to variations in dye/finish absorption. This is because higher twist regions tend to absorb less dyes/finishes, as compared to the thick places. Another implication of higher imperfections in the cotton yarn is the low yarn strength. Since a regular yarn is stronger than the irregular one, a reduction in the number of imperfections will improve yarn regularity leading to an improvement in yarn evenness and strength.

3.4.2.2 Effect of Count on Yarn Quality

By taking the mean values of the Tex 20 yarns (*a*, *b* and *c*) an average value for the Tex 20 yarn can be obtained. Similarly the mean of Tex 25 was obtained from the values of *d* and *e*. The same procedure was used to compute the mean values for the yarn quality characteristics. These values can be used to study the effect of yarn count (Tex) on yarn quality characteristics.

According to Figure 13 the tensile properties of cotton yarn (strength and elongation) increased with the increase in Tex, while yarn evenness (CVm) and imperfection (Thin, Thick Neps) decreased with the increase of Tex.

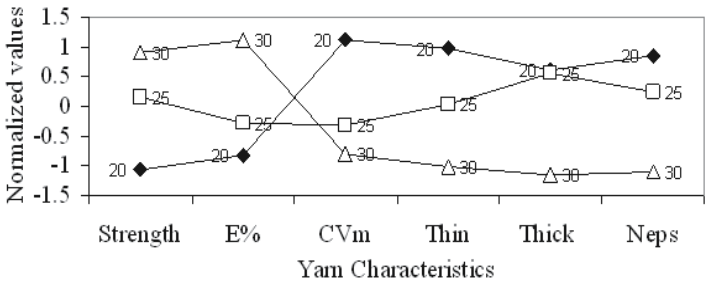


Fig. 13. Effect of count on yarn characteristics

Increase in yarn Tex increases the diameter of the yarn. The increase of imperfection as the count of the yarn decreases could have been caused by the decrease of the number of fibers in the yarn cross-section. As the number of fibers decreased the spinning limit was stretched due to the difficulties encountered as the fewer fibers try to combine together to form a yarn, hence leading to an increase in unevenness and imperfections.

3.4.2.3 Yarn Quality Analysis Using SOM

The yarn quality data comprised of 150 yarn samples each having 7 yarn quality characteristics (Tex, strength, elongation (E%), CVm, thin places/km (Thin), thick places/km (Thick) and neps/km (Nep)). SOM reduced the 150x7 yarn quality data to 11x6. The final quantization and topographic errors for the SOM maps were 0.814 and 0.014 respectively. The errors were reasonable enough and therefore the SOM maps can be used to study the yarn quality of the Kenyan manufactured cotton yarns.

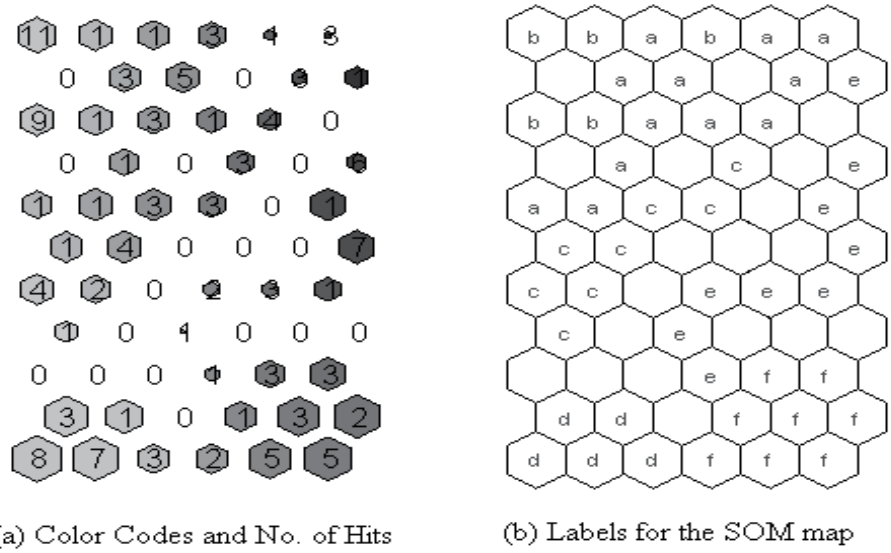


Fig. 14. SOM for yarn quality Characteristics

The SOM map with color codes and hits (number of samples per neuron) for yarn quality characteristics for the six cotton yarn samples is given in Fig. 14. The SOM algorithm was able to separate the fine counts (*a*, *b* and *c*) from the other three samples which were relatively coarser. Samples *c* and *e* were widely scattered in the map. This could be an indication of large variations in the quality of the yarn samples. Some members of sample *e* overlapped with samples *a* and *f*. This is a symptom of the presence of outliers in *e*, which implies that sample *e* has high irregularity.

Sample *e* was spun at a lower speed when compared to sample *d* (same Tex). Lower spinning speed should lead to lower irregularity, but in this case the converse has happened. Therefore the only logically explanation for the presence of high irregularity in sample *e* is the presence of quality control problems in the concerned factory. Samples *a*, *b*, *d* and *f* were less scattered in the SOM map. There was an overlap between samples *a* and *b*. This was expected, as the two samples had everything in common except the raw material (cotton lint) used to spin them.

The yarn quality characteristics for all the six yarn samples were visualized individually as shown in Figure 15. The SOM map for strength shows that the coarser counts (*d*, *e* and *f*) exhibited a higher strength as compared to the finer counts (*a*, *b* and *c*). It can also be noticed that the SOM map for count (Tex) is similar to that of strength. This is an indication of high correlation between the two characteristics. Similar results had been obtained in Fig. 13 and discussed in section 3.4.2.2. Sample *e* showed high CVm (bad from quality point of view).

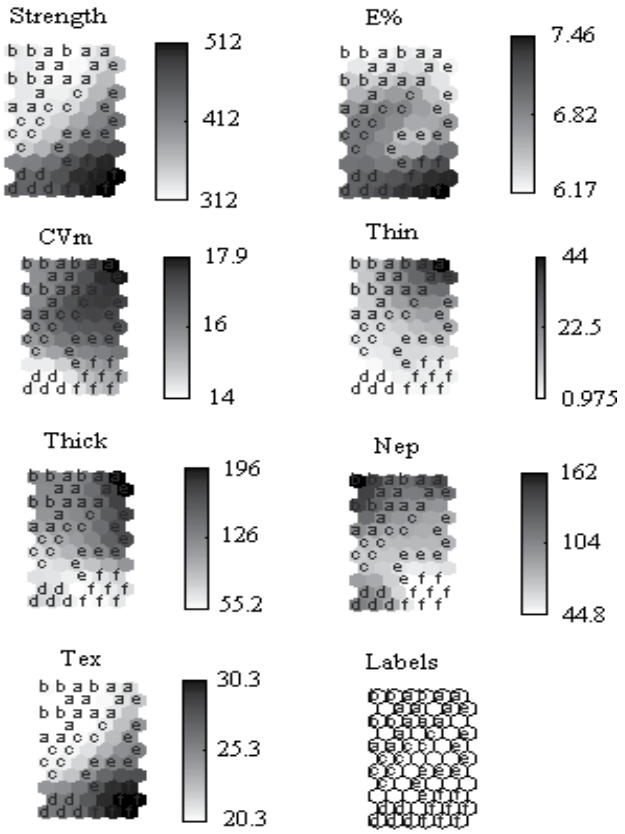


Fig. 15. The Quality Characteristics of Kenyan Cotton

Sample *d* showed the best yarn evenness while the other remaining samples were average. Samples *a*, *b* and *d* showed a higher number of neps. This was unexpected especially for sample *d* which showed low level of thin and thick places. Given that the three samples (*a*, *b* and *d*) were collected from the same factory, it could be an indication of poor neps control in the concerned factory.

4. Conclusion and recommendations

4.1 The trend of cotton growing in Kenya

Based on the collected samples and data the following conclusions can be made:

- The production of seed cotton in Kenya has been declining in the last two decades
- Kenyan farmers spend more on land preparations and at the expense of good yields spend less than expected on farm inputs like fertilizers and pesticides
- The break even cost of cotton growing in Kenya is US$ 0.31/ kg
- The farmers received US$ 0.29 for a kg of seed cotton
- There is a positive correlation between cotton yields and the costs of pesticides and weeding during cotton farming

Based on the results obtained in this study, it is hereby recommended that,

- The Kenyan cotton farmers should adapt mechanized farming methods instead of the current manual farming methods
- There is an urgent need for research to be done to establish the proper usage of fertilizers and other chemicals in the cotton growing industry
- SOM techniques can be used to analyze the high dimensional data consisting factors which affect cotton yield.

4.2 Classification of Kenyan cotton lint

A cotton clustering algorithm consisting of data visualization by SOM, data clustering by k-means technique and the classification of cotton lint clusters by PNN was designed and tested using Kenyan cotton lint. SOM successfully visualized the HVI data reducing it from 192x13 to 12x6, indicating that the Kenyan cotton lint can be partitioned into four clusters. Clustering the Kenyan lint using k-means clustering technique revealed that the Kenyan cotton lint tend to have higher values of micronaire, maturity, spinning consistency index, strength, uniformity, elongation and reflectance but lower values for yellowness, short fiber index and trash measurements values. The PNN algorithm successfully classified the cotton lint into four clusters (R-value of 1).

4.3 Classification of cotton bales using SOM techniques

A cotton bale classification model has been proposed and used to classify 2421 cotton bales. The model reduced the 2421x13 HVI high dimensional data into 18x13 grids, with a quantization error of 1.879 and a topographic error of 0.083, and initially identified 19 groups in the data. Three of the groups containing a total of 144 data failed the compactness test. The final classification of the 2421 cotton bales contains 16 groups of cotton bales having a total of 2277 bales and five sets of 144 bales containing outliers.

4.4 Use of SOM techniques to study cotton yarn quality

The study of cotton ringframe yarn manufactured in Kenya was undertaken. The manufactured yarn samples showed low (good) CV for yarn evenness and tensile

properties, while imperfection showed very high CV. This was an indication of high irregularity in the Kenyan manufactured yarns. The study of the effect of count on the yarn quality characteristics revealed that tensile properties increased with yarn count (Tex), while yarn evenness and imperfections decreased with increase in Tex. Use of Kohonen Self Organizing Maps (SOM) to study yarn quality characteristics produced the same results as reported above, hence SOM can also be used to analyze yarn quality characteristics. The SOM algorithm produced SOM maps which could easily show the behavior of the yarn quality characterizes for all the yarn samples. The SOM maps revealed that one of the factories had neps control problems.

This study on the quality of carded cotton yarn manufactured in Kenya, revealed that the Kenyan manufactured yarn tend to show high levels of irregularity due to a high number of imperfections. Bases on the results obtained in this study, it is hereby recommended that an investigation should be done to find out the cause of the high level of imperfections in the Kenyan cotton yarn so that the industry can enhance the quality of its cotton ring spun yarns.

5. References

Majumdar, A.; Majumdar, P. K. & Sarkar, B. (2004). Selection of cotton bales by spinning consistency index and micronaire using artificial neural networks, *Autex Research Journal*, Vol. 4, No. 1, March 2004, 1-8, ISSN: 1470-9589.

Jain, A.K. & Dubes, R.C. (1988). *Algorithms for Clustering Data*, Prentice-Hall, ISBN 013022278X, Englewood Cliffs, NJ,

Azzam, H. A. & Mohamed, S. T. (2005): Adapting and tuning quality management in spinning industry, *Autex Research Journal*, Vol. 5, No. 4, December 2005, 246-258, ISSN: 14709589

Bluman, A. G., (2004). *Elementary Statistics*, McGraw Hill, ISBN: 10-000557689X, New York, 5th Edition

Panagariya, A; Shah, S & Mishra, D (2001). Demand Elasticities in International Trade: Are They Really Low, *Journal of Development Economics,* Vol. 64, No. 2, April 2001, 313-342, ISSN: 0304-3878

Bolat, B. & Yildirim, T. (2003), Performance Increasing Methods for Probabilistic Neural Networks, *Pakistan Journal of Information and Technology*, Vol. 2 No. 3, 250- 255, ISSN: 1682-6027

D. F. Specht (1996), Probabilistic neural networks and general regression neural networks, In : Fuzzy Logic and Neural Network Handbook, C.H. Chen (Ed.), 301-344, McGraw-Hill Inc., ISBN:0-07-011189-8, New York

White, E. Jones, P. & Yankey, J (1999), Cotton Testing, In: *Progress in Textiles: Science & Technology, vol. 1,* V.K. Kothari, (Ed.), 45-91, ISBN: 81-901033-0-X, New Delhi

G.D. Joubert, A. Swanepoel, E.G. Eulitz, A.P.F. Cornelissen, and D. Venter, An interdisciplinary approach to the improvement of fibre quality, *Proc. of the 61st International Cotton Advisory Committee*, pp. 14-17, Cairo, Egypt, October 2002, ICAC, Washington

H. Demuth, and M. Beale, (2005), *Neural Network Toolbox User Guide*, chapt. 7, 8 and 9, The MathWorks, Natick

ICAC, Cotton: Review of world situation, Vol. 58, No.2, 2004, pp. 3-5, Washingtone

Frydrych, and M. Malgorzata, (2002), *Proc. of the 61st International Cotton Advisory Committee*, 7

Cling, J.-P., Razafindrakoto, M., & Roubaud, O. (2005), Export Processing Zones in Madagascar: Success Story under Threat, *World Development*, 2005, Vol 33, No. 5, 785–803, ISSN: 0305-750X

Jang, J.-S.R., & Mizutani, E. (1997), Unsupervised Learning and Other Neural Networks, In *Neuro-Fuzzy and Soft Computing*, Jang, J.-S.R.; Sun, C.-T, & Mizutani, E., (Ed.), 310-310, Prentice Hall, ISBN: 0-13-261066-3, Upper Saddle River, NJ

Klein, W. (1987): The Technology of Short Staple Spinning, In D.H. Stalder (Eds), *short staple spinning series*, The Textile Institute, Manchester. ISBN:1870812859.

Karim Hussein (2004), Importance of Cotton Production and Trade in West Africa, Proceedings of WTO African Regional Workshop on Cotton, Cotonou, Benin, March 2004.

Kenya Government, (1996a): Industrial Transformation to the Year 2020, Sessional paper No.2, Government Printers, Nairobi.

Kenya Government, 1996b: Rapid Industrialization for Sustained Development, Development plan 1997-2001, Government Printers, Nairobi.

Kenya Government, 1982, Report and Recommendations of the Working Party on Government Expenditures (*Ndegwa Report*), Government Printers, Nairobi.

Kothari, V.K., 1999, Yarn evenness and appearance, In Kothari, V.K. (Eds): *Progress in Textiles: Science & Technology, vol. 1: Testing & Quality Management*, IAFL Publications, New Delhi., ISBN-10: 0412540304

Mario Rodriguez Rico (2002), Impact of low prices on production practices and fiber quality-A farmer's perspective, *Proceedings of the International Cotton Advisory Committee Technology, Management and Processing for Quality Fiber, 61st Meeting*, Cairo, Egypt, October, 2002: 3-7.

McCreight, D. J.; Feil, R. W; Booterbaugh, J. H. and Backe, E. E. (1997): *Short Staple Yarn Manufacturing,* Carolina Academic Press, Durham, North Carolina, ISBN-10: 0890898537

Hagan, M.T.; Demuth, H,B & Beale, M, (2002), *Neural Network Design*, China Machine Press, Beijing, ISBN: 10-0971732108

Mwangi MI.M. (2004), The Textile Industry in Kenya, Nairobi, Government Printers, 2004

Kimenyi M.S. (2001). Agriculture: The Solution to Economic Problems and Poverty, Agriculture, Economic Growth and Poverty Reduction, Economic Management, Nairobi, 2001.

Weiss, N. A., (1997), *Introductory Statistics*, Addison-Wesley Publishing Company, Inc., New York, 4th Edition, pp. 41-178(1997), *ISBN*: 0201539292

Oxtoby E, (1987): *Spun Yarn Technology*, Butterworth & Co (Publishers) Ltd, London, ISBN: 10-0408014644

Uster, (2007): *Uster Statistics 2007*, Uster Technologies AG, Wilstrasse.

Kohonen, T. (1997), *Self-Organizing Maps*, pp. 85-144, Springer-Verlag Berlin/Heidelberg, 1997, ISBN 3-540-67921-9

Vesanto, J.; Himberg, J. Alhoniemi, E. and ParhanKangas, J. (2000): *SOM Toolbox for Matlab 5*, Helsinki University of Technology, ISBN: 951-22-4951-0

Si, J.; Nelson B.J. & Runger G.C. (2003), Artificial Neural Network Models for Data Mining, In: The Handbook of Data Mining, Ye, N., (Ed.), LEA Publishers, 2003, London, ISBN 0-8058-4081-8, pp. 42-65

Ikiara, M.M. and L.K. Ndirangu (2004). *Prospects for Kenya's Clothing Exports Under AGOA after 2004*, Chapter in Clothing and Footwear in Africa's Industrialisation, edited by D. McCormick and C. Rogerson. African Institute of South African Publishers.

Part 8

Design and Application of Novel Variants of SOM

30

Associative Self-Organizing Map

Magnus Johnsson[1], Max Martinsson[1], David Gil[2] and Germund Hesslow[3]
[1]*Lund University Cognitive Science*
[2]*Computing Technology and Data Processing, University of Alicante*
[3]*Department of Experimental Medical Science, Lund*
[1,3]*Sweden*
[2]*Spain*

1. Introduction

There is not enough genetic information to specify the connectivity of the brain in detail (Miikkulainen et al., 2005). The total number of neurons in the neocortex of an adult human brain is estimated to be about 28 billion and the number of connections (synapses) between them and to other cells in the brain to more than 100 trillion (Mountcastle, 1997). In comparison, the human genome only contains about 3 billion base pairs (Consortium, 2004). Thus a reasonable view is that the cortex is not directly specified genetically but constructed by input-driven self-organization (Miikkulainen et al., 2005).

The self-organizing process works by using sensory input to adjust the networks organization instead of specifying all connections in advance. The form of self-organization that seems to be active in the cortex gives rise to a special property called topological mapping. This means that the neurons that are activated for similar sensory inputs are found close to each other. The properties self-organization and topology preservation are caught in the Self-Organizing Map (SOM) (Kohonen, 1988), which shares many features with brain maps (Kohonen, 1990).

However, the cortex consists of many brain maps and different parts of the cortex obviously interact. For example, different sensory modalities interact with each other. A dramatic illustration of this can be seen in the McGurk-MacDonald effect. If you hear a person making the sound /ba/ but the sound is superimposed on a video recording on which you do not see the lips closing, you may hear the sound /da/ instead (McGurk & MacDonald, 1976). The neural mechanisms underlying such interaction between different sensory modalities are not known but recent evidence suggests that different primary sensory cortical areas can influence each other.

Interaction between sensory modalities may also be important for internal simulation of perceptions. An idea that has been gaining popularity in cognitive science in recent years is that higher organisms are capable of simulating perception. In essence, this means that the perceptual processes normally elicited by some ancillary input can be mimicked by the brain (Hesslow, 2002). There is now a large body of evidence supporting this contention. For instance, several neuroimaging experiments have demonstrated that activity in visual cortex when a subject imagines a visual stimulus resembles the activity elicited by a corresponding ancillary stimulus (for a review of this evidence see e.g. (Kosslyn et al., 2001); for a somewhat different interpretation, see (Bartolomeo, 2002).

A critical question here is how simulated perceptual activity might be elicited. One possibility is that signals arising in the frontal lobe in anticipation of consequences of incipient actions are sent back to sensory areas (Hesslow, 2002). Another possibility is that perceptual activity in one sensory area can influence activity in another.
Inspired by these findings we suggest that in a multimodal perceptual model, the subsystems of different sensory modalities should co-develop and be associated with each other. This means that suitable activity in some modalities that for the moment receive input should, at least to some degree, elicit appropriate activity in other sensory modalities as well. This provides an ability to activate the subsystem for a modality even when its sensory input is limited or nonexistent as long as there is activity in subsystems for other modalities.
Another probable ability of the brain is to elicit continued and reasonable activity in different perceptual subsystems in the absence of input, i.e. an ability to internally simulate sequences of perceptions as proposed in the neuroscientific simulation hypothesis (Hesslow, 2002). This means an ability to elicit activity patterns that are normally subsequent to the present activity pattern in a subsystem even when sensory input is absent. It would also imply an ability to anticipate future sequences of perceptions that normally follows a certain perception within a modality, but also over different modalities if the modalities have co-developed and are associated. For example, a gun seen to be fired from a long distance, would yield an anticipation of a bang to follow soon.
This chapter presents a novel variant of the SOM. This variant of the SOM is called the Associative Self-Organizing Map (A-SOM) and we think it would be suitable in models that catch phenomena like those sketched above.
The A-SOM is similar to the SOM and develops a representation of its input space, but in addition it also learns to associate its activity with the (possibly time delayed) activities of an arbitrary number of other neural networks, or its own earlier activity (which makes it into an unsupervised recurrent neural network).
The A-SOM differs from earlier attempts to build associated maps such as the Adaptive Resonance Associative Map (Tan, 1995) and Fusion ART (Nguyen et al., 2008) in that all layers (or individual networks) share the same structure and uses topologically arranged representations. Unlike ARTMAP, the A-SOM also allows associations to be formed in both directions (Carpenter et al., 1992).
The most similar existing unsupervised recurrent neural network is the Recursive SOM that feeds back its activity together with the input for the next iteration (Voegtlin, 2002). The Recursive SOM is similar but not equivalent to the A-SOM and lacks ability to associate with the activity of other neural networks. Other less similar examples are the Temporal Kohonen Map (Chappell & Taylor, 1993), the Recurrent Self-Organizing Map (Varsta et al., 1997) and the Merge SOM (Strickert & Hammer, 2005).
The chapter both summarizes our previous work (Johnsson & Balkenius, 2008; Johnsson et al., 2009a;b) with the A-SOM and adds new results and insights. It describes the A-SOM in detail and its use in the modelling of cross-modal expectations and in the modelling of internal simulation.

2. Associative self-organizing map

The A-SOM is based on the ordinary SOM and thus finds a representation of its input space. In addition it also learns to associate its activity with (possibly delayed) additional ancillary inputs. These ancillary inputs could be the activities of a number of external SOMs or A-SOMs, or the earlier activity of the A-SOM itself. It consists of a grid of neurons with

a fixed number of neurons. Each neuron has multiple sets of weights, one for main input (which is similar to the input of an ordinary SOM) and one for each ancillary input. All neurons receive both main input (e.g. from a sensor), and ancillary inputs (e.g. the activity in associated representations of other sensory modalities or the A-SOMs activity from previous iterations). Each neuron calculates activities for its main input and for each ancillary input. The main input activity is calculated in a way similar to the ordinary SOM, with dot product as the similarity measure. Also the adaptation of the weights corresponding to the main input are calculated as in an ordinary SOM, i.e. so that the neuron with the highest main activity and the neurons in its vicinity are adjusted. The ancillary activities of a neuron are calculated using dot product and are adjusted by the delta rule to approach the main activity. The total activity of a neuron is calculated by averaging the main activity and the ancillary activities. By connecting the total activity of the A-SOM back to itself as an ancillary input with a time delay the A-SOM is turned into a recurrent A-SOM able to learn sequences. This is so because then the ancillary weights will have learned to evoke activity based on the previous activity in the A-SOM.
Formally the A-SOM consists of an $I \times J$ grid of neurons with a fixed number of neurons and a fixed topology. Each neuron n_{ij} is associated with $r+1$ weight vectors $w_{ij}^a \in R^n$ and $w_{ij}^1 \in R^{m_1}, w_{ij}^2 \in R^{m_2}, \ldots, w_{ij}^r \in R^{m_r}$. All the elements of all the weight vectors are initialized by real numbers randomly selected from a uniform distribution between 0 and 1, after which all the weight vectors are normalized, i.e. turned into unit vectors.

At time t each neuron n_{ij} receives $r+1$ input vectors $x^a(t) \in R^n$ and $x^1(t-d_1) \in R^{m_1}, x^2(t-d_2) \in R^{m_2}, \ldots, x^r(t-d_r) \in R^{m_r}$ where d_p is the time delay for input vector $x^p, p = 1,2,\ldots,r$.

The main net input s_{ij} is calculated using the standard cosine metric

$$s_{ij}(t) = \frac{x^a(t) \cdot w_{ij}^a(t)}{||x^a(t)||||w_{ij}^a(t)||}, \tag{1}$$

The activity in the neuron n_{ij} is given by

$$y_{ij}(t) = \left[y_{ij}^a(t) + y_{ij}^1(t) + y_{ij}^2(t) + \ldots + y_{ij}^r(t) \right] / (r+1) \tag{2}$$

where the main activity y_{ij}^a is calculated by using the softmax function Bishop (1995)

$$y_{ij}^a(t) = \frac{\left(s_{ij}(t) \right)^m}{\max_{ij} \left(s_{ij}(t) \right)^m} \tag{3}$$

where m is the softmax exponent.
The ancillary activity $y_{ij}^p(t), \quad p = 1,2,\ldots,r$ is calculated by again using the standard cosine metric

$$y_{ij}^p(t) = \frac{x^p(t-d_p) \cdot w_{ij}^p(t)}{||x^p(t-d_p)||||w_{ij}^p(t)||}. \tag{4}$$

The neuron c with the strongest main activation is selected:

$$c = \arg\max_{ij} y^a_{ij}(t) \tag{5}$$

The weights w^a_{ijk} are adapted by

$$w^a_{ijk}(t+1) = w^a_{ijk}(t) + \alpha(t) G_{ijc}(t) \left[x^a_k(t) - w^a_{ijk}(t) \right] \tag{6}$$

where $0 \leq \alpha(t) \leq 1$ is the adaptation strength with $\alpha(t) \to 0$ when $t \to \infty$. The neighbourhood function $G_{ijc}(t) = e^{-\frac{||r_c - r_{ij}||}{2\sigma^2(t)}}$, where $r_c \in R^2$ and $r_{ij} \in R^2$ are location vectors of neurons c and n_{ij}, is a Gaussian function decreasing with time.

The weights $w^p_{ijl}, p = 1,2,\ldots,r$, are adapted by

$$w^p_{ijl}(t+1) = w^p_{ijl}(t) + \beta x^p_l(t - d_p) \left[y^a_{ij}(t) - y^p_{ij}(t) \right] \tag{7}$$

where β is the adaptation strength.
All weights $w^a_{ijk}(t)$ and $w^p_{ijl}(t)$ are normalized after each adaptation.

3. Modelling cross-modal expectations

3.1 Associating the A-SOM with two ancillary SOMs

We have tested the A-SOM in a model of cross-modal expectations (Johnsson et al., 2009a). In this experiment we connected an A-SOM to two ancillary SOMs and trained all three neural networks with a set of 10 samples, Fig. 1. This set was constructed by randomly generating 10 points with a uniform distribution from a subset $s = \{(x,y) \in R^2; 0 \leq x \leq 1, 0 \leq y \leq 1\}$ of the plane, Fig. 2, left. The selected points were then mapped to a subset of R^3 by adding a third constant element of 0.5, yielding a training set of three-dimensional vectors. The reason for this was that a Voronoi tessellation of the plane was calculated from the generated points to later aid in the determination of were new points in the plane were expected to invoke activity in the A-SOM. To make this Voronoi tessellation, which is based on a Euclidian metric, useful for this purpose with the A-SOM, which uses a metric based on dot product, the set of points in the plane has to be mapped so that the corresponding position vectors after normalization are unique. One way to accomplish such a mapping is by adding a constant element to each vector. The result of this is that each vector will have a unique angle in R^3. We chose the value 0.5 for the constant elements to maximize the variance of the angles in R^3.
The A-SOM was connected to two SOMs (using the same kind of activation as the main activation in the A-SOM, i.e. dot product with softmax activation) called SOM 1 and SOM 2, and thus also received their respective activities as associative input, see Fig. 1. The A-SOM, SOM 1 and SOM 2 were then simultaneously fed with samples from the training set, during a training phase consisting of 20000 iterations. The two SOMs and the A-SOM could as well be fed by samples from three different sets, always receiving the same combinations of samples from the three sets (otherwise the system could not learn to associate them). This could be seen as a way of simulating simultaneous input from three different sensory modalities when an animal or a robot explores a particular object. Each of the three representations, the A-SOM and the two SOMs, consists of 15×15 neurons. The softmax exponent for each of them were set to 1000. Their learning rate $\alpha(0)$ was initialized to 0.1 with a learning rate decay of 0.9999 (i.e. multiplication of the learning rate with 0.9999 in each iteration), which means the minimum learning rate, set to 0.01, will be reached at the end of the 20000 training iterations. The neighbourhood radius, i.e. σ of the neighbourhood function $G_{ijc}(t)$ in eq. (6),

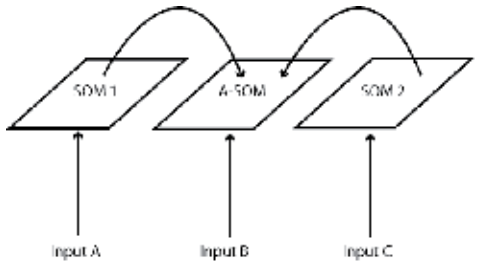


Fig. 1. Schematic depiction over the connections between the two SOMs and the A-SOM in the model of cross-modal expectations. The test system consists of three subsystems, which develop representations of sample sets from three input spaces (for simplicity we use the same input set for all three representations in this study). One of the representations (the A-SOM) also learns to associate its activity with the simultaneous activities of the two SOMs. This means proper activity can be invoked in the A-SOM of the fully trained system even if it does not receive any ordinary input. This is similar to cross-modal activation in humans, e.g. a tactile perception of an object that invokes an internal visual imagination of the same object.

was initialized to 15 for all three representations and shrunk to 1 during the 20000 training iterations by using a neighbourhood decay of 0.9998 (i.e. multiplication of the neighbourhood radius with 0.9998 in each iteration). All three representations used plane topology when calculating the neighbourhood. The β for the associative weights in the A-SOM was set to 0.35.

After training the system was evaluated by feeding it with samples from the training set again to one, two or all three representations in all possible combinations. When a representation did not receive any input it was fed with null vectors instead (thus simulating the input of

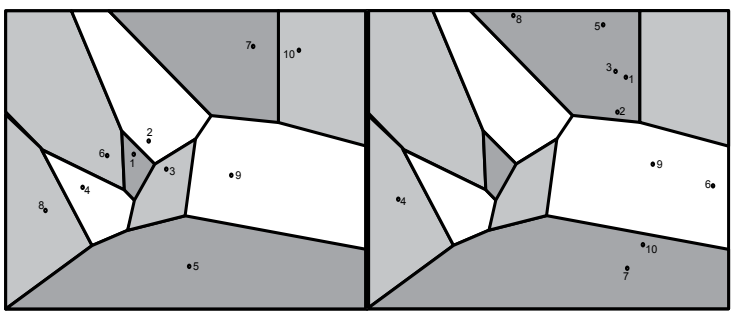


Fig. 2. Left: The Voronoi tessellation of the points used when constructing the training set used for the A-SOM and the two SOMs. This set was constructed by randomly generating 10 points from a subset of R^2 according to a uniform distribution. To make this Voronoi tessellation, which is based on a Euclidian metric, valid as a measure of proximity the training set had to be transformed by addition of a constant element to each sample vector. This is because the A-SOM using a dot product based metric and normalizing its input would consider all position vectors of a particular angle equal. By adding a constant element each point in the plane becomes a position vector in R^3 with a unique angle. Right: The same Voronoi tesselation but with the points used in the generalization test depicted. Also this set was mapped to a new set in R^3 by addition of a third constant element to each sample vector, and for the same reason as for the samples in the training set.

no signal from sensors of the modality of that representation). The centers of activity in the A-SOM as well as in the two SOMs were recorded for all these tests.
The result was evaluated by using the training set on the fully trained system. First we recorded the centers of activation in the A-SOM when fed by main input from the training set only (i.e. the two SOMs were fed with null vectors) and the centers of activation in the two SOMs. Then we calculated Voronoi tessellations for the centers of activation in all three representations (Fig. 3, uppermost row) to see if they could separate the samples and in particular if the A-SOM could separate the samples when fed by the activity of one or both of the SOMs only. If the center of activation for a particular sample in the training set were located in the correct Voronoi cell, this is considered as a successful recognition of the sample, because this means the center of activation is closer to the center of activation of the same object than to the center of activation of any other sample in the training set when the A-SOM is fed by main input only like an ordinary SOM. By comparing the Voronoi tessellations of the A-SOM and the two SOMs, Fig. 3, and the Voronoi tessellation of the plane for the training set, Fig. 2, we can see that the ordering of the Voronoi cells for the training set are to a large extent preserved for the Voronoi cells for the centers of activation in the A-SOM and the two SOMs. In Fig. 3 we can also see that all, i.e. 100% of the training samples are recognized in the A-SOM as long as at least one of the three representations receives input.

3.1.1 Generalization

To test if the system was able to generalize to a new set of samples, which it had not been trained with, we constructed another set of 10 samples with the same method as for the training set. This generalization test set was used as input to the two SOMs and the A-SOM, i.e. each of these representations received the same sample simultaneously (or a null vector).
The generalization ability of the system was evaluated by feeding it with samples from the generalization set to one, two or all three representations in all possible combinations. When a representation did not receive any input it was fed with null vectors instead. The centers of activity in the A-SOM as well as in the two SOMs were recorded for all these tests.
The result was evaluated by now using the generalization set on the fully trained system. We recorded the centers of activation in the A-SOM when each of the SOMs were the only recipient of input, when both SOMs received input, when each of the SOMs and the A-SOM received input, when all three representations received input, and when only the A-SOM received input. As before a representation which did not receive input received null vectors (signifying the lack of sensory registration for that modality). We then looked at in which Voronoi cell the centre of activation was located in the A-SOM and in the SOMs for each sample of the generalization set. When a generalization sample belongs to the Voronoi cell for sample $k = 1,2,\ldots,10$ of the training set, see Fig. 2, and its activation in the A-SOM or one of the SOMs is located in the Voronoi cell for the centre of activation for the same training sample, see Fig. 3, then we consider the centre of activation for the generalization sample to be properly located and we consider it to be successfully generalized.
Leftmost in the upper row of Fig. 3 we can see that the centers of activation for all the generalization samples besides sample 8 is within the correct Voronoi cell in the A-SOM when it receives main input only. However that sample 8 is outside, and barely so, the correct Voronoi cell is probably not an indication that it is incorrect because the A-SOM consists of 225 neurons and is not a continuous surface but a discretized representation.
In the middle of the upper row of Fig. 3 we can see that all centers of activation for the generalization samples are correctly located in SOM1 besides 1 and 6 which are on the

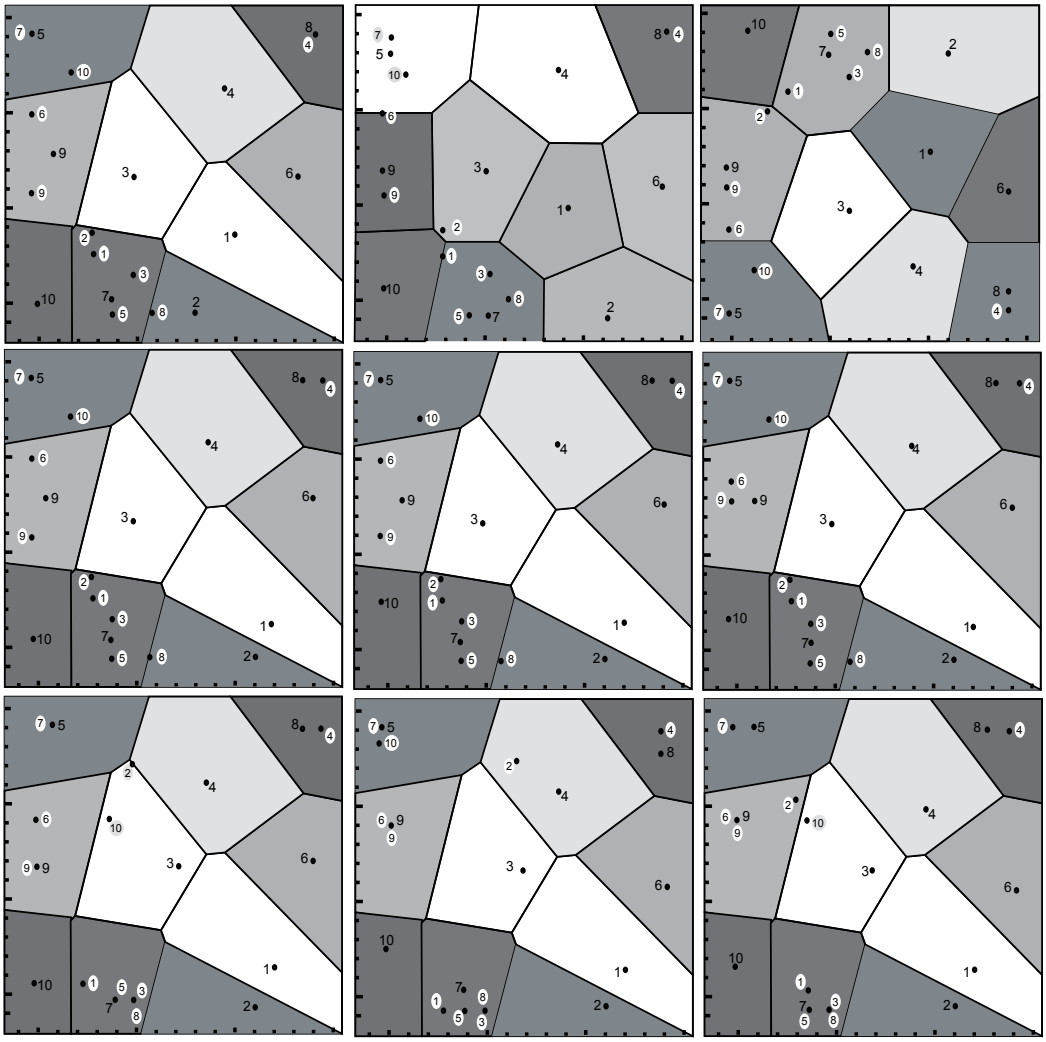


Fig. 3. The center of activation for different constellations of input to the fully trained system in the A-SOM and in the two SOMs. The centers of activation for the generalization samples have been written on circles with a contrasting colour to differentiate them from the training samples. Upper row left: The A-SOM when only main input to the A-SOM is received. The Voronoi tessellation for these centers of activation has also been drawn. This is also true for the other images in this figure depicting activations in the A-SOM. Upper row middle: The SOM1 with the Voronoi tesselation for the training set drawn. Upper row right: The SOM2 with the Voronoi tesselation for the training set drawn. Middle row left: The A-SOM receiving main input and the activity of SOM1. Middle row middle: The A-SOM when receiving main input and the activity of SOM2. Middle row right: The A-SOM when receiving main input and the activities of SOM1 and SOM2. Lower row left: The A-SOM when receiving the activity of SOM1 only. Lower row middle: The A-SOM when receiving the activity of SOM2 only. Lower row right: The A-SOM receiving the activities of SOM1 and SOM2.

border to the correct Voronoi cell (but this should probably not be considered an indication of incorrectness for the same reason as mentioned above), and 2 which is located close to the correct Voronoi cell.
Rightmost of the upper row of Fig. 3 we can see that all centers of activation for the generalization samples are correctly located in SOM2 besides 2, which is located close to the correct Voronoi cell.
Leftmost in the middle row of Fig. 3 we can see that the centers of activation for all the generalization samples besides sample 8 (which should probably not be considered an indication of incorrectness for the same reason as mentioned above) is within the correct Voronoi cell in the A-SOM when it receives main input as well as the activity of SOM1 as input.
In the middle of the middle row of Fig. 3 we can see that the centers of activation for all the generalization samples besides sample 8 (which should probably not be considered an indication of incorrectness for the same reason as mentioned above) is within the correct Voronoi cell in the A-SOM when it receives main input as well as the activity of SOM2 as input.
Rightmost of the middle row of Fig. 3 we can see that the centers of activation for all the generalization samples besides sample 8 (which should probably not be considered an indication of incorrectness for the same reason as mentioned above) is within the correct Voronoi cell in the A-SOM when it receives main input as well as the activities of both SOM1 and SOM2 as input.
Leftmost of the lower row of Fig. 3 we can see that the centers of activation for all the generalization samples besides sample 2 and 10, i.e. 80%, is within the correct Voronoi cell in the A-SOM when it receives the activity of SOM1 as its only input.
In the middle of the lower row of Fig. 3 we can see that the centers of activation for all the generalization samples besides sample 2, i.e. 90%, is within the correct Voronoi cell in the A-SOM when it receives the activity of SOM2 as its only input.
Rightmost of the lower row of Fig. 3 we can see that the centers of activation for all the generalization samples besides sample 2 and 10, i.e. 80%, is within the correct Voronoi cell in the A-SOM when it receives the activities of SOM1 and SOM2 as its only input.
In Fig. 4 we can see a graphical representation of the activity in the two SOMs as well as total, main and ancillary activities of the A-SOM while receiving a sample from the generalization set. The lighter an area is in this depiction, the higher the activity is in that area.

3.1.2 Discussion

The ability of the A-SOM proved to be good in this experiment, with 100% accuracy with the training set and about 80-90% accuracy in the generalization tests, depending on which constellation of inputs which was provided to the system. It was also observed that the generalization in the ordinary SOMs was not perfect. If this had been perfect the generalization ability would probably be even better. This is probably a matter of optimizing the parameter settings.
It is interesting to speculate, and later test, whether there are any restrictions on the sets that are used as input to the different SOMs and A-SOMs in this kind of system. A reasonable guess would be that to learn to associate the activity arising from the training sets impose no restrictions on the training sets, but when it comes to generalization there would probably be one restriction. The restriction is that there should probably need to exist a topological function between the different input spaces so that the sequences of input samples from the

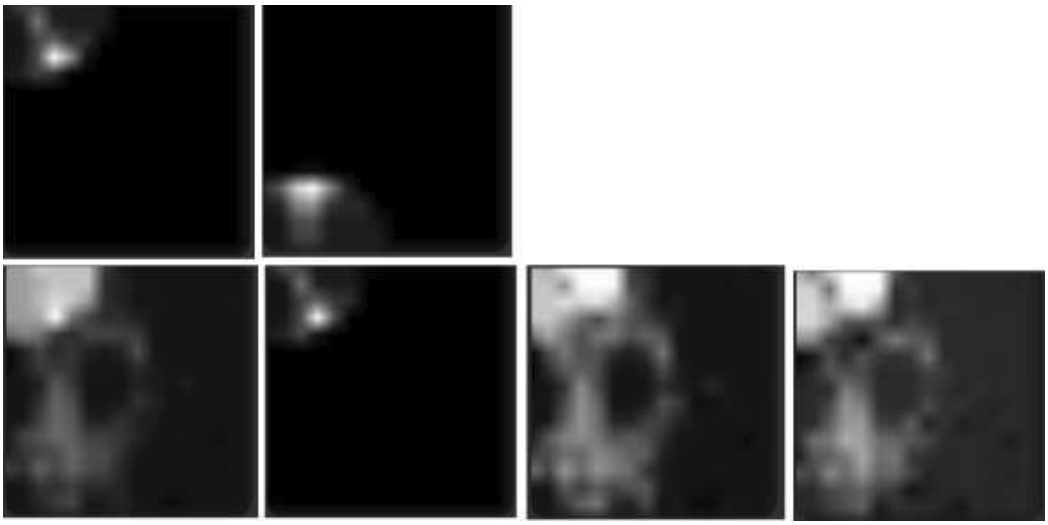


Fig. 4. Activations at a moment in the simulation. The lighter an area is in this depiction, the higher the activity is in that area. Upper row left: The activity in SOM1. Upper row right: The activity in SOM2. Lower row left: The total activity in the A-SOM. Lower row, the second image from the left: The main activity in the A-SOM. Lower row, the third image from the left: The ancillary activity in the A-SOM due to the activity in SOM1. Lower row right: The ancillary activity in the A-SOM due to the activity in SOM2.

different input spaces will invoke traces of activities over time in their respective SOM or A-SOM that in principle would be possible to map on each other by using only translations, rotations, stretching and twisting. Otherwise the generalization would be mixed up at least partially. The same would be true if the parameter setting implies the development of fragmentized representations.

3.2 Associating SOM representations of haptic submodalities

We have also tested the A-SOM together with a couple of real sensors (texture/hardness)(Johnsson & Balkenius, 2008). This system developed representations for texture as well as hardness and could trigger an activation pattern in the other modality, which resembled the pattern of activity the object would yield if explored with the sensor for this other modality.

3.2.1 Sensors in the experiment

The system employs two sensors (Fig. 5) developed at Lund University Cognitive Science (LUCS). One of these sensors is a texture sensor and the other is a hardness sensor.

The texture sensor consists of a capacitor microphone with a tiny metal edge mounted at the end of a moveable lever, which in turn is mounted on a servo. When exploring a material the lever is turned by the servo, which moves the microphone with the attached metal edge along a curved path in the horizontal plane. This makes the metal edge slide over the explored material, which creates vibrations in the metal edge with frequencies that depend on the textural properties of the material. The vibrations are transferred to the microphone since there is contact between it and the metal edge. The signals are then sampled and digitalized by a NiDaq 6008 (National Instruments) and conveyed to a computer via a USB-port. The Fast

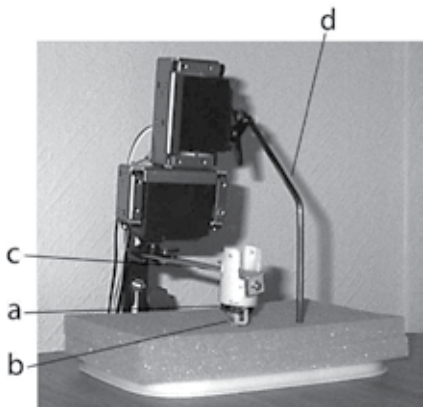


Fig. 5. The texture and hardness sensors while exploring a piece of foam rubber. The texture sensor consists of a capacitor microphone (a) with a metal edge (b) mounted at the end of a moveable lever (c), which in turn is mounted on a servo. The hardness sensor consists of a stick (d) mounted on a servo. The servo belonging to the hardness sensor contains a variable resistor that provides a measure of the turning of the servo, and thus the displacement of the stick, which is proportional to the compression of the explored material. The actuators are controlled via a SSC-32 controller board (Lynxmotion Inc.). The measure of the resistance of the variable resistor in the servo for the hardness sensor and the microphone signal of the texture sensor are digitalized using a NiDaq 6008 (National Instruments) and conveyed to the computer via a USB-port.

Fourier Transform (FFT) algorithm is then applied to the input, thus yielding a spectrogram of component frequencies.
The hardness sensor consists of a stick mounted on a servo. During the exploration of a material the servo tries to move to a certain position, which causes a downward movement of the connected stick at a constant pressure. In the control circuit inside the servo there is a variable resistor that provides the control circuit with information whether the servo has reached the wanted position or not. In our design, we measure the value of this variable resistor at the end of the exploration of the material and thus get a measure of the end position of the stick in the exploration. This end position is proportional to the compression of the explored material. The value of the variable resistor is conveyed to a computer and represented in binary form.
The actuators for both the sensors are controlled from the computer via a SSC-32 controller board (Lynxmotion Inc.). The software for the system presented in this paper is developed in C++ and runs within the Ikaros system (Balkenius et al., 2010). Ikaros provides an infrastructure for computer simulations of the brain and for robot control.

3.2.2 Exploration of objects

Our system was trained and tested with two sets of samples. One set consists of 40 samples of texture data and the other set consists of 40 samples of hardness data. These sets have been constructed by letting both the sensors simultaneously explore each of the eight objects described in Table 1 five times.
During the hardness exploration of an object the tip of the hardness sensor stick (Fig. 5 d) is pressed against the object with a constant force and the displacement is measured.

Label	Object	Estimated Hardness	Estimated Texture
a	Foam Rubber	Soft	Somewhat Fine
b	Hardcover Book	Hard	Shiny
c	Bundle of Paper	Hard	Fine
d	Cork Doily	Hard	Rough
e	Wood Doily	Hard	Fine
f	Bundle of Denim	Soft	Somewhat Fine
g	Bundle of Cotton Fabric	Soft	Somewhat Fine
h	Terry Cloth Fabric	Soft	Rough

Table 1. The eight objects used in the experiment. The objects a-h were used both for training and testing. The materials of the objects are presented and they are subjectively classified as either hard or soft by the authors. A rough subjective estimation of their textural properties is also provided.

The exploration with the texture sensor is done by letting its lever (Fig. 5 c) turn 36 degrees during one second. During this movement the vibrations from the metal edge (Fig. 5 b) slid over the object are recorded by the microphone (Fig. 5 a) mounted at the end of the stick.
The output from the texture sensor from all these explorations has then been written to a file after the application of the FFT. Likewise the output from the hardness sensor has been written to a file represented as binary numbers. The hardness samples can be considered to be binary vectors of length 18 whereas the texture samples can be considered to be vectors of length 2049. The eight objects have various kinds of texture and can be divided into two groups, one with four rather soft objects and one with four rather hard objects. During the exploration, the objects were fixed in the same location under the sensors.

3.2.3 Experiment

Our system (Fig. 6) is bimodal and consists of two monomodal subsystems (hardness and texture), which develop monomodal representations (A-SOMs) that are associated with each other. The subsystem for hardness uses the raw sensor output from the hardness sensor, represented as a binary number with 18 bits and conveys it to an A-SOM with 15×15 neurons. After training, this A-SOM will represent the hardness property of the explored objects.
In the subsystem for texture, the raw sensor output from the texture sensor is transformed by a FFT module into a spectrogram containing 2049 frequencies, and the spectrogram which is represented by a vector, is in turn conveyed to an A-SOM with 15×15 neurons. After training, this A-SOM will represent the textural properties of the explored objects.
The two subsystems are coupled to each other in that their A-SOMs also receive their respective activities as ancillary input.
Both A-SOMs begun their training with the neighbourhood radius equal to 15. The neighbourhood radius was decreased at each iteration by multiplication with 0.998 until it reached the minimum neighbourhood size 1. Both A-SOMs started out with $\alpha(0) = 0.1$ and decreased it by multiplication with 0.9999. β where set to 0.35 for both A-SOMs.
The system was trained with samples from the training set, described in the previous section, by 2000 iterations before evaluation.

3.2.4 Results and discussion

The results of the experiment with the texture/hardness system are depicted in Fig. 7. The 6 images depict the centres of activation in the A-SOMs when the fully trained system was

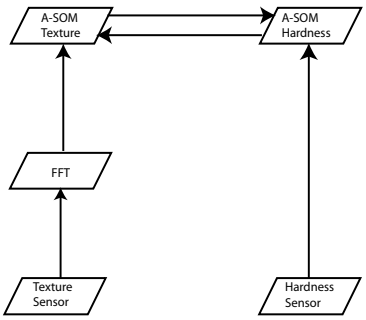


Fig. 6. Schematic depiction over the architecture of the haptic hardness and texture perception system. The system consists of two monomodal subsystems, which develop monomodal representations (A-SOMs) of hardness and texture that learn to associate their activities. The hardness subsystem uses the raw sensor output from the hardness sensor as input to an A-SOM, which finds a representation of the hardness property of the explored objects. The texture subsystem transforms the raw sensory data by the aid of a FFT module and then forwards it to another A-SOM, which finds a representation of the textural properties of the explored objects. The two A-SOMs learn to associate their respective activities.

tested with the test set (described above) constructed with the aid of the objects a-h in Table 1. Images 7 A, 7 B and 7 C correspond to the texture representing A-SOM. Likewise the images 7 D, 7 E and 7 F correspond to the hardness representing A-SOM. Each cell in an image represents a neuron in the A-SOM. In the images 7 A, 7 B, 7 D and 7 E there are black circles in some of the cells. This means that the corresponding neurons in the A-SOM are the centre of activation for one or several of the samples in the test set. The centres of activation from the samples in the test set corresponding to each object in Table 1 when only main input was provided have been encircled in 7 A and 7 D to show where different objects are mapped in the A-SOMs. Main input should be understood as texture input for the texture representing A-SOM, and hardness input for the hardness representing A-SOM. The encirclings are also present in the other four images. This is so because we want to show how the A-SOMs are activated when there are both main and ancillary input provided to the system (7 B and 7 E), and when there are only ancillary input provided (7 C and 7 F). Ancillary input should be understood as hardness input in the case of the texture representing A-SOM, and as texture input in the case of the hardness representing A-SOM.
Fig. 7 A depicts the texture representing A-SOM in the fully trained system when tested with the test set (only main texture input). As can be seen, most objects are mapped at separate sites in the A-SOM (c, d, e, f, h). There are some exceptions though, namely a, b and g. So the system is able to discriminate between individual objects when provided with main input only, although not perfectly.
The hardness representing A-SOM in the fully trained system when tested with the test set (only main hardness input), depicted in Fig. 7 D, also maps different objects at different sites in the A-SOM but not as good as the texture representing A-SOM. The hardness representing

A-SOM recognizes b, f and h perfectly and blurs the other more or less. However, the hardness representing A-SOM perfectly discriminates hard from soft objects.
When the texture representing A-SOM receives main texture input as well as ancillary hardness input (as can be seen in Fig. 7 B) its activations are very similar to those in Fig. 7 A. Likewise when the hardness representing A-SOM receives main hardness input as well as ancillary texture input (as can be seen in Fig. 7 E) its activations are very similar to those in Fig. 7 D.
Fig. 7 C depicts the activations in the texture representing A-SOM when it receives only ancillary hardness input. As can be seen this ancillary hardness input very often triggers an activity similar to the activity following main texture input. Likewise Fig. 7 F depicts the activity in the hardness representing A-SOM when it receives only ancillary texture input. Even in this case the ancillary input very often triggers an activity similar to the activity following main input. This means that when just one modality in the system receives input, this can trigger activation in the other modality similar to the activation in that modality when receiving main input. Thus an object explored by both sensors during training of the system can trigger a more or less proper activation in the representations of both modalities even when it can be explored by just one sensor during testing. However, as can be seen in Fig. 7 C and Fig. 7 F, the activity triggered solely by ancillary input does not map every sample properly. The worst cases are the objects c, d and g in the texture representing A-SOM (Fig. 7 C) and the objects a, b and g in the hardness representing A-SOM (Fig. 7 D). As can be seen in Fig. 7 D, the objects c, d and g are not distinguishable in the hardness representing A-SOM, and the objects a, b and g are not distinguishable in the texture representing A-SOM (Fig. 7 A). Thus the ancillary activity patterns for these objects are overlapping and the receiving A-SOM cannot be expected to learn to map these patterns correctly even if the objects where well separated by the A-SOM when it received main input.

4. Modelling internal simulation

In this section we will focus on the use of the A-SOM as a memory for perceptual sequences. Theses experiments was accomplished by using the total activity of the A-SOM as time-delayed ancillary input to itself.

4.1 A bimodal system

We have set up a bimodal model consisting of two A-SOMs (Fig. 8) and tested its ability to continue with reasonable sequences of activity patterns in the two A-SOMs in the absence of any input. This could be seen as an ability to internally simulate expected sequences of perceptions within a modality likely to follow the last sensory experience while simultaneously elicit reasonable perceptual expectations in the other modality.
One of the A-SOMs, the A-SOM A, is a recurrent A-SOM, and one, the A-SOM B, is an ordinary A-SOM without recurrent connections. A-SOM A is connected to A-SOM B (see Fig. 8). Thus the activity in A-SOM A will elicit associated activity in A-SOM B.
To test the model a set of 10 training samples were constructed (Fig. 9 left). This was done in the same way as when testing the model of cross-modal expectations described in section 3.1 above. I.e. by randomly generating 10 points with a uniform distribution from a subset s of the plane $s = \{(x,y) \in R^2; 0 \leq x \leq 1, 0 \leq y \leq 1\}$ and map these points to a subset of R^3 by adding a third constant element of 0.5.
The A-SOM A receives its main input from the constructed input set described above. In addition, its total activity is fed back as ancillary input with a time delay of one iteration.

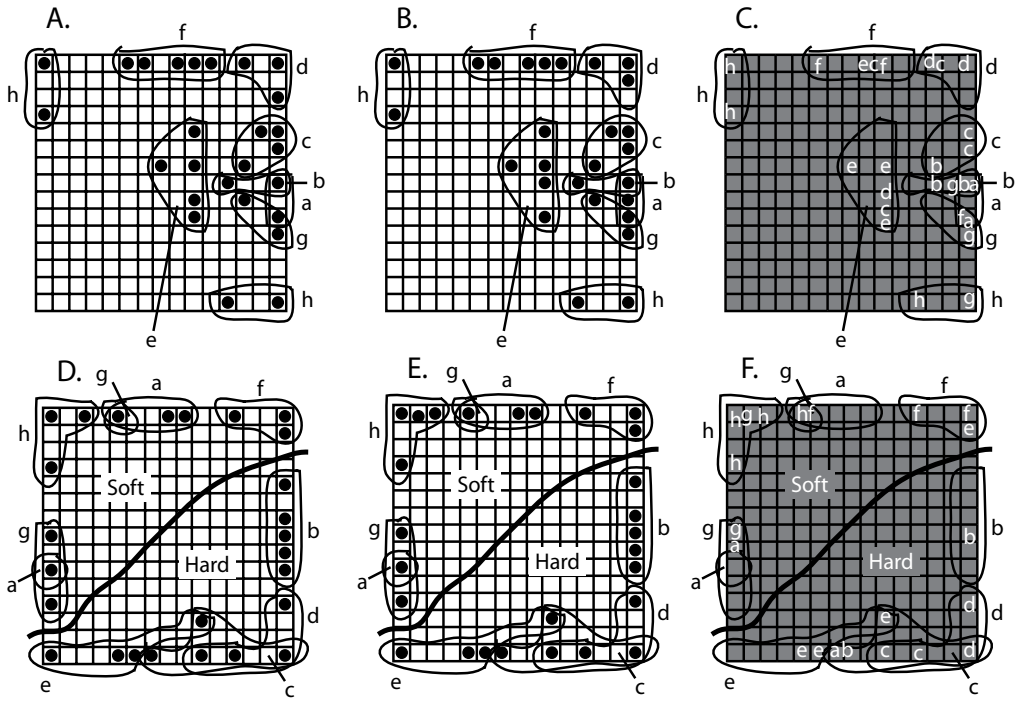


Fig. 7. The mapping of the objects used in the experiments. The characters a-h refer to the objects in Table 1. The images in the uppermost row correspond to the texture representing A-SOM and the images in the lowermost row correspond to the hardness representing A-SOM. Each cell in an image represents a neuron in the A-SOM, which consists of $15 \times 15 = 225$ neurons. A filled circle in a cell is supposed to mean that that particular neuron is the centre of activation for one or several explorations. The occurrence of a certain letter in the rightmost images means that there are one or several centres of activation for that particular object at that particular place. The centres of activation from the samples in the test set corresponding to each object in Tabel 1 when only main input was provided have been encircled in the images. A: The texture representing A-SOM when tested with main texture input. Most objects are mapped at separate sites so the system is able to discriminate between individual objects when provided with main input, although not perfectly. B: The texture representing A-SOM when tested with main texture input together with ancillary hardness input. Its activations are very similar to those in A. C: The texture representing A-SOM when it receives only ancillary hardness input. This often triggers an activity similar to the activity following main texture input. D: The hardness representing A-SOM when tested with main hardness input maps different objects at different sites and it perfectly discriminates hard from soft objects. E: The hardness representing A-SOM when tested with main hardness input together with ancillary texture input. Its activations are very similar to those in D. F. the hardness representing A-SOM when it receives only ancillary texture input. This often triggers an activity similar to the activity following main hardness input.

Besides the main input from the constructed input set, the A-SOM B receives the total activity of the A-SOM A as ancillary input without any time delay. Both A-SOMs were simultaneously fed with the 10 samples of the training set over and over again, all the time in the same

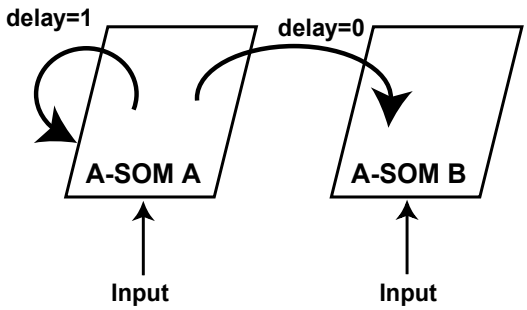


Fig. 8. Schematic depiction of the connections between the two A-SOMs and the recurrent connections of A-SOM A in the bimodal system. The bimodal system consist of two subsystems, which develop representations of sample sets from two input spaces (for simplicity we used the same input set for both representations). The A-SOM B learns to associate its activity with the activity of A-SOM A. This means proper activity can be elicited in the A-SOM B of the fully trained system even if it does not receive any main input. This is similar to cross-modal activation in humans, e.g. a tactile perception of an object invoking an internal visual imagination of the same object. One of the representations (A-SOM A) also learns to reproduce the sequence of activity patterns presented during training. Thus the sequence of activity patterns in A-SOM A elicits an appropriate sequence of activity patterns also in A-SOM B even when this lacks main input.

sequence, during a training phase consisting of 2000 epochs (i.e. 20000 iterations). The two A-SOMs could as well have been fed by samples from two different sets, always receiving the same combinations of samples from the two sets (otherwise the system could not learn to associate them). This could be seen as a way of simulating simultaneous input from two different sensory modalities when an animal or a robot explores its environment. Each of the two A-SOMs consisted of 15×15 neurons. The softmax exponent for each of them were set to 1000. Their learning rate $\alpha(0)$ was initialized to 0.1 with a learning rate decay of 0.9999 (i.e. multiplication of the learning rate with 0.9999 in each iteration), which means the minimum learning rate, set to 0.01, will be reached at the end of the 20000 training iterations. The neighbourhood radius, i.e. σ of the neighbourhood function $G_{ijc}(t)$ in eq. (6), was initialized to 15 for both A-SOMs and shrunk to 1 during the 20000 training iterations by using a neighbourhood decay of 0.9998 (i.e. multiplication of the neighbourhood radius with 0.9998 in each iteration). Both A-SOMs used plane topology when calculating the neighbourhood. β for the associative weights in both A-SOMs was set to 0.35.

After training, weight adaptation was turned of and the system was tested by feeding both A-SOM A and A-SOM B with the 10 samples from the training set once again in the same sequence as during the training phase, i.e. the system received input for one epoch. The centres of activity for each sample in both A-SOMs were recorded, and the corresponding Voronoi tesselations for the A-SOMs were calculated (Fig. 9 middle and right). The centres of activity, of course, always correspond to the localizations of the neurons in the A-SOMs. However, if we consider the centres of activity to be points in the plain, then we can calculate a Voronoi tesselation of the plane according to these points. In this way we will also get a division of the grid of neurons of each A-SOM. This is because each neuron in an A-SOM will be localized in a Voronoi cell or on the border between several Voronoi cells (when we see the localizations of the neurons as points in the plane).

Voronoi tesselations for the activity centres of the A-SOMs are used to assess the performance

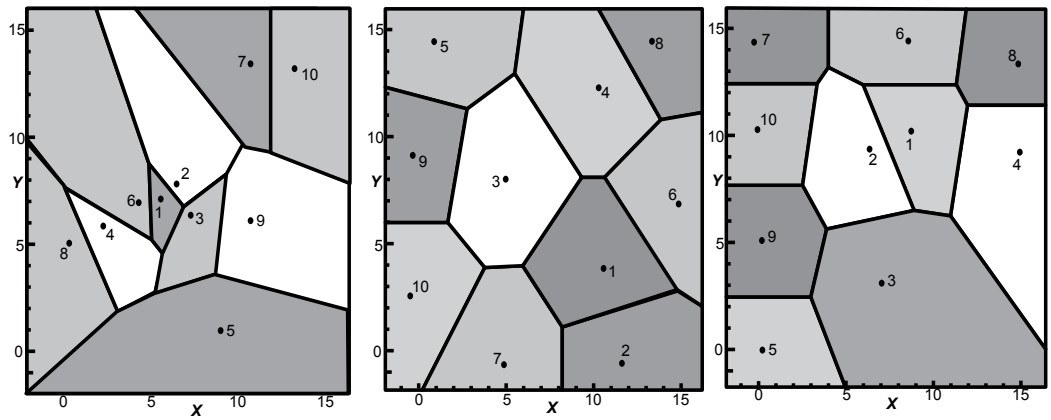


Fig. 9. Left: The Voronoi tessellation of the points used when constructing the training set used as input to the two A-SOMs in the bimodal system. Middle and Right: The Voronoi tessellations corresponding to the centres of activity during the first epoch of the test phase for the two A-SOMs. The image in the middle depicts the Voronoi tesselation of the fully trained A-SOM A together with the 10 centres of activity corresponding to the 10 first iterations of the test phase when the system received input from the sample set. The right image depicts the same but for the fully trained A-SOM B.

of the system. This is done in the following way: During the first epoch after training when the A-SOMs received main input, we recorded the sequences of Voronoi cells containing the centres of activity for the sequences of activity patterns in both A-SOMs. After the first epoch the A-SOMs did not receive main input anymore, i.e. only null vectors were received as main inputs. Anyway, sequences of activity patterns continued to be elicited in both A-SOMs. This means the system continued to run with internal states only. This is possible since A-SOM A received its own total activity as ancillary input with a time delay of one iteration and the A-SOM B received the total activity of A-SOM A as ancillary input without any time delay. For each of the following 25 epochs (without any main input to the A-SOMs) we recorded whether the centres of activity for each iteration in the epoch was in the correct Voronoi cell. If the centre of activity is in the correct Voronoi cell, then it is considered correct because then it is sufficiently similar to the centre of activity (from the first test epoch) that corresponds to that Voronoi cell. This is because then it is closer to the centre of activity (from the first test epoch) that corresponds to that Voronoi cell than to any other centre of activity from the first test epoch. This procedure enabled us to calculate the percentage of correct activity patterns for each of the 25 epochs without main input to the A-SOMs during the test phase. During these 25 epochs the activity is elicited solely by recurrent connections.

In Fig. 9 (middle and right) we can see that both A-SOMs perfectly discriminate between the 10 samples in the sample set, and by comparing the Voronoi tessellations of the A-SOMs (Fig. 9 middle and right) with the Voronoi tessellation of the plane for the training set (Fig. 9 left) we can see that the ordering of the Voronoi cells for the training set are to a large extent preserved for the Voronoi cells for the centres of activation in the A-SOMs.

Fig. 10 shows the percentages of correct activity patterns in each epoch (i.e. each sequence of 10 iterations) for each of the first 25 epochs when the system did not receive anymore main input. The diagram to the left in Fig. 10 depicts the result for A-SOM A, whereas the diagram to the right in Fig. 10 depicts the result for A-SOM B. As can be seen the percentage of correct

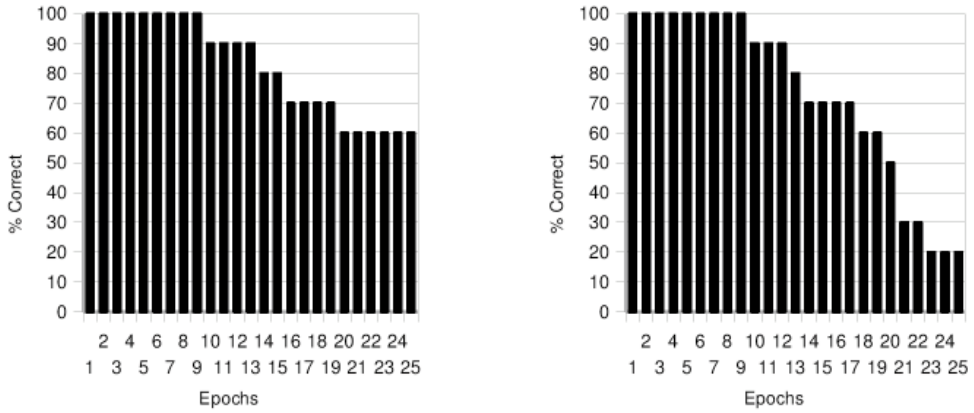


Fig. 10. The percentages of correct activity patterns in each epoch (i.e. sequence of 10 iterations) for each of the first 25 epochs when the system did not receive anymore main input. The diagram to the left depicts this for A-SOM A, whereas the diagram to the right depicts it for A-SOM B.

activity patterns is 100% for the first 9 epochs without main input in both A-SOM A and A-SOM B. The percentage of correct activity patterns then decline gradually in both A-SOMs and at the 25th iteration it is 60% for A-SOM A and 20% for A-SOM B.
In Fig. 11 we can see a graphical representation of the total, main and ancillary activities of the two A-SOMs when these receive input from the sample set as well as when they do not. The lighter an area is in this figure, the higher the activity is in that area.
To summarize, our model has shown ability to continue to produce proper sequences of activity in both A-SOMs for several epochs even when these have stopped receiving any main input. These results confirms the models ability of internal simulation as well as of cross-modal activation.

4.2 More experiments with recurrently connected A-SOMs

We have done three more experiments to investigate the properties, capabilities and limitations of the A-SOM when its activity is connected recurrently to itself as ancillary input with a delay of one iteration.
These experiments were inspired by Elman (1990) where a recurrent network is trained with sequences starting with a consonant and followed by a variable number of vowels. In his example he used the three sequences 'ba', 'dii' and 'guuu'. Each of these six letters (b, a, d, i, g and u) is coded as a vector of six binary digits. Elman (1990) motivates each of the positions of the vector as a feature of the letter such as consonant/vowel, if it is interrupted, hard, articulated in the back of the mouth and if it is voiced. These different features are, however, irrelevant for the functioning of the neural network. As long as the vectors for each letter is distinct from each other the network will learn them.
In all three experiments the neighbourhood radius was initialized to the same size as the network, i.e. in a $n \times n$ A-SOM the neighbourhood radius was initialized to n. The neighbourhood radius was multiplied by 0.9998 in each iteration, thus decaying exponentially. The learning rate was initialized to 0.1 and also decayed but by a factor of 0.9999. Minimum values for the neighbourhood radius and the learning rate was set to 1 and 0.01, respectively.

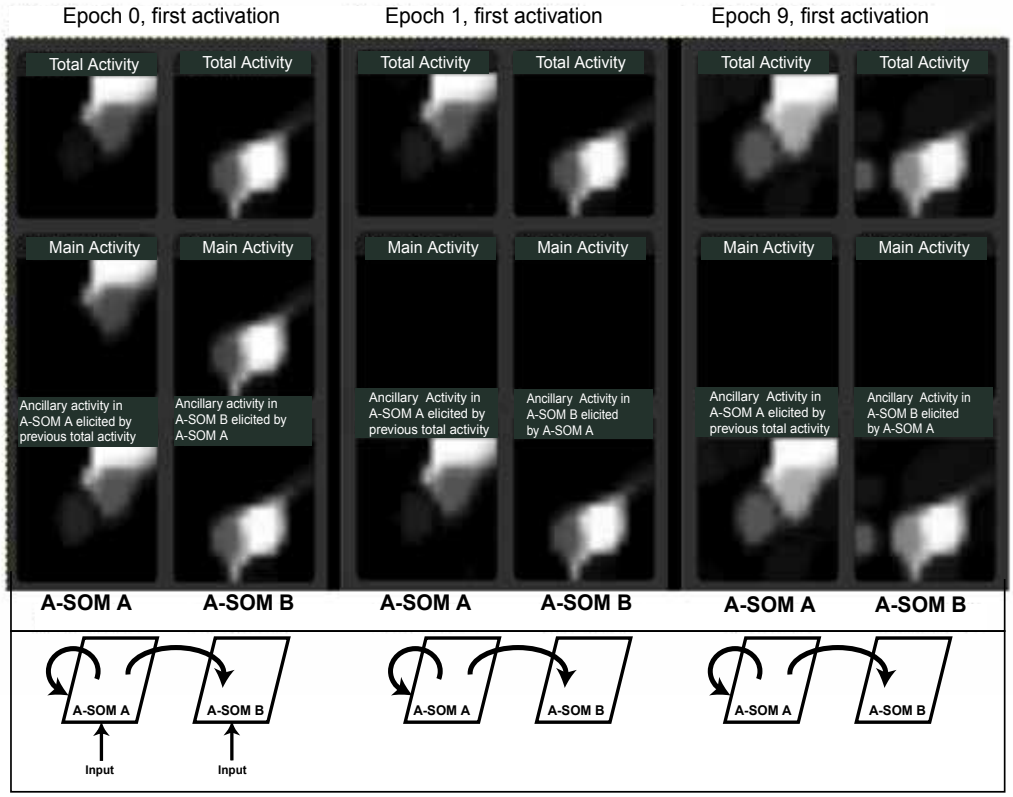


Fig. 11. Activations at three different occasions in the simulation with the bimodal system. The lighter an area is in the figure, the higher the activity is in that area. Column 1 (from the left): The three kinds of activity in A-SOM A when receiving the first sample of the sequence in the test phase. Column 2: The activities in A-SOM B when receiving the first sample of the sequence in the test phase. Column 3: The activities in A-SOM A when simulating the activity corresponding to sample 1 in the first epoch without input. Notice that there are no main activity. Column 4: The activities in A-SOM B elicited by the associative connections from A-SOM A due to the latter A-SOMs activity corresponding to the simulated activity of sample 1 in the first epoch without input. Column 5: The activities in A-SOM A when simulating the activity corresponding to sample 1 in the ninth epoch without input. This is the last epoch with perfect recall in the first cycle (see Fig. 10). Column 6: The activities in A-SOM B elicited by the associative connections from A-SOM A due to the latter A-SOMs activity corresponding to the simulated activity of sample 1 in the ninth epoch without input.

In all experiments the training phase lasted for 20 000 iterations.
In our experiments we created the training data in the same way as Elman (1990) i.e by repeating the sequences above in random order. This results in a semi-random sequence, i.e the consonants occur randomly but vowels always follow the consonants in a consistent manner (Elman, 1990). Structuring the input data in this semi-random way has an obvious advantage in that the network can be taught several sequences of input data. By having several sequences, the network is forced to be more versatile and generalized. The test data was produced in the same way as described above, but only leaving the first letter of every

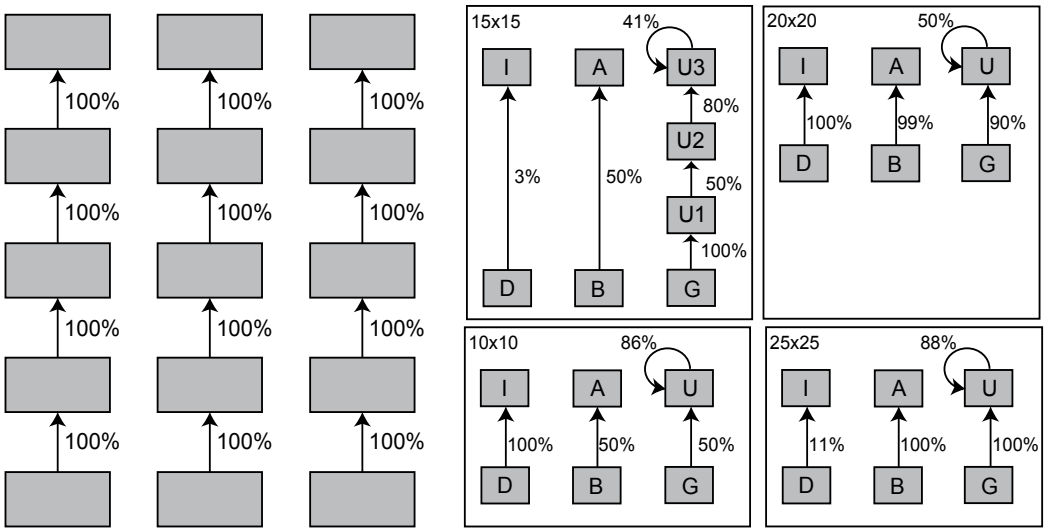


Fig. 12. Left: An example of an ideal state diagram for three sequences of length 5. Right: State diagrams for networks with 10×10, 15×15, 20×20 and 25×25 nodes.

sequence untouched and changing the others to null vectors (thus simulating no input at all). For every test iteration the coordinates of the winner neuron on the A-SOM surface were recorded. Then we analysed the sequential order of the winners by generating state diagrams. A state diagram can visualize a systems behaviour by specifying a number of states that the system may be in, as well as possible transitions between these states. A state diagram is generally drawn as a number of boxes, representing states, and arrows, representing the transitions. We label transitions with a percentage, indicating the probability for that transition compared to all transitions from the same source state. This means that our state diagrams are graphical representations of non-deterministic finite state machines, or what is also referred to as Markov chains.

An example of an ideal state diagram for three sequences of length 5 is shown to the left in Fig. 12. There will of course also be transitions from the topmost states to the bottom states (from the last letter of each sequence to the first letter of each following sequence), with, on average, a transition probability of 33%. However, these have been left out since they only indicate the start of a new sequence.

4.2.1 Different lengths, same vowels

The first experiment was a replication of one of Elman (1990) experiment where the network was trained on sequences representing 'ba', 'dii' and 'guuu'. That is, three sequences starting with a unique consonant followed by one, two or three of the same vowel. For these experiments, validation was made to find out which winner neuron corresponded to which letter in the sequences. This was done by simply giving the letters as input and registering which neuron was the winner for each letter. The A-SOM was then tested with test data constructed in the same way as the training data but with the vowels substituted with null vectors, as described above. The state diagrams for this experiment is shown to the right in Fig. 12 for four different sizes of the A-SOM; 10×10, 15×15, 20×20 and 25×25. To make the diagrams more readable, only the correct connections have been included. As an example of the activity in this experiment, Fig. 13 (left) shows the activity of the network one iteration

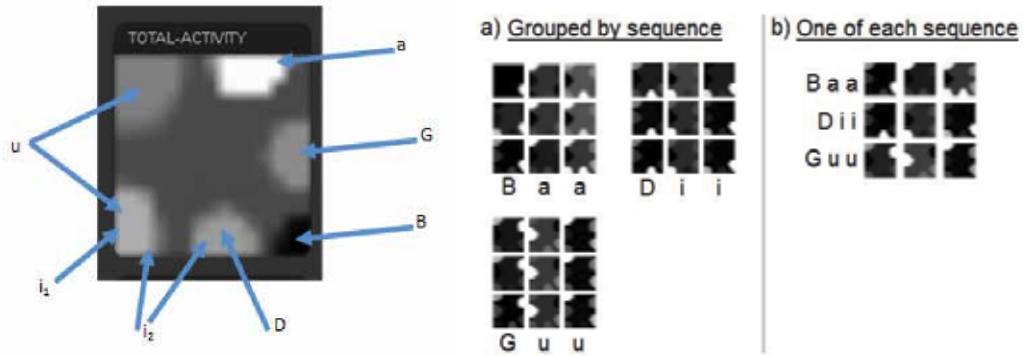


Fig. 13. Left: Activity pattern one iteration after presenting a 'B' in the first experiment with same vowels, different lengths. Right: Activity patterns for sequences in the experiment with same length and same vowels.

after presenting it with the letter B. The validated areas are shown with arrows. The letter 'u' is represented in two places (they both are activated when validating), as is 'i', but 'i' elicit maximum activity in different areas based on its position in the sequence, as indicated by i_1 and i_2.
In Fig. 13 (left) we can also see that there is a black spot in the lower left corner where there is no activity at all. One interpretation of this is that the network has learnt that a 'B' never follows an 'a'. One can also see that there is a little bit of activity in the rest of the network, even in parts that do not seem to represent any letter at all. The reason for this could be that for the lower left 'B' to be entirely void of activity, there need to be activity elsewhere to make that part have relatively lower activity, i.e. activity must be relative to all other activity.

4.2.2 Same length, same vowels

Having run Elman's original test, the experiment was modified so that the sequences were of equal length. They were all set to be three letters long; one consonant and two equal letters ('baa', 'dii' and 'guu'). The analysis of this experiment was different. Pictures were taken of the total activity of the network and assembled into a composite image so that the sequences of activity could be visualized easily. This was done to make a qualitative analysis of interesting ways in which the network represent different relations between states.
Fig. 13 (right) shows the activity of the network in the test of this experiment. In Fig 13a, the activity patterns of the network are grouped by sequence, to enable comparisons within sequences, whereas Fig 13b enables comparisons between sequences. One can see that activity patterns are distinct from each other for different positions even though the letter is the same (horizontally compare the first 'a' and second 'a' following a 'B' in Fig 13a.), though very similar for the same position between different trials (vertically compare the first 'a's in Fig 13a).

4.2.3 Same length, unique vowels

In the third experiment all sequences had the same length and all vowels were different, even within sequences. Note that we use the word vowel here to mean an element in the sequence, there is no connection to alphabetical vowels. The length of the sequences ranged from 2 to 19 elements. The aim here was to find the smallest network that could represent sequences of

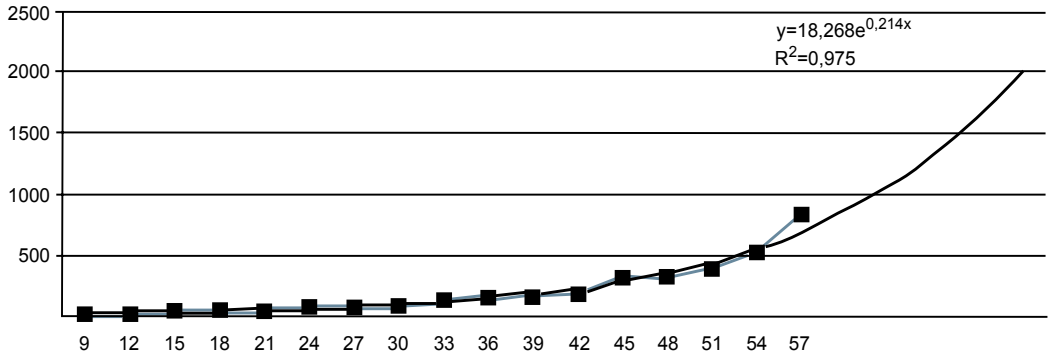


Fig. 14. Graph plotted for the number of nodes in smallest network with regard to the total number of letters in the sequences.

each length with 100% correctness or very close to it. To speed up this process we made the assumption that the minimum network size would not decrease when increasing the sequence length. That is, for a new sequence length the initial network size tested was the network size of the previously run test. It has turned out that this assumption does not hold strictly. In one training trial it was discovered that a network of 9×9 nodes was able to represent the sequences, while a network of 10×10 nodes performed worse. So even though the required network size seemed to increase with increased sequence length overall, there are minor local variations to this rule. This may simply be the effect of random variations in training data or the initial connection weights.

No validation was made for whether the sequence of winners from the testing was in the correct order. Only the pattern of states and their transitions percentages were used and it was manually tested whether the state diagram fitted with an ideal diagram, as shown in Fig. 12 (left). This should not be a problem since it would be extremely unlikely that an incorrect state transition would have a probability of 100%.

When running the trial where the sequence length was nine, no network seemed to be able to represent the sequences fully. Sizes up to 25×25 nodes were tried without success. It is worth mentioning that sequences of length eight only required 9×9 nodes. But, regenerating the test and training data and running the trial again relieved the matter and a 9×9 network was found that performed 100%. This could indicate that the network is sensitive to the training and test data, but to be certain further research should be done on the difference between these two training/test-sets.

Running the experiment with every trial having three sequences of the same length and all unique letters, and then plotting the smallest network size that could represent the sequences, produced the graph seen in Fig. 14. The graph shows that the number of nodes in the smallest network is exponential with regard to the number of total letters of the sequences in the trial.

4.2.4 Discussion

A simplification that has been made in these experiments to make analysis faster as well as more straightforward, has been to use only the sequence of winner neurons. Other activity of the network has thus been ignored. As one can see in Fig. 13 (left), the top area of the network (the 'a') is the winner here, but there is still much activity in other parts of the network. This can also be seen in the activity sequence series, Fig. 13 (right), where the same letter in different positions of the sequence exhibit different activity patterns. These same letter patterns have

distinct winners. This means that it is not completely satisfying to only record the winner neurons.
What one would want, rather, is to use the entire activity pattern instead of only the winner neuron. This would require some method to classify similar activity patterns, while separating not too similar patterns. Incidentally, this is a very suitable task for a regular SOM and we could thus use the activity of the A-SOM as input to a separate SOM, an analysis SOM, that would classify the activity of the A-SOM. Then the winner neurons of the analysis SOM, rather than the A-SOM, could be used to determine whether the sequences had been learnt.

5. Conclusion

We have presented a novel variant of the Self-Organizing Map called the Associative Self-Organizing Map (A-SOM), which develops a representation of its input space but also learns to associate its activity with the activities of an arbitrary number of (possibly time delayed) ancillary inputs. The A-SOM has been explored in several experiments.
In one experiment we connected an A-SOM to two ancillary SOMs and tested with randomly generated points from a subset of the plane. The system in this experiment could be seen as a model of a neural system with two monomodal representations (the two SOMs) and one multimodal representation (the A-SOM) constituting a neural area that merges three sensory modalities into one representation.
In another experiment we used the A-SOM in a bimodal self-organizing system for object recognition which used real sensors for the haptic submodalities hardness and texture. The results from this experiment are encouraging. The system turned out to be able to discriminate individual objects based on input from each submodality as well as to discriminate hard from soft objects. More importantly, the input to one submodality has shown to be sufficient to trigger an activation pattern in the other submodality, which resembles the pattern of activity the object would yield if explored with the sensor for this other submodality.
In other experiments we explored the ability of the A-SOM to learn sequences and we presented an A-SOM based bimodal model of internal simulation, and tested its ability to continue with reasonable sequences of activity patterns in its two A-SOMs in the absence of any input.
It is worth noting that although so far it has not been tested the authors can see no impediments to why it should not be possible to have several sets of connections that feed back the total activity of the A-SOM to itself as ancillary input but with varying lengths of the time delays. This would probably yield an enhanced ability for internal simulation and to remember perceptual sequences (at the cost of more computations).
Among other unsupervised recurrent architectures the Recursive SOM (Voegtlin, 2002) is probably the most similar to an A-SOM with recurrent connections. It is worth commenting on some similarities and differences. The two architectures mainly differ in the way a winner neuron is selected. The selection of a winner in the Recursive SOM depends on both the input vector and the time delayed feedback activity. This is not the case for the A-SOM, where the winner selection depends only on the input vector. Because of this, a reasonable guess would be that the A-SOM with recurrent connections would perform better than the Recursive SOM in classification of single inputs when not considering where in the sequence it comes. This is so because the organization of the A-SOM is completely independent of the recurrent input. The recurrent connections in the A-SOM are ancillary connections, which means there is a separate set of weights that during learning are adjusted to produce ancillary activity that is similar to the main activity. There might of course also be some disadvantages with the

A-SOM with recurrent connections when compared to the Recursive SOM. This would need further investigation.
The A-SOM actually develops several representations, namely one representation for its main input (the main activity) and one representation for each of the ancillary neural networks it is connected to (the ancillary activities), and one representation which merges these individual representations (the total activity). One could speculate whether something similar could be found in cortex, perhaps these different representations could correspond to different cortical layers.

6. Acknowledgements

This work was supported by the Swedish Linnaeus project Cognition, Communication and Learning (CCL), funded by the Swedish Research Council.

7. References

Balkenius, C., Moren, J., Johansson, B. & Johnsson, M. (2010). Ikaros: Building cognitive models for robots, *Advanced Engineering Informatics* 24(1): 40–48.

Bartolomeo, P. (2002). The relationship between visual perception and visual mental imagery: a reappraisal of the neuropsychological evidence, *Cortex* 38: 357–378.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*, Oxford University Press.

Carpenter, G., Grossberg, S., Markuzon, N., Reynolds, J. & Rosen, D. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps, *IEEE Transactions on Neural Networks* 3: 698–713.

Chappell, G. J. & Taylor, J. G. (1993). The temporal kohonen map, *Neural Networks* 6: 441–445.

Consortium, I. H. G. S. (2004). Finishing the euchromatic sequence of the human genome, *Nature* 431(7011): 931–945.

Elman, J. (1990). Finding structure in time, *Cognitive Science* 14(2): 179–211.

Hesslow, G. (2002). Conscious thought as simulation of behaviour and perception, *Trends Cogn Sci* 6: 242–247.

Johnsson, M. & Balkenius, C. (2008). Associating SOM representations of haptic submodalities, *in* S. Ramamoorthy & G. M. Hayes (eds), *Towards Autonomous Robotic Systems 2008*, pp. 124–129.

Johnsson, M., Balkenius, C. & Hesslow, G. (2009a). Associative self-organizing map, *International Joint Conference on Computational Intelligence (IJCCI) 2009*, pp. 363–370.

Johnsson, M., Balkenius, C. & Hesslow, G. (2009b). Neural network architecture for crossmodal activation and perceptual sequences, *Papers from the AAAI Fall Symposium (Biologically Inspired Cognitive Architectures) 2009*, pp. 85–86.

Kohonen, T. (1988). *Self-Organization and Associative Memory*, Springer Verlag.

Kohonen, T. (1990). The self-organizing map, *Proceedings of the IEEE, 78, 9*, pp. 1464–1480.

Kosslyn, S., Ganis, G. & Thompson, W. L. (2001). Neural foundations of imagery, *Nature Rev Neurosci* 2: 635–642.

McGurk, H. & MacDonald, J. (1976). Hearing lips and seeing voices, *Nature* 264: 746–748.

Miikkulainen, R., Bednar, J. A., Choe, Y. & Sirosh, J. (2005). *Computational maps in the visual cortex*, Springer.

Mountcastle, V. (1997). The columnar organization of the neocortex, *Brain* 120(4): 701–722.

Nguyen, L. D., Woon, K. Y. & Tan, A. H. (2008). A self-organizing neural model for multimedia information fusion, *International Conference on Information Fusion 2008*, pp. 1738–1744.

Strickert, M. & Hammer, B. (2005). Merge som for temporal data, *Neurocomputing* 64: 39–71.
Tan, A. H. (1995). Adaptive resonance associative map, *Neural Networks* 8: 437–446.
Varsta, M., Millan, J. & Heikkonen, J. (1997). A recurrent self-organizing map for temporal sequence processing, *ICANN 1997*.
Voegtlin, T. (2002). Recursive self-organizing maps, *Neural Networks* 15: 979–991.

31

Growing Topology Learning Self-Organizing Map

Vilson L. Dalle Mole and Aluizio F. R. Araújo
Federal Technologic University of Paraná, Federal University of Pernambuco
Brasil

1. Introduction

This chapter presents a neural model capable of learning the input space topology producing a true representation map. Such a model is an extension of GSOSM (Growing Self-Organizing Surface Map) (DalleMole & Araújo, 2008, 2010a) which was firstly developed considering the surface reconstruction problem. The GSOSM algorithm starts with an empty map and learns the surface topology from a dense cloud of points. With GSOSM, a folded surface is reproduced by a mesh of approximately equal and almost equilateral triangles whose granularity is controlled by a simple parameter. Because of its initial purpose of Surface Reconstruction, the GSOSM algorithm uses a cross product operation which is not applicable for $\mathbb{R}^n$, $n > 3$ spaces. Then, we have modified GSOSM replacing the cross product with another approach. Also, we have added an activation function to produce an unglued activation value. This makes possible some inferences without knows the value of parameter e_{max}. Then, we called the modified model as Growing Topology Learning Self-Organizing Map (GTLSOM) (DalleMole & Araújo, 2010b). Moreover, here we give a proof on GTLSOM capability to produce a topology preservation map. Thereafter, we used GTLSOM as a basic memory of perceptions in our work with mobile robots. We also described our proposed model to give to a robot the capability of learn about feasibility of navigation actions starting from contextual perceptions.
Section 2 presents a brief discussion about topology preservation meaning. Section 3 presents de GSOSM model. Section 4 presents the changes maid into GSOSM to produce the GTLSOM model. Section 5 presents our model for robot perception of obstacles and also presents some of obtained results. Our conclusions are given in Section 6.

2. Why topology learning

Consider a folded surface immersed into $\mathbb{R}^3$ space. The surface defines the topology of space comprising all surface points. This is very distant of classical math definition for vector space. However, a map describing the surface topology makes possible some inferences i.e. about the relative position of two surfaces. Pattern classification is another great area in which topology preserving maps are useful. Because a topology preserving map represents the patterns distribution subdividing the input space while representing the regions where a specific pattern has a positive probability density. Then, given a map or a set of maps describing the space topology for a set of patterns classes, the correct class of an input

sample is determined by inspecting which stored pair (pattern, class) better represent it. Therefore, topology preserving maps is useful for computational intelligence acting as a memory where patterns are stored and classified.

3. The GSOSM Model

The main goal of Growing Self-Organizing Surface Map (GSOSM) is to provide a learning approach for surface reconstruction from an unstructured point cloud producing a mesh of approximately equilateral triangles with similar areas. Its design can be characterized by: (i) an incremental mapping process in which the neighbourhood nodes are defined by a learning process; (ii) an algorithm without any dependence on error accumulators or cycle counters; (iii) a learning rule which handle input sequences presentation of similar or very distinct patterns; (iv) the capacity of produce a suitable representation for any shape; and (v) a mesh construction process which is eventually free of internal, overlapping, or false faces (DalleMole & Araújo, 2010a). The GSOSM algorithm have nine simple steps: (i) parameter setup; (ii) sample presentation; (iii) determination of closest neighbouring nodes starting from current sample; (iv) insertion of a new node; (v) weight vector update; (vi) node collapse; (vii) determination of winner connection with the current sample; (viii) insertion or substitution of a connection; and (ix) removal or swap of connections.

3.1 Initial map and parameters

The initial state of GSOSM map is an empty graph $G(S, C)$, where S is a set of nodes and C is the set of connections (edges) linking each pair of neighboring nodes. And, four parameter values needs to be specified: (i) e_{max} - The maximum local error, which is directly related to the length of connections between nodes (triangle edges); (ii) α - The learning rate; (iii) θ_{min} – specify the smallest value accepted for the maximum internal angle of a triangle; and (iv) β_{min} - The minimal correlation rate required to insert a new connection between a selected pair of nodes. This parameter constrains the insertion of a wrong connection because GSOSM employs it to identify two distinct parallel sheets of a surface.

3.2 The nodes insertion operator

The insertion of a new node aims to represent the samples belonging to non-mapped areas. GSOSM nodes distribution aims to represent the whole input space using the lowest possible number of nodes. Each node has a spherical receptive field with radius e_{max}, hence any input signal $\boldsymbol{\xi}$ within it is considered as mapped. Then a new node is inserted even an input signal $\boldsymbol{\xi}$ is determined as unmapped. The weight vector $\boldsymbol{\omega}_{s_{new}}$ of a newly node s_{new} is set equal to input signal $\boldsymbol{\xi}$. Moreover, a newly node has no initial connections because the neighbourhood relationships are established during connection learning process.

3.3 The adaptation operator

This step aims to produce fine adjustments in the map considering the topology of input space. The GSOSM adaptation schema has two self excluding adaptation steps. Given an input signal $\boldsymbol{\xi}$, find its three closest nodes $\{s_0, s_1, s_2\}$. Then, consider the plan P spanning the triangle $\mathcal{T}$ defined by the weigh vectors of these nodes, and a prism $\mathcal{P}$ defined by the three orthogonal planes intersecting P at the edges of $\mathcal{T}$. The first adaptation step is as in SOM (Kohonen, 1989) and is employed if $\boldsymbol{\xi}$ is not within $\mathcal{P}$ or if the nodes s_0, s_1 and s_2 are not interconnected.

$$\boldsymbol{\omega}_{s_0} = \boldsymbol{\omega}_{s_0} + \alpha\left[\boldsymbol{\xi} - \boldsymbol{\omega}_{s_0}\right] \tag{1}$$

Otherwise if $\boldsymbol{\xi}$ is inside of $\mathcal{P}$, the second adaptation step is employed. This step rotates the triangle $\mathcal{T}$ around an imaginary axis containing $\boldsymbol{\omega}_{s_1}, \boldsymbol{\omega}_{s_2}$ and towards $\boldsymbol{\xi}$, approximating the triangle face to the correct surface reconstruction.

$$\boldsymbol{\omega}_{s_0} = \boldsymbol{\omega}_{s_0} + \alpha\, \mathbf{P}_n\, d(P, \boldsymbol{\xi}) \tag{2}$$

where, $\mathbf{P}_n$ is the normal vector of plane P, and $d(P, \boldsymbol{\xi})$ is the distance between $\boldsymbol{\xi}$ and P.
This two step operator enables GSOSM to handle correlated samples while avoiding node overfitting**.** Whereas, if only the original SOM adaptation operator is applied, it will pulls the winner node towards the second closest node undoing the configuration of the equilateral triangles.

3.4 The merging operator

The GSOSM nodes insertion operator ensures a suitable distance between a newly node and any other node in the map. However, all nodes are subject to being repositioned by the adaptation operator. As a consequence, a repositioned node can invade the receptive field of node. Then, given an input signal $\boldsymbol{\xi}$, if the distance between the two closest nodes s_0 and s_1 is smaller than e_{max}, the nodes are merged. The resultant node s_{new} is positioned between the original positions of merged nodes.

$$\boldsymbol{\omega}_{s_{new}} = 0.5\left[\boldsymbol{\omega}_{s_0} + \boldsymbol{\omega}_{s_1}\right] \tag{3}$$

where $\boldsymbol{\omega}_{s_{new}}$ is the weight vector of resulting node.

3.5 The competitive connection hebbian learning – CCHL

GSOSM model introduced a new learning rule called Competitive Connection Hebbian Learning (CCHL). This is a substitute for CHL (Martinetz & Schulten, 1991) to learn the node-neighborhood relationships producing a triangulation. It is because surface reconstruction procedures need to reach a complete triangulation that can not be obtained using CHL (DalleMole & Araújo, 2008). This new learning rule is based on the concept of second order Voronoi diagram (Martinetz & Schulten, 1994), and it can be formally stated as: given an input signal $\boldsymbol{\xi}$ and its three closest nodes $\{s_0, s_1, s_2\}$, the pair to be connected is that forming the closest edge to $\boldsymbol{\xi}$.
Employing the CCHL rule, the winner connection could be determined considering the triangle $\mathcal{T}$ formed by the three closest nodes to the current sample $\boldsymbol{\xi}$. The line segments connecting each vertex to its incenter[1] determine the regions where the sample $\boldsymbol{\xi}$ is closest to one of its edges. Therefore, each possible connection has a Voronoi region where a sample induces its creation.

3.6 The connection insertion operator

In GSOSM, the main objective of insertion of connections is to obtain a complete triangulation producing a mesh representation with approximately equilateral triangles with similar size. Hence, given a sample $\boldsymbol{\xi}$, the CCHL rule decides the connection $c_{new}(s_i, s_j)$ to be inserted. However, the GSOSM imposes three other conditions that need to be satisfied before creating a connection: (i) The coefficient of correlation β (Equation 4)

[1] The point that is equidistant from all triangle edges.

satisfies (Equation 5); (ii) the coefficient of similarity θ (Equation 6) between s_i and s_j with respect to node s_k , $k \in [0,1,2]$, $k \neq i,j$ is greater than or equal to parameter θ_{min}; and (iii) c_{win} does not cross the Voronoi region of any other connection. The first condition verify the continuity of the input space between the nodes s_i and s_j; the second assures the formation of triangles that are approximately equilateral; and the third condition identifies crossing connections that could create overlapping triangle faces.

$$\beta = \frac{(\boldsymbol{\xi} - \boldsymbol{\omega}_{s_i}).(\boldsymbol{\omega}_{s_j} - \boldsymbol{\omega}_{s_i})}{\left\|\boldsymbol{\omega}_{s_j} - \boldsymbol{\omega}_{s_i}\right\|^2} \quad , \quad \left\|\boldsymbol{\xi} - \boldsymbol{\omega}_{s_i}\right\| \leq \left\|\boldsymbol{\xi} - \boldsymbol{\omega}_{s_j}\right\| \tag{4}$$

$$\beta_{min} \leq \beta \leq 0.5 \qquad \beta_{min} \in [0, 0.5] \tag{5}$$

The coefficient β is the projection of input signal $\boldsymbol{\xi}$ into the connection that is to be created. As the projection of $\boldsymbol{\xi}$ approximates to the connection midpoint the probability of inserting an erroneous connection is decreased.

The parameter β_{min} specifies a minimum value for the coefficient β and controls the GSOSM capability to differentiate between two parallel surface parts with a small gap between them. The coefficient θ is the cosine of the angle between vectors $\mathbf{v} = \boldsymbol{\omega}_{s_i} - \boldsymbol{\omega}_{s_k}$ and $\mathbf{u} = \boldsymbol{\omega}_{s_j} - \boldsymbol{\omega}_{s_k}$, therefore $\theta \in [-1,1]$. The insertion of a new connection inside of a regular pentagon has to satisfy $\theta_{min} \lesssim \cos(108°) \cong -0.309$ because this is the angle between two consecutive pentagon edges. However, as θ_{min} goes to -1 the algorithm becomes inefficient allowing insertion of connections producing irregular triangles with a large variation of edge lengths which will be removed rapidly. The coefficient θ is determined by:

$$\theta = \frac{\boldsymbol{\omega}_{s_i} - \boldsymbol{\omega}_{s_k}}{\left\|\boldsymbol{\omega}_{s_i} - \boldsymbol{\omega}_{s_k}\right\|} \cdot \frac{\boldsymbol{\omega}_{s_j} - \boldsymbol{\omega}_{s_k}}{\left\|\boldsymbol{\omega}_{s_j} - \boldsymbol{\omega}_{s_k}\right\|} \tag{6}$$

The third constraint stated that c_{win} does not cross any Voronoi region of other connections. This constraint is verified using the CCHL to verify if the midpoint of c_{win} is within a Voronoi region of another connection $c_{s_k s_t}$, $t \neq i,j$. Moreover, the search is restricted to the connections emanating from s_k and going towards the mid point between $\boldsymbol{\omega}_{s_i}$ and $\boldsymbol{\omega}_{s_j}$.

3.7 The fidelity coefficient

The coefficient ϕ (Equation 7) is a measure about the fidelity representation of a connection and GSOSM employs it to determine if the connection insertion process should be aborted; or if a existent connection should be replaced by c_{win}. To select the action to be carried out, GSOSM choose the action that preserves the connection with the highest ϕ value.

$$\phi = \left(1 + |\beta - 0.5| + \frac{d(c_{s_i s_j}, \boldsymbol{\xi})}{\left\|c_{s_i s_j}\right\|}\right)^{-1} \tag{7}$$

with $$d\left(c_{s_i s_j}, \boldsymbol{\xi}\right) = \left\|\left[\boldsymbol{\omega}_{s_i} + \beta\left[\boldsymbol{\omega}_{s_j} - \boldsymbol{\omega}_{s_i}\right]\right] - \boldsymbol{\xi}\right\| \text{ and } \left\|c_{s_i s_j}\right\| = \left\|\boldsymbol{\omega}_{s_i} - \boldsymbol{\omega}_{s_j}\right\|$$

where $\boldsymbol{\omega}_{s_i}$ and $\boldsymbol{\omega}_{s_j}$ are the weight vectors of nodes at the extremities of connection $c_{s_i s_j}$.
The last component of Equation 7 is the distance between the sample $\boldsymbol{\xi}$ and the connection c_{ij}, whereas the middle component ensures that samples closest to the connection midpoint have a ϕ value greater than other ones. The constant first term is used to avoid numeric ill condition only. Thereby, the coefficient ϕ represents the connection fidelity as a direct function of samples that are captured by the Voronoi region of the connection. Moreover, the stored ϕ value is updated as the connection is a winner, therefore maintaining even the highest value.

3.8 The connection removal operator

The removal operator aims to eliminate long connections which combined with suitable connections produce non-equilateral triangles. Another objective is to eliminate cross connections which produce overlapping triangles in the mesh. Therefore, this is a two steps operator. The first step, is based on the connection removal operator of the ITM model (Jockusch & Ritter, 1999) and removes long connections. GSOSM considers the nodes s_i and s_j at extremities of connection c_{win} and removes the connections $c_{s_i s_t}$ satisfying the Equation 8 (except $c_{s_i s_j}$). In GSOSM, the parameter θ_{min} replaces the fixed value $\frac{\pi}{2}$ of the ITM model.

$$\frac{\boldsymbol{\omega}_{s_i} - \boldsymbol{\omega}_{s_j}}{\left\|\boldsymbol{\omega}_{s_i} - \boldsymbol{\omega}_{s_j}\right\|} \cdot \frac{\boldsymbol{\omega}_{s_k} - \boldsymbol{\omega}_{s_j}}{\left\|\boldsymbol{\omega}_{s_k} - \boldsymbol{\omega}_{s_j}\right\|} < \theta_{min} \ , \tag{8}$$

If the removal of a long connection results in a lozenge without any diagonal, GSOSM uses a swap operator ensuring a complete local triangulation. To couple with the surface folds, the whole process is aborted if the swap results in a connection whose value of coefficient ϕ is smaller than $\phi_{c_{s_i s_k}}$.
The second step employ CCHL rule to find and remove a cross connection, eliminating overlapping triangle faces. It is carried out using the CCHL to verify if the midpoint of c_{win} is within another connection Voronoi region (DalleMole & Araújo, 2010a). If an intersection is detected, the connection with the smallest ϕ value is removed.

3.9 GSOSM algorithm

The algorithm of the GSOSM is as follow.

1. Give the values to the parameter set and initialize an empty map;
2. While the terminate condition is not satisfied
   - 2.1. Sample a new stimulus $\boldsymbol{\xi}$;
   - 2.2. Run the node insertion operator as in Subsection 3.2;
     - If a new node is inserted or the map size is lower than three nodes, go to Endwhile;
   - 2.3. Construct the neighborhood set $N(\boldsymbol{\xi})$, the three closest nodes of $\boldsymbol{\xi}$;
   - 2.4. Run the adaptation operator as in Subsection 3.3;
   - 2.5. Run the merging operator as in Subsection 3.4;
     - If the pair s_0, s_1 is merged, go to Endwhile;
   - 2.6. Run CCHL to determine the winner connection $c_{s_i s_j}$ to current input signal $\boldsymbol{\xi}$ as in Subsection 3.5;

```
    2.7. If a winner connection c_{s_i,s_j} is identified
      • Determine or actualize the value of coefficient ϕ as in Subsection
        3.7;
      • Run the connection insertion operator as in Subsection 3.6;
      • If a new connection was inserted, go to Endwhile;
      • Run the connection removal operator as in Subsection 3.8;
    End_while;
3. End.
```

4. The GTLSOM model

Basically, the GTLSOM model is an extension of GSOSM model which was initially conceived for surface reconstruction. GTLSOM introduces two little changes turning the model suitable to work with any entry space. The first one includes an adjustment of Equation 2, because the cross product used in GSOSM is not defined for $\mathbb{R}^n$, $n > 3$. The second includes an activation function to represent the activation of nodes as a value ungrounded to parameter e_{max}. These changes are detailed bellow.

4.1 Adjusting the second adaptation atep

The second step of adaptation of GSOSM algorithm (Equation 2) performs a triangle rotation. This step requests the computation of a normal vector considering the plane formed by the three closest nodes to input signal $\boldsymbol{\xi}$. Commonly, a normal vector is obtained employing the cross product operator. However, this is not defined for a space $\mathbb{R}^n$, $n > 3$, then we used the method proposed by Sloughter (2001) and so the following text is based on its work.

In order to get an suitable orthogonal vector, given the input signal $\boldsymbol{\xi}$ and its three closest nodes s_0, s_1 and s_2 and the respective model vectors $\boldsymbol{\omega}_{s_i}$, $i = 0,1,2$. If vectors $\boldsymbol{\omega}_{s_i}$ are linearly independent they determine a plane P. Then determine the vectors $\mathbf{v}$ and $\mathbf{w}$ lien in the plane P as

$$\mathbf{v} = \boldsymbol{\omega}_{s_1} - \boldsymbol{\omega}_{s_0} \text{ , } \mathbf{w} = \boldsymbol{\omega}_{s_2} - \boldsymbol{\omega}_{s_0} \tag{9}$$

and the parametric equation of plane P as

$$\mathbf{y} = t\mathbf{v} + s\mathbf{w} + \boldsymbol{\omega}_{s_0}. \tag{10}$$

In a general sense, given a plane P with equation $\mathbf{y} = t\mathbf{v} + s\mathbf{w} + \mathbf{p}$, if we find two vectors $\mathbf{a}$ and $\mathbf{b}$ lying in P such that $\mathbf{a} \perp \mathbf{b}$ with $\|\mathbf{a}\| = 1$ and $\|\mathbf{b}\| = 1$, becomes that equation $\mathbf{y} = t\mathbf{a} + s\mathbf{b} + \mathbf{p}$ describes the same plane P. Therefore, taking $\mathbf{c}$ as the projection of $\mathbf{w}$ into $\mathbf{v}$ the vectors $\mathbf{a}$ and $\mathbf{b}$ can be wrote as

$$\mathbf{a} = \frac{1}{\|\mathbf{v}\|}\mathbf{v}, \tag{11}$$

$$\mathbf{b} = \frac{1}{\|\mathbf{w} - \mathbf{c}\|}(\mathbf{w} - \mathbf{c}). \tag{12}$$

Moreover, given a vector $\mathbf{q}$ not in P, let $\mathbf{r}$ be the sum of the projections of $\mathbf{q}$ onto $\mathbf{a}$ and $\mathbf{b}$,

$$\mathbf{r} = (\mathbf{q}.\mathbf{a})\mathbf{a} + (\mathbf{q}.\mathbf{b})\mathbf{b}. \tag{13}$$

Without lost of generallity, consider the plane P through the origin. It follows that for any $\mathbf{y} = t\mathbf{a} + s\mathbf{b} + \mathbf{p}$ in plane P, becomes

$$(\mathbf{q} - \mathbf{r}).\mathbf{y} = (\mathbf{q} - \mathbf{r}).(t\mathbf{a} + s\mathbf{b}) = t(\mathbf{q} - \mathbf{r}).\mathbf{a} + s(\mathbf{q} - \mathbf{r}).\mathbf{b} = 0, \tag{14}$$

that is, $\mathbf{q} - \mathbf{r}$ is orthogonal to every vector in the plane P. Returning to the initial problem, all we need is to find $\mathbf{r}$ and then doing $\mathbf{P}_n = \frac{\boldsymbol{\xi}-\mathbf{r}}{\|\boldsymbol{\xi}-\mathbf{r}\|}$ and $d(P, \boldsymbol{\xi}) = \|\boldsymbol{\xi} - \mathbf{r}\|$, the Equation 2 becomes

$$\boldsymbol{\omega}_{s_0} = \boldsymbol{\omega}_{s_0} + \alpha(\boldsymbol{\xi} - \mathbf{r}). \tag{15}$$

4.2 Setting up an activation function

We have considered a Gaussian function as a suitable function to express the node activation in terms of its similarity to the input signal $\boldsymbol{\xi}$. It is because the output of this function type is constrained into interval [0;1]. Moreover, we can control its radial base enforcing the value of σ as need. Then, setting $\sigma = e_{max}$, the equation of activation level of a node s_i becomes.

$$a_{s_i} = exp\left(-\frac{\left\|\boldsymbol{\xi} - \boldsymbol{\omega}_{s_i}\right\|^2}{2.{e_{max}}^2}\right), \tag{16}$$

where a_{s_i} is the activation of node s_i, $\|\cdot\|$ is the Euclidean distance.
Note that using Equation 16, a value of $a_{s_i} \cong 0.6$ is obtained when the input signal is located near the boundaries of receptive field of node s_i. Moreover, the relative position of $\boldsymbol{\xi}$ and its three closest neighbouring nodes can be estimated using the information provided by a_{s_i}.

4.3 Topology preservation

In the common sense, the topology preservation is understood as the degree to which a map preserves the information about proximity between points and about the continuity of entry space. The topology preservation implicates into the equivalence between the generated map and the entry space for all represented points and its neighbourhoods (Vilmann, 1999).
Definition 1: A graph G together its nodes i associated to points $\mathbf{w}_i$ forms a topology preserving map of a entry space (*manifold*) Ω, if exists a mapping function $\varphi: \Omega \longrightarrow G$ leading neighbour signals in Ω to neighbour nodes in G; and its inverse mapping $\varphi^{-1}: G \longrightarrow \Omega$ leading nodes that are neighbour in G to neighbour places in Ω (Martinetz, 1994).
From insertion operator (Subsection 3.2) each input signal $\boldsymbol{\xi}$ is classified to a node $s_i \in S$ with a receptive field, or produces the insertion of a new node s_{new} with a weight vector $\boldsymbol{\omega}_{s_{new}} = \boldsymbol{\xi}$. Therefore, the first condition is directly satisfied by the GTLSOM mapping process:

$$\varphi: \Omega \longrightarrow S \mid \varphi(\boldsymbol{\xi}_k) = s_i,\ s_i \in S,\ \forall \boldsymbol{\xi}_k \in \Omega. \tag{17}$$

The inverse mapping φ^{-1} is also directly verified because a new node s_{new} is created setting $\boldsymbol{\omega}_{s_{new}} = \boldsymbol{\xi}$:

$$\varphi^{-1}: S \longrightarrow \Omega \mid \varphi^{-1}(s_i) = \boldsymbol{\xi}_k, \qquad \forall s_i \in S \rightarrow \boldsymbol{\xi}_k \in \Omega. \tag{18}$$

Moreover, using CCHL the connection insertion operator is fired only if an input signal falls between two nodes. Therefore, the connection represents the input space continuity. Then, the existence of mapping functions φ and φ^{-1} plus CCHL connection learning rule ensure the topology preservation.

5. Robot obstacle perception and safe navigation learning

We have used GTLSOM to enables a robot to learn about its surrounding environment while identifying the presence of obstacles or empty space. We used a GTLSOM map as a basic memory for robot perceptions because its capabilities of storing and clustering similar inputs. Whereas a learning schema was employed to produce synapses linking the robot perceptions to navigation actions. The main objective was learning about feasibility of each possible navigation action, starting from a remembering process of context similarities. The cognitive process recovers the knowledge stored into synapses producing a value judgment to subsidize the robot behavior. In other words, the robot learns about the feasibility of each action into the perceived surrounding environment context. Once a local environment has been explored the assimilated knowledge remains available and the robot makes use of it to navigate into unknown environments. Moreover, the stored knowledge is continuously refined and updated. Lastly, we developed a layered schema (Fig. 1) to accommodate the subsystems in a modular way.

The first layer is responsible for sense the surrounding environment producing perceptions. The second layer is a GTLSOM map grouping an storing perceptions producing a map, a basic memory, $\mathcal{M}^p$, for perceptions. At layer 3 a learning process conjugates information about recollected memories, selected actions and motor feedbacks, producing and modifying synapses between memories and actions (Fig. 2). Another process uses the knowledge stored into the synapses to produce a value judgment about action feasibility. The process at layer 4 considers the produced judgments to select the action to be carried out and signal it to motor subsystem. The motor subsystem accepts, as entry, a vector specifying the action command which is executed producing a feedback signal for Synapse Learning module.

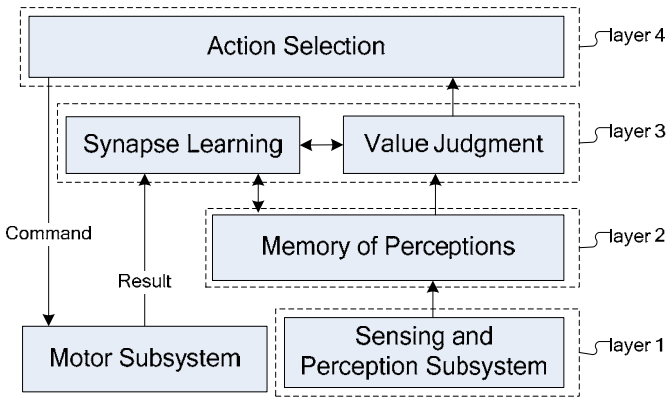


Fig. 1. Diagram of blocks of layered schema for robot navigation system

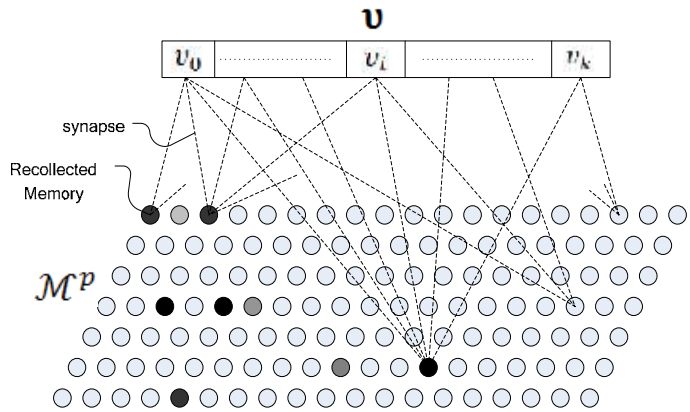


Fig. 2. Memory of perceptions and the synapses between actions $\boldsymbol{v_i}$ and memory units

5.1 Sensing and perception subsystem

We have considered an internal and structured environment composed of rooms, corridors and passages between them. And a robot with a body shaped like a box having six laser based sensor devices one on each lateral face one on top and other on bottom face. Each sensor device s_i produces a flow $\mathcal{F}^{s_i}$ of reads $\mathbf{x}_t^{s_i}$ sweeping its laser spot by 180º stepping no more that 10º between reads. The perception module uses the vectors of sensor reads recovering a environment semi contour (Fig. 3) surrounding the robot for each instant t.

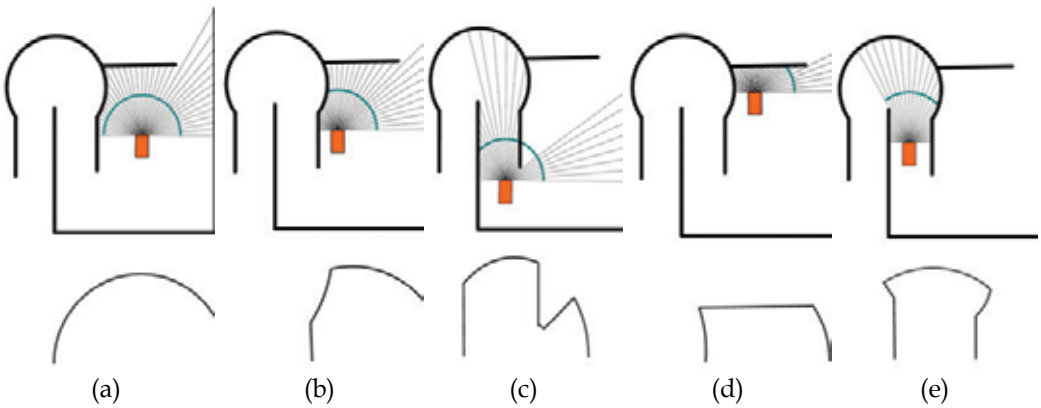


Fig. 3. Samples of surrounding contour perception produced using reads from frontal sensor device. The circular line crossing the lines of reads indicates the prunning distance

The sensor reads are pruned (Equation 19) so that only the immediate surrounding environment is considered as specified by parameter $d_{\max}$. Then, each perception p is codified into a vector $\mathbf{w}_i \in \mathbb{R}^{\hbar}$ using $\hbar$ first geometric moments of Zernick (1934). Firstly the surrounding contour is plotted into a circular bitmap image with radius equals to value of parameter r and then the values of geometric moments are determined. Therefore, a perception represents a contour image as illustrated in Fig. 3 and is the basic unit of perceptive flows.

$$\mathbf{x'}_t^{s_i} = [x'_0, \dots, x'_i, \dots], x'_i = (\mathbf{u}_i, d'_i),\ d'_i = \begin{cases} d_i & if\ d_i \le d_{max} \\ d_{max} & if\ d_i > d_{max} \end{cases}, \tag{19}$$

where $\mathbf{u}_i$ is the direction vector of read; d_i and d'_i are, respectively, the measured and pruned distance; and $\mathbf{x'}_t^{s_i}$ is the vector of pruned reads.
This representation schema preserves the image discretionary capability while enables direct comparisons employing Euclidean distance. Moreover, it was proved effective and under determined conditions could reduce the space of configurations to a finite number. Such reduction could be reached because the pruning step and the use of a finite number of geometric moments and because GTLSOM groups similar inputs with a receptive field.

5.2 Synapse learning

This module knows which units of memory were activated by perceptions coming from each sensor device, as well as actions taken based on them and the feedback received from motor subsystem. Such information is employed for synapse establishment or adjustment. The actions are represented by a vector $\boldsymbol{\upsilon} \in \mathbb{R}^k$ with k being the degrees of freedom of robot. For each possible action (*go_ahead, go_back, turn_left, turn_right and soon)* a set of synapses is constructed and the set of all synapses and its stored values forms a base of knowledge about actions feasibility for each local environment context. Each synapse is valued into an interval [0,1] with initial value being given by parameter μ, which specify the minimum value to an action be considered as feasible. Therefore, learning occurs by adjusting synapse value through a schema of punishment and reward (Kaebling *et al*., 1996); (Sutton & Barto, 1998).
The synapse adjustment consider the coefficients represented by parameters α_r and α_p and affect only the synapses $s_{l_i v_j}$ between the recollected memory l_i and the associated action v_j. The adjustments are determined using Equation (20) for successful actions and Equation (21) for those producing unwanted results i.e. a collision with an obstacle.

$$s'_{l_i v_j}(t') = \begin{cases} s_{l_i v_j}(t') + \alpha_r^{t'}\left(1 - s_{l_i v_j}(t')\right)\ if\ sucess \\ s_{l_i v_j}(t') + \alpha_p^{t'}\left(0 - s_{l_i v_j}(t')\right)\ if\ failure \end{cases},\ \forall t' \in]t - \tau, t], \tag{20}$$

with $\alpha_r^t = \alpha_r,\ \alpha_r^{t-1} = 0.5\alpha_r^t$
$\alpha_p^t = \alpha_p,\ \alpha_p^{t-1} = 0.5\alpha_p^t$,

$$s'_{l_i v_j}(t) = s_{l_i v_j}(t) + \alpha_p^t\left(0 - s_{l_i v_j}(t)\right),\ \alpha_p^t = \alpha_p, \tag{21}$$

where $s_{l_i v_j}$ is the synapse value at time t and $s'_{l_i v_j}$ is the synapse value after adaptation step; v_j is the $j - ésima$ component of $\boldsymbol{\upsilon}$ and represents an possible action; α_r and α_p are paramters representing the rates of reward and punishment respectively, $\alpha_r, \alpha_p \in [0; 1]$; t is the current time instant and t' is the time instant in which the action was executed; and τ is a parameter controlling the number of time instants to back propagation of adjustments for a selected action executed successfully.
Therefore, the adjustment affects positively the η stronger synapses, if the action is executed successfully. And negatively the synapses that subsidized the selection of an action with unwanted results.

5.3 Action value judgment

For each time instant a value judgment is determined for all possible actions. Then, for each component v_j of $\mathbf{v}$ this process considers the average of values stored in the synapses between it and the first λ activated units of memory. Therefore, the output of this process is a k dimensional vector of values into interval $[0; 1]$ representing the robot knowledge about the feasibility of each navigation action into the current context. As mentioned early, a parameter μ is used to signal a minimum value for an action to be considered as feasible.

5.4 Algorithm

1. Get the sets P^{s_i} of current perceptions from flows $\mathcal{F}^{s_i}$;
2. Present each set P^{s_i} to the basic memory $\mathcal{M}^p$ getting the respective list $\mathcal{L}^{s_i}$ of activated units;
3. For each component action $v_j \in \mathbf{v}$. produce a value judgment considering the respective list $\mathcal{L}^{s_i}$
   - 3.1. Sort $\mathcal{L}^{s_i}$ according synapses values;
   - 3.2. Determine the value judgment of v_j as the average of first λ synapses in $\mathcal{L}^{s_i}$;
   - 3.3. Registry the first λ synapses into the historic list ℓ of value subsidy;
4. Signal the value judgment for the Action Selection module;
5. For each navigation action taken
   - 5.1. Catch the feedback signal (**SUCESS** / **UNWANTED**);
   - 5.2. Update the historic list l of last τ navigation actions taken;
   - 5.3. If the status of action $l(t)$ indicates **SUCESS** employ the adjustment schedule defined by Equation (20).
     - 5.3.1 Initialize $t' = t$, $\alpha_r^{t'} = \alpha_r$ and $\alpha_p^{t'} = \alpha_p$;
     - 5.3.2 For each action in l, consider the synapses registered in $\ell(t')$
       - If **REWARD**, adjust the values of η less pessimistic synapses;
       - If **PUNISHMENT**, adjust the values of λ more pessimistic synapses;
       - Decrement t' updating $\alpha_r^{t'}$ and $\alpha_p^{t'}$
   - 5.4. If the status of action $l(t)$ indicates **UNWANTED**
     - 5.4.1 Adjust the values of λ more pessimistic synapses in $\ell(t')$ (Equation 21);
6. Return to step 1;

5.5 Results

This section presents three sets (A, B and C) of results obtained with our model for robot obstacle perception and safe navigation learning. In test sets A and C the robot is constrained to navigate in 2D whereas set B refers to robot navigating in 3D. In set C the robot was removed from a known environment part and introduced into another similar environment part. The selection of navigation actions was subject of a probabilistic algorithm. Then the robot remain wandering and swaps the selected navigation action if it collides with an obstacle or the value judgment of current action indicates that it is unsafe.

5.5.1 Parametrization

The basic memory of perception $\mathcal{M}^p$ is a GTLSOM map then its parameter set was established setting $\mathcal{k} = 10$, $e_{max} = 20$ and $e_{max} = 40$ defining two test subsets. The parameters set for the laser based sensors was established setting $sweep\_angle = 180°$, $sweep\_step = 10°$ and $rotation\_angle = 180°$, $rotation\_step = 15°$ for the case 3D. The robot velocity was fixed as two units of measurement per time unit and two degrees per time unit for rotation velocity. The association between sensors and navigation actions is given in the Table 1.

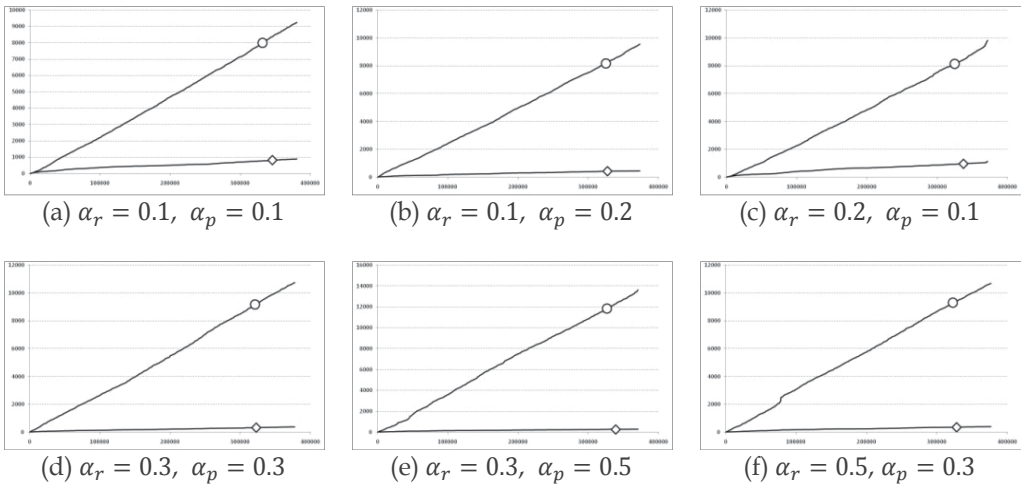
Sensor	Front	Left	Right	Back	Top	Botton
Action	go ahead	turn left	turn right	go back	float up	float down

Table 1.Sensors and Navigation Actions Association

The parameter set of our model was established setting $d_{max} = 100$, $r = 200$, $\mu = 0.6$, $\tau = 3$, $\lambda = 3$, $\eta = 1$. The coeficientes α_r and α_p were set according the test case.

5.5.2 Test set A – navigating in 2D

The graphs of Fig. 4 and Fig. 5 report the robot behaviour evolution when positioned towards an obstacle. Each graph show the relation between the number of navigation actions aborted in function of its judgment value (circular marker) and the number of navigation actions with unwanted results (lozenge marker).

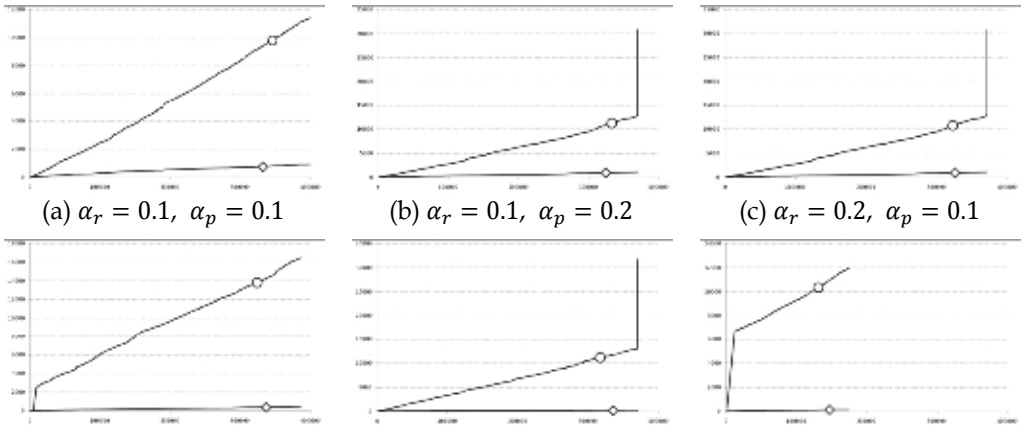


(a) $\alpha_r = 0.1,\ \alpha_p = 0.1$ (b) $\alpha_r = 0.1,\ \alpha_p = 0.2$ (c) $\alpha_r = 0.2,\ \alpha_p = 0.1$

(d) $\alpha_r = 0.3,\ \alpha_p = 0.3$ (e) $\alpha_r = 0.3,\ \alpha_p = 0.5$ (f) $\alpha_r = 0.5, \alpha_p = 0.3$

Fig. 4. Robot navigation behavior if positioned towards an obstacle and the parameter of perception memory granularity set to $e_{max}^{\mathcal{M}^p} = 20$

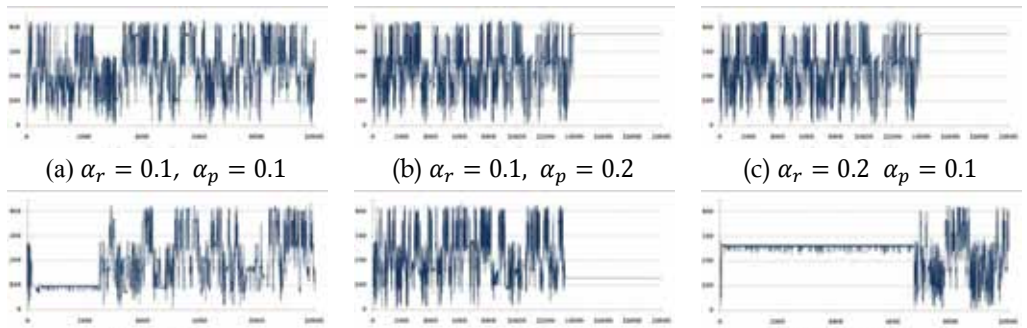
The Fig. 4 shows that the robot is learning to avoid the obstacles while maintaining a wandering behaviour. However, note the abnormalities into graphs (b, c, and e) of Fig. 5, this is because the robot stuck in a local minima where all selected navigation actions were judged as not feasible. While, the graphs (d) and (f) of Fig. 5 reveals a local minima where the robot stuck however not totally immobile, it remained alternating between antonym actions as *turn_left* and *turn_right*. This behaviour could be better visualized in Fig. 6.



(a) $\alpha_r = 0.1,\ \alpha_p = 0.1$ (b) $\alpha_r = 0.1,\ \alpha_p = 0.2$ (c) $\alpha_r = 0.2,\ \alpha_p = 0.1$

(d) $\alpha_r = 0.3,\ \alpha_p = 0.3$ (e) $\alpha_r = 0.3,\ \alpha_p = 0.5$ (f) $\alpha_r = 0.5,\ \alpha_p = 0.3$

Fig. 5. Robot navigation behavior if positioned frontwards an obstacle and the parameter of perception memory granularity set to $e_{max}^{\mathcal{M}^p} = 40$



(a) $\alpha_r = 0.1,\ \alpha_p = 0.1$ (b) $\alpha_r = 0.1,\ \alpha_p = 0.2$ (c) $\alpha_r = 0.2\ \ \alpha_p = 0.1$

(d) $\alpha_r = 0.3,\ \alpha_p = 0.3$ (e) $\alpha_r = 0.3,\ \alpha_p = 0.5$ (f) $\alpha_r = 0.5,\ \alpha_p = 0.3$

Fig. 6. Robot navigation behavior for $e_{max}^{\mathcal{M}^p} = 40$

Moreover, the wandering behaviour was re-established (Fig. 6 (d) and (f).) after a human mediation suspending the autonomous navigation and teaching some safe navigation actions and then returning to autonomous mode. Notably, the cases where the robot stuck occurred only when we set the memory granularity to a large value. It is because the robot capacity to distinguish between similar contexts was biased by memory granularity as shown Table 2.

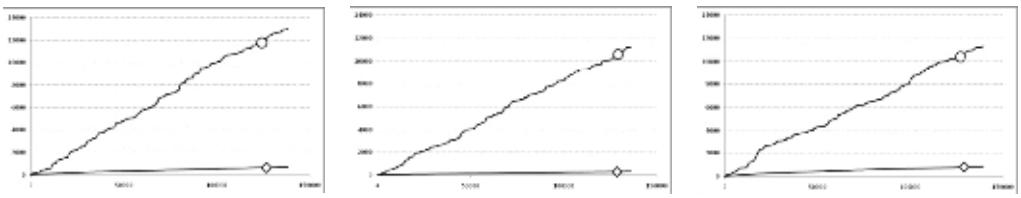
Units into Perception Memory $\mathcal{M}^p$						
$e_{max}^{\mathcal{M}^p}$	$\alpha_r = 0.1$ $\alpha_p = 0.1$	$\alpha_r = 0.1$ $\alpha_p = 0.2$	$\alpha_r = 0.2$ $\alpha_p = 0.1$	$\alpha_r = 0.3$ $\alpha_p = 0.3$	$\alpha_r = 0.3$ $\alpha_p = 0.5$	$\alpha_r = 0.5$ $\alpha_p = 0.3$
20	73,210	78,204	80,779	80,347	82,495	79,219
40	6,798	6,921	6,835	6,977	6,798	6,870

Table 2. Number of memory units used to represent the robot perceptions

The relationship between the memory granularity and robot capacity to learn to distinguish safe navigation actions from unsafe, becomes clear confronting the data reported in Table 2 with the graphs in Fig. 4 and Fig. 5. Then found a suitable value for parameter e_{max} of GTLSOM is a main job to obtain a suitable memory of perceptions. Moreover, the presented results indicates high values of α_p weaken the synapses so that the algorithm is unable to recover them and the learning process stuck in a local minima. Whereas setting up α_p values lightly greater than α_r seem to produce better results (Fig. 4 (b) and (e)).

5.5.3 Test set B – navigating in 3D

For this test set, the robot was released to navigates in 3D employing actions to floating up and down besides that used to navigate in 2D. Then we have also considered the perceptions originated from sensors on Top and Bottom of robot. The parameter of memory granularity was set as $e_{max}^{\mathcal{M}^p} = 20$ because of the results reported in Subsection 5.5.2. The parameters α_r and α_p were set using the pairs (0.1; 0.1), (0.1; 0.2) and (0.2; 0.1). Other parameters were set as in Subsection 5.5.2 and the time of test was established as 150.000 attempts of navigation. The Fig. 7 shows that results are similar to those obtained with robot constrained to navigate in two dimensions (Fig. 4).



(a) $\alpha_r = 0.1,\ \alpha_p = 0.1$ (b) $\alpha_r = 0.1,\ \alpha_p = 0.2$ (c) $\alpha_r = 0.2,\ \alpha_p = 0.1$

Fig. 7. Robot navigation behavior if positioned frontwards an obstacle while navigating in 3D

5.5.4 Test set C – navigating into similar environments

In this test set, the objective was asset the knowledge reuse to navigate in another similar environment. Firstly, the robot was placed into an initial environment and left to acquire navigation skills learning about contexts and the feasibility of navigation actions. Further, the robot was removed and reinserted into a similar environment for two times without restart its memory. The parameter set was established with $e_{max}^{\mathcal{M}^p} = 20,\ \alpha_r = 0.1, \alpha_p = 0.1$ and others were set with values used in test set A (Subsection 5.5.2). The time of each phase was set to one hundred thousand navigation actions. The Fig. 8 shows percent of collisions for each phase. Note the spikes following the reinsertion indicating that the robot found unknown contexts which are learned rapidly. Moreover, the percent of collisions remains decreasing indicating the success of knowledge reuse.

6. Conclusion

We have discussed briefly the issue of topology learning and topology preserving maps and its applicability to surface reconstruction and to computational intelligence. Thereafter we have presented the GSOSM model which was developed considering the surface reconstruction problem. Then we have updated GSOSM algorithm to work with space $\mathbb{R}^n$,

$n \geq 3$. Specifically we have modified the adaptation step by introducing a method to obtain a suitable $\mathbb{R}^n$ normal vector. Moreover, we have added an activation function to report the nodes activation in a normalised way. Then we called the resulting model as Growing Topology Learning Self-Organizing Map (GTLSOM) and we attempt to give a proof of its topology preservation capability.

Further, we have used GTLSOM as a tool to produce a basic memory which was employed to enable a robot with a remembrance capability. Basically, in this aplication GTLSOM acts as memory by constructing a map grouping patterns into nodes with a receptive field (units of memory). In our model the memory inputs are representations of robot perceptions about its surrounding environment. The knowledge about contextual feasibility of each navigation action is stored into synapses linking units of memory and actions. Then, the incoming perceptions are classified to nodes firing a remembrance process that subsidizes a value judgment for each possible action to be selected. We also have reported the obtained results which are indicatives of success of our model. Moreover note that in our model, the basic memory does not know the origin of each perception as well as what happens out of its boundaries. Therefore, the basic memory could be shared with other cognitive process.

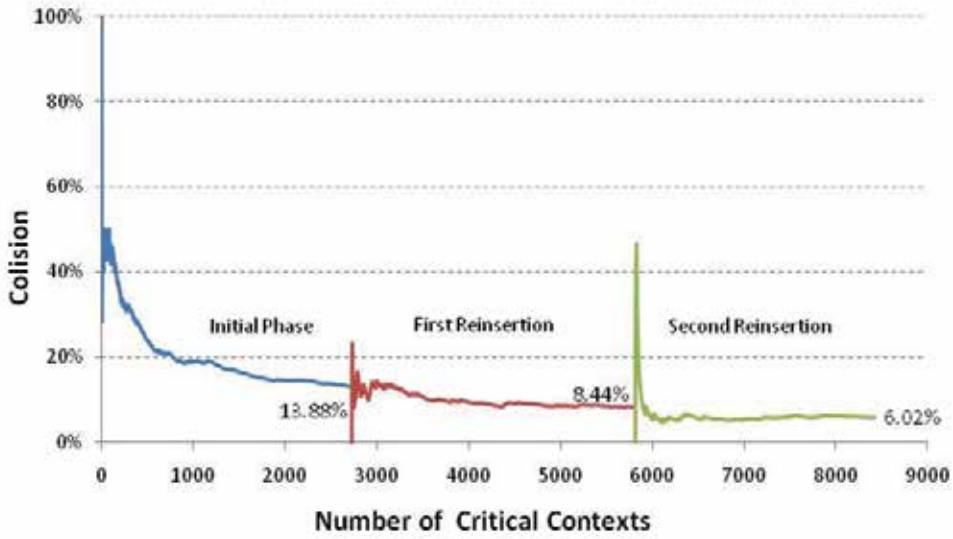


Fig. 8. Robot navigation behavior if moved from an environment to other similar one

7. References

DalleMole, V. L. & Araújo, A. F. R. (2010a) Growing Self-Organizing Surface Map: Learning a Surface Topology from a Point Cloud, Neural Computation, Vol. 22, No. 3, pp. 689-729.

DalleMole, V. L. & Araújo, A. F. R. (2010b) A Novel Topological Map of Place Cells for Autonomous Robots, To appear in: *Proceedings of 20th International Conference on Artificial Neural Networks*, Thessaloniki, Greece, September 15-18, 2010, Springer Academic Publishers.

DalleMole, V. L. & Araújo, A. F. R. (2008) The Growing Self-Organizing Surface Map, *Proceedings of International Joint Conference on Neural Networks IJCNN´08*, pp. 2061 – 2068, ISBN: 978-1-4244-1820-6, Hong Kong.

Jockusch, J. & Ritter, H. (1999). An instantaneous topological mapping model for correlated stimuli. In: *International Joint Conference on Neural Networks IJCNN'99*, Vol. 1, pp. 529-563, ISBN: 0-7803-5529-6, Washington.

Kaebling, L. P. Littman, M. L. & Moore, A. W. (1996) Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*,Vol 4, pp. 237-285.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. In: *Biological Cybernetic,* Vol. 43, pp. 59-69.

Martinetz, T., Schulten, K. (1994) Topology Representing Network. Neural Networks, Vol. 7, No. 3, pp. 507-522.

Martinetz, T.; Schulten, K. (1991). A "Neural-Gas" Network Learns Topologies. In: *Artificial Neural Networks*, T. Kohonen, K. Makisara, O. Simula and J. Kangas, Elselvier Science, pp. 397- 402, North-Holland.

Sloughter, D. (2001) *The Calculus of Functions of Several Variables*, Furman University, 260p.

Sutton, R. S. & Barto, A. G. (1998) *Reinforcement Learning: An Introduction*. Mit Press.

Vilmann, T. (1999) Topology preservation in self-organizing maps, In: *Kohonen maps*, Erkki Oja and Samuel Kaski, editors, Elselvier Science, pp. 279-291.

Zernike, F. (1934) *Physics*, pp. 1- 689.

32

Is it Visible? Micro-artefacts' Nonlinear Structure and Natural Formation Processes

Dimitris Kontogiorgos[1] and Alexandros Leontitsis[2,3]
[1]*Weiner Laboratory, American School of Classical Studies at Athens, Athens,*
[2]*Department of Education, University of Ioannina, 45110 – Dourouti, Ioannina,*
[3]*Center for Research and Applications of Nonlinear Systems, University of Patras, 26500 – Rio, Patras,*
Greece

1. Introduction

Micro-artefacts (i.e., cultural particles smaller than 2 mm in diameter), due to their abundance and incorporation in an archaeological deposit constitute a significant part of the cultural particles present, and their analysis can assist the archaeological interpretation (Stein, 1987). The analysis of micro-artefacts is extensively complex due to the different micro-artefact categories that may appear in an archaeological context and also because of the number of different cultural (and natural) formation processes that may have been involved in the creation of characteristics specific to an archaeological context.
Recently, a non-linear data analysis technique (S-SOFM) was applied on micro-artefact data. The technique revealed hidden structures among the data that linear methods were unable to classify and also offered an additional way to unfold cultural information encoded within them (i.e., micro-artefacts), and thereby, observe the dynamics of cultural factors in site formation processes (Kontogiorgos *et al*., 2007). The spherical self-organizing feature map method was implemented in this study to perform the initial pattern searching operation in a sample of 4-dimensional micro-artefact data shaped *only* by the effects of natural formation processes (hereafter: N.F.P.), and so further explore the developed visualization tool as a means to identify patterns in micro-artefacts prior to their detailed analysis. An example is given on micro-artefact data from the colluvial deposit, covering the theatre of the Hellenistic period, located outside the city walls of the Hellenistic city of Gitana in the Thesprotia region-Epirus-NW Greece (Kontogiorgos and Preka, 2009). Section 2 briefly describes how the spherical self-organizing map creates a 3D visual or graphical representation of the data. Section 3 applies the method on a sample of 4-dimensional micro-artefact data. General findings and concluding remarks are presented in Section 4.

2. Spherical self-organizing feature map

The Self-Organizing Feature Map (SOFM), introduced by Kohonen (1981), maps n-dimensional data into a low-dimensional space. The spherical SOFM (Sangole, 2003) the

low-dimensional space is a tessellated sphere that is formed by subdividing an icosahedron. Every vertex on this sphere is a strategic location of an n-dimensional vector that represents an ensemble of similar data vectors which are assigned to the vector during the mapping operation. It is therefore necessary to visually enhance variations in the data using the physical attributes of the mapping lattice. The benefit of a spherical lattice in the implementation of the SOFM is that the enclosed space can be used to generate a 3D visual representation of some physical aspect of the n-dimensional data.

Conventional implementation of the SOFM method have used a 2D lattice as the low-dimensional space, and associations in the data are visualized by means of a terrain map, wherein elevation represents some aspect of the vector(s) at that location (Vesanto, 1999; Ultsch and Siemon, 1990). Relative similarity between data vectors mapped into the sphere can be visualized by introducing distortions in the sphere accompanied by changes in the colour. Informative characteristics of the data are reflected as distortions and colour gradations on the surface of the sphere. The formulation of these measures is a non-trivial task and often application dependent. The measures reflect desired data correlations (either linear or non-linear) and must be defined by the researcher who is familiar with the underlying data set. It is this aspect of the S-SOFM that differs from existing literature about the self-organizing feature map. The S-SOFM utilizes the spherical lattice of the SOFM space to generate a visual form of the clustered data that is more intuitive and easy to perceive. A visual form of the data is created by scaling the radial distance of the vertices on the sphere in proportion to a measure characterizing some physical aspect of the data. Examples illustrating the various implementations of the spherical SOFM on different data and the use of possible measures to create spherical SOFM graphical representations are discussed in Sangole (2003) and Sangole and Knopf (2003).

3. Implementation of the S-SOFM on micro-artefact data

3.1 Previous work on micro-artefacts from the site

The systematic excavation in the theatre of the Hellenistic period (ca. 330 B.C-167 B.C) at Gitana (Thesprotia region–Epirus-NW Greece) during five seasons (1996-1997 and 2005-2008) has brought into light the auditorium of the theatre below a thick (ca. 1.25m-1.50m) colluvial deposit. The source of the colluvial deposit was thought to be sediments and cultural materials eroded from the abandoned Hellenistic city of Gitana, once expanded on top of the theatre (Kontogiorgos and Preka, 2009). Although is impossible to assess whether micro-artefacts ended up in the deposit as micro-materials or were generated after deposition; however, since the identified micro-artefact types (i.e., shell, bone, burnt clay) come from fragile or size unstable materials, their density trends (i.e., peaks and irregularity) might have been produced post-depositionally from the effects of the different N.F.P. (i.e., erosion, in situ weathering and translocation of smaller cultural sedimentary particles) that affected the larger macro-artefacts (Sherwood *et al*., 1995) present in the deposit.

In this case, since anthropogenic activity was absent from the site after abandonment, their variability most likely depicts the intensity of these types of formation processes (i.e., N.F.P.) and possibly the time span capable of producing variable micro-artefact concentrations (Kontogiorgos and Preka, 2009).

3.2 Laboratory procedures

A total of forty four (44) sediment samples weighing ca 1000g each, were collected from five (5) columns, mostly at 10cm vertical intervals, providing good coverage across the exposed stratigraphy of the deposit, and were labeled according to depth. The laboratory procedure used two divisions of the phi (ф) scale: - 2.00ф and 0ф. Contents of the bulk samples were passed through a stack of 4mm (-2.00ф) and 1mm (0ф) sieves. The material retained in the 1mm sieve created the sub-sample that was processed for microartifacts and an optical microscope was used for identifications. To avoid damaging the artefactual contents (e.g., shell, bone) there was no pretreatment for removal of organic matter or carbonate. The sub-sample was saturated with 1% sodium hexametaphoshate and washed through the 1mm sieve to separate the sand fraction from the silt/clay fraction. For each sub-sample, 500 particles were point-counted.

The identified micro-artefact types were: Microfragments of Burnt Clay, Microbone, Microshell. To deal efficiently with the large numbers of samples derived from the contexts, and reduce the processing time, the point-counting procedure had to be applied. The procedure for determining the proportions of compositional types follows the one described by Stein and Telster (1989). A small fraction of the sub-sample was poured gradually, into a glass petri-dish, below which was attached a piece of graph paper of no greater than 1cm graph intervals. The particles are spread evenly across the grid. While looking through the optical microscope, the particles located in one grid unit were counted according to compositional types.

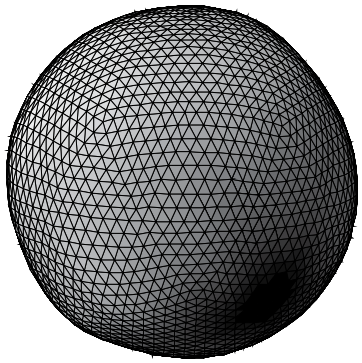
To improve the identifications and to observe more accurately the measurement error, 100 particles were counted and recorded each time, until a total of 500 particles were examined, since in the previous exercise a good stabilisation of the point estimation between 250 and 500 counted particles was achieved (Kontogiorgos and Leontitsis, 2005;2009).

3.3 Seeing the unseen! N.F.P. creating micro-artefact patterning

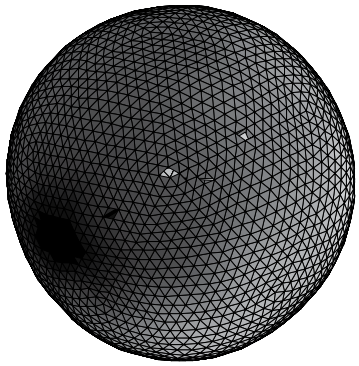
The construction of the S-SOFM graphical representation was based on a database of 44 three-dimensional records, each dimension representing a micro-artefact category. Every row represented the point-counting results (Kontogiorgos and Leontitsis, 2009). A spherical-SOFM graphical representation was created as described in Sangole (2003), Sangole and Knopf (2003) and Leontitsis and Sangole (2005). The glyph was generated after 20 epochs (training cycles). Three views of the resulting S-SOFM graphical representation are illustrated in Fig. 1 showing the formation of a distinct black region. A nonlinear micro-artefact structure lies within this statistical space which can be distinguished into a separate sub-structure. The spherical-SOFM pattern recognition procedure therefore serves as the initial step in the analysis of the multidimensional micro-artefact data. It provides a comprehensive preliminary visual representation of inherent non-linear characteristics in data. In this example one meaningful component was revealed, which appeared to be the determinant for the constitution of the analyzed data set.

The results of this study suggest that the proposed implementation of the spherical-SOFM non-linear method is able to recognize and to provide a visual representation of micro-artefact patterns prior to performing any further analysis on the data. It also depicts the ability of the N.F.P. to create patterns in micro-artefacts. The implication is that these complex multidimensional data (i.e., micro-artefacts) should be used with care due to the many factors that may account for the observed pattern and should not be simply used to

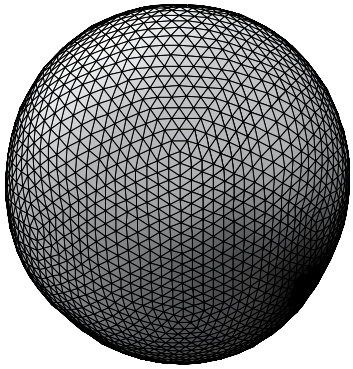
reflect 'noise' in larger artefacts (e.g., Dunnell and Stein, 1989). Finally, this exercise enhances attempts for developing interpretations on micro-artefact patterning by providing strong pattern recognition.



(a)



(b)



(c)

Fig. 1(a-c). Three views of the S-SOFM graphical representation showing the formation of distinct black region corresponding to the micro-artefact data

The next step will be to understand the reasons for this pattern by further investigating either the proportional relationships between micro-artefact types (Kontogiorgos, 2008) and/or by a complete geoarchaeological analysis of archaeological sediments which has shown concentration (i.e., peaks) of micro-artefacts in fine sediment attributable possibly to the effects of N.F.P. to produce such concentration of micro-artefacts in an archaeological deposit lacking any anthropogenic activity (Kontogiorgos and Preka, 2009).

4. Conclusions

A non-linear data analysis technique, the spherical self-organizing feature map, was applied on micro-artefact data. This technique revealed patterns among the data created by N.F.P further suggesting care when interpreting micro-artefacts. The implementation of the method needs to be explored in other site contexts and further tested. Given the abundance of large data sets of micro-artefacts, it will be a useful visualisation tool for the researcher to facilitate a quick search and identification of possible micro-artefact patterns in soil samples prior to performing any detailed analysis of the complex multidimensional data.

5. References

Dunnell, R.C. & Stein, J.K. (1989). Theoretical issues in the interpretation of microartifacts. *Geoarchaeology* 4 (1), pp. 31-42.

Kontogiorgos, D. & Leontitsis, A. (2005). Micro-artefacts weight estimation by Genetic Algorithm minimisation. *Journal of Archaeological Science* 32, 1275-1282.

Kontogiorgos, D., Leontitsis, A. & Sangole, A. (2007). Telling a non-linear story: the exploration of micro-artefacts' non-linear structure. *Journal of Archaeological Science* 34, pp. 1532-1536.

Kontogiorgos, D. (2008). Geoarchaeological and Microartifact Analysis of Archaeological Sediments. A Case study From a Neolithic Tell Site in Greece. Nova Science Publishers, Inc., New York.

Kontogiorgos, D. & Leontitsis, A. (2009). Test again for the optimum! Confirming the Use of Genetic Algorithm Minimization on Microartifact Weight Estimation. In: *On Site Geoarchaeology on a Neolithic Tell Site in Greece: Archaeological Sediments,Microartifacts and Softwear Development*, Kontogiorgos, D. (Ed)., Nova Science Publishers, Inc., New York.

Kontogiorgos, D. & Preka, K. (2009). From Neolithic to Hellenistic. A Geoarchaeological Approach to the Burial of the a Hellenistic Theatre: The Evidence from Particle Size Analysis and Microartifacts. In: *On Site Geoarchaeology on a Neolithic Tell Site in Greece: Archaeological Sediments,Microartifacts and Softwear Development*, Kontogiorgos, D. (Ed)., Nova Science Publishers, Inc., New York.

Kohonen, T. (1981). Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43, pp. 59-69.

Leontitsis, A.& Sangole, A.P. (2005). Estimating an optimal neighborhood size in the spherical self-organizing feature map. *International Journal of Computational Intelligence* 2, pp. 94-98.

Sangole A. (2003) Data-driven modeling using spherical self-organizing feature maps, PhD thesis, University of Western Ontario, Canada; Universal Publishers (ISBN 1-58112-319-1).

Sangole, A. & Knopf, G.K. (2003). Visualization of random ordered numeric data sets using self-organized feature maps. *Computers and Graphics* 27, 963-976.

Stein, J.K. (1987). Deposits for archaeology, Advances in Archaeological Method and Theory 11, pp. 337–392.

Stein, J.K. & Telster, P.A. (1989). Size distributions of artefact classes: combining macro- and micro-fractions. *Geoarchaeology* 4 (1), pp. 1-30.

Sherwood, S.C. (2001). Microartifacts. In: *Earth Sciences and Archaeology*, P.Goldberg, V.T. Holiday, R. Ferring (Eds.), Academic/Plenum Pubs, pp. 327–351.

Sherwood, S.C., Simek, J.F. & Polhemus, R.R. (1995). Artifact size and spatial process: macro- and microartifacts in a Mississipian House. *Geoarchaeology* 10 (6), pp. 429-455.

Vesanto, J. (1999). SOM-based data visualization methods. *Journal of Intelligent Data Analysis* 3, pp. 111-126.

Ultsch, A., Siemon, H.P. (1990). Kohonen's self-organizing feature maps for exploratory data analysis. *Proceedings of the International Neural Network Conference.* Dordrecht, The Netherlands, pp. 305-308.

33

Self-Organization of Object Categories in a Cortical Artificial Model

Alessio Plebe
Department of Cognitive Science, University of Messina
Italy

1. Introduction

Explaining the human capacity to construct mental categories is a central question in cognitive science, yet how reality is perceived and compartmentalized by our mind has puzzled philosophers since Aristotle. Today, categorization is believed to be a strategy humans have in common with many animals. In most species, individuals have the opportunity to directly experience only a tiny fraction of the entities, objects, features and events offered by their environments. By relying on categorization, organisms become able to determine which things in the world belong together or are alike in some way, therefore reacting in an appropriate manner, thanks to the previous experiences had with other instances of the same category.

The purpose of this chapter is to investigate computational models that can simulate mechanisms underlying the ability to develop categories. We argue that a neural mechanism for detecting and coding recurrent coincidences in stimuli could be the key component in our brain's ability to build the categories it uses to organize reality. The mathematical framework that has given the best interpretation so far on how signals with high levels of autocorrelation affect networks of neurons with local coincidence detection and coding abilities, is the one referred to as *self-organization*.

Being categorization a phenomena appearing at so many different cognitive levels, and that is pervasive in so many different modalities, we doubt that it could rely upon a single mechanism. Moreover, we believe that humans unlike any other animal reshape their categories and build new ones, by virtue of the information provided to them by language as soon as they have developed the most rudimentary understanding of it. Nevertheless, we do not see reasons for posing that the pre-linguistic development of categories in infants should happen as a result of mechanisms different from those found in other species. Even if we are convinced that different mechanisms might contribute, the basic mechanism briefly discussed is one we consider to play a key role.

1.1 The problem of mental categories

The research on categorization in humans can be traced back to traditional domains. In philosophy the objective of epistemology has included explaining the acquisition of knowledge, with empiricists attempting to formulate explanations in terms of mental mechanisms (Locke, 1690; Hume, 1739). It was within the field of psychology, however, that the search for this sort of mechanism became a central issue, employing the strategy of experimental analysis of the ontogenesis of categories in infants. While holding in high regard

the theoretical frameworks proposed within those two traditional disciplines in the past, we believe that the possibility of identifying the basic mechanisms of categorization is today more feasible thanks to recent advances in some specific domains. In particular, we are building on the following:

- Developmental psychology, that thanks to improved testing methodologies has produced an impressive amount of research on early categorization in the last 20 years.
- The neurobiology of learning, that has opened the door to the understanding of the mechanisms underlying the representational power of neural circuits.
- Computational models of emerging representations, that have contributed to understanding the modalities and constraints by which experiential inputs build representations in the brain.

Basing ourselves on evidence that has come from these three fields, we argue that a mechanism implementing coincidence detection of stimuli, and coding recurrent coincidences by self-organization, could be essential for the brain to build mental categories.
There are at least three crucial aspects of this mechanism. One is the special sensibility of organisms, in detecting that two or more elements in the stream of stimuli from the environment, happen with remarkable synchrony. A second is the ability of the organism to code in its nervous system traces of repeated coincidences of the detected elements. One effect of this internal representation, is that subsequent experiences of just one (or a few) of the coded elements, might elicit the activation of the internal representation of the other(s). Finally, in a population of neurons every reinforcement of the binding between two or more elements, should be compensated by reduction of other antagonistic elements. The combination of self-reinforcing interaction with competitive compensation may give rise, on the scale of cortical maps, to the emergence of self-organized representations corresponding to mental categories.

1.2 Computational models of categorization

The first attempt to simulate lexical semantics by means of self-organization was done by Ritter & Kohonen (1989). It was a very abstract model, in which names of animals and certain binary attributes (like “has feathers” or “likes to swim”) were coded in vectors.
This approach was extended by Miikkulainen (1997), in a model that combined three SOM maps, two for orthography and phonology, and a semantic feature map. Associations between them were formed, based on co-occurrence of the lexical symbol and its meaning. The model was able to simulate dyslexic and category-specific aphasic impairments similar to those observed in human patients. More recently Li et al. (2004) proposed a model of early lexical categorization that combines two self-organizing maps as semantic map and phonological map, connected via associative links. Their model reproduced aspects of a growing lexicon during development. Mayor & Plunkett (2010) have also explored specific aspects of early lexical acquisition, using standard self-organizing maps. This model reproduces interesting aspects of language learning, like slips of the tongue and mispronunciation effects.
All these models, however, lack correspondence between their mathematics and how brain computations are performed, in particular, there is no account on how, on a large scale, processes relevant to categorization are distributed in the cortex.
The model here proposed is an attempt to simulate early phases of lexical categorization of visual objects, exploiting mathematical implementations of the self-organizing principle, in a way that takes into account the real structure of the cortex, with its pathways and hierarchies

of processes. In the next section, the core concept of the mechanism supporting categorization will be introduced, together with its available implementations and relevant theoretical issues. Section §3 will describe the model, with results shown in §4.

2. Artificial networks with self-organizing properties, and category formation

Computational models have been a valid complement to neurophysiology and neuroimaging in the enterprise of understanding how our brain works. Models, in abstracting biophysical phenomena at different levels, can help bridge neurophysiology with behavior and cognition. However, in the past the guiding principles of modeling cognitive functions were strongly based on the epistemological value of the algorithmic design, corroborated by the philosophical view of the mind as a computing machine, that prevailed in the 80's and '90s (Fodor, 1981; 1983). The serious weakness of this early form of computationalism was the risk of solving problems in our own head and then subsequently creating specific algorithms, that can be a far cry from how the brain solves the same problem.
Today, computational neuroscience has grown, reaching solid theoretical foundations Dayan & Abbott (2001), making it possible to constrain models, especially of low-level vision, in order to get relatively close to real brain mechanisms, avoiding the errors made by previously unconstrained computational models. However, the adherence to the known chemical and electrical processes of the nervous system is necessarily inverse to the degree of complexity of the problem at hand. In the case of categorization, even if limited to objects with visual salience, complexity is very high, and therefore, the replication of biology cannot be detailed. This means that the risk, mentioned above, of creating algorithms that are unrelated to actual brain functions, is always present. A good antidote for this, is to focus models on the repertoire of basic principles underlying cortical organization, and to study the ontogenesis of complex functions by implementing these principles in a simulated environment. This way of proceeding is characteristic of research done within a recent framework called neuroconstructivism (Quartz & Sejnowski, 1997; Westermann et al., 2007).

2.1 The mathematics of cortical development

In our view category formation is directly correlated with the development of cortical maps, in the meaning originally introduced by Mountcastle (1957), as a portion of the cortex unified functionally by the consistent responsiveness of its cells to contiguous sensorial stimulation. At this level, we consider the effect of being exposed to possible coincidences in a multitude of signals. In a sense, categories are ways of coding in neural aggregates, which signals co-occur or are co-incidental enough to be representative of the key traits of the objects or events that caused those signals. One of the most fruitful mathematical frameworks for simulating cortical development driven by the degree of autocorrelation of the incoming signals is the concept of self-organization. It is a vast topic, related to dynamic systems theory, with a long tradition in physics, physical chemistry, and astronomy (Haken, 1978). In general this framework studies the spontaneous formation of organized patterns in systems consisting of a large number of interacting elements, and initially in a relative undifferentiated state (Ball, 1999). In most organizing systems two fundamental facts are that global order can arise from local interactions, and that interactions work in two opposite directions: that of enhancing the degree of similarity (that in some systems can be interpreted intentionally as cooperation) between some elements, or by the differentiation between some other elements (intentionally read as competition). In the field of neural circuit development it was first introduced by von der Malsburg (1973), for modeling how the visual cortex can spontaneously develop its

mature organization, and was based exclusively on the local interaction of neurons, ruled by Hebb's principle. In the case of the development of orientation-sensitive patterns in the primary cortex, the model was based on the following system of differential equations:

$$\frac{\partial y_i}{\partial t} = -\alpha y_i + \vec{k}_i \cdot f(\vec{y}_i) + \vec{w}_i \cdot f(\vec{z}_i) + x_i, \quad (1)$$

$$\frac{\partial w_{i,j}}{\partial t} = \eta f(y_i) \left(f\left(z_j\right) - w_{i,j} \sum_{l \in \mathcal{Z}_i} f(z_l) \right), \quad (2)$$

where y and z are the activities of neurons in different layers of the same cortical area, the former in layer IV, where typically thalamocortical afferents project, the latter in other layers, where lateral intercortical interactions take place. $w_{i,j}$ is the connections strength between z_j and x_i, and the vector $\vec{w}_i$ is composed by all connections of intercortical neurons projecting to i. The vectors $\vec{y}_i$ and $\vec{z}_i$ are the activations of all neurons belonging to columns where intercortical connections with neurons i exist. Using the same convention for the corresponding scalars, $\vec{y}_i$ refers to cells in layer IV and $\vec{z}_i$ to cells in other layers. The constant kernel vector $\vec{k}$ modulates the activation of neighbors with the distance from i, typically with a Gaussian difference. The function f is the monotonic squeezing non-linearity, typically a sigmoid. In equation (2) the first term corresponds to Hebbian growth in response to coinciding activity on the presynaptic and postsynaptic sides of the connection between i and j. The second term, where $\mathcal{Z}_i$ is the set of cells out of layer IV connected to i, performs a competitive effect, ensuring that the sum of synaptic strengths converging on position i is kept constant at 1.

2.2 The SOM implementation

Kohonen (1982; 1995) has made the mechanism of self-organization popular, with his very simple but efficient SOM (*Self-Organizing Maps*) model.

The learning rule is on a *winner-takes-all* basis: if the input data are vectors $\vec{v} \in \mathbb{R}^N$, the SOM will be made of some M neurons, each associated with a vector $\vec{x} \in \mathbb{R}^N$ and a two dimensional (in vision applications) coordinate $\vec{r} \in \{< [0,1],[0,1] >\} \subset \mathbb{R}^2$. For an input v there will be a winner neuron w satisfying:

$$w = \arg \min_{i \in \{1,\ldots,M\}} \{\|\vec{v} - \vec{x}_i\|\}. \quad (3)$$

The winner-takes-all mechanism is a mathematical substitution for the effect of lateral connections, which play a fundamental role in the emergence of complex response functions in the cortex. Once the winner is identified, neural vectors are updated during training using:

$$\Delta \vec{x}_i = \eta e^{-\frac{\|\vec{r}_w - \vec{r}_i\|^2}{2\sigma^2}} (\vec{v} - \vec{x}_i), \quad (4)$$

where η is the learning rate, σ the amplitude of the neighborhood affected by the updating. Often both η and σ are decreasing functions of the training epochs.

As said in §1.2 the SOM soon proved to be very effective in modeling categorization, at the lexical level as well, it is however, a significant departure from the behavior of biological cortical circuits. The winner-takes-all mechanism works as a mathematical substitution for the effect of lateral connections, but only assuming fixed connections and uniform neighborhoods. The input lacks a receptive field structure, therefore SOM cannot be used

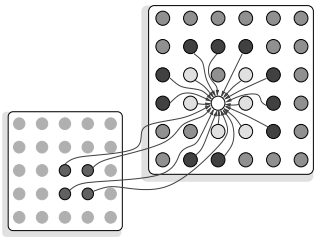


Fig. 1. The LISSOM architecture with lateral connections for one neuron (in white). Excitatory links are in pale gray and inhibitory links in dark gray.

as a building block in a hierarchy of cortical maps. However, there are concepts in SOM categorization that we believe to be relevant to brain structure as well. One is the mapping of meaning into two dimensions. The question concerning whether our disposition to better understand patterns depends on their being presented in a two dimensional manner due to the bidimensional structure of our cortex remains an open question (Chklovskii & Koulakov, 2004). Kohonen (1998) clearly defended the biological relevance of SOM-like maps in the cortex and proposed a modification of the SOM algorithm in order to relate the adaptation processes to physiological evidence in the brain, still preserving the winner-takes-all rule (Kohonen & Hari, 2000).

Several new mathematical formulations have been provided for learning mechanisms in populations of neurons in cortical maps, that still achieve competitive self-organization, but without the SOM winner-takes-all mechanism. Wallis & Rolls (1997) designed a model with explicit lateral inhibitory effect on neighbor units. Learning in the network is achieved with a special modification of Hebb's rule, in which a decaying trace of previous cell activity is kept. The trace rule can be interpreted as a way of detecting regularities between appearances of objects in short time sequences, therefore coincidental features that regularly happen when the same object is under view. This architecture has been successful in recognizing categories of artificial objects and faces (Rolls, 1992; Wallis & Rolls, 1997). This architecture is closer to the reality of cortical maps than the SOM is, but still, due to the fixed modeling of inhibition, and the lack of excitatory connections, may miss the full set of functional and developmental phenomena of a real cortical map.

2.3 The LISSOM architecture

An attempt to implement flexible and modifiable lateral connections, while keeping the architecture relatively simple, is represented by the model developed by Sirosh & Miikkulainen (1997) called LISSOM (*Laterally Interconnected Synergetically Self-Organizing Map*). In this model, each neuron is not only connected with the afferent input vector, but receives excitatory and inhibitory inputs from several neighbor neurons on the same map, as shown in Fig. 1

The activation level x_i of a neuron i at a certain time step k will now be given by:

$$x_i^{(k)} = f\left(\frac{\gamma_{\mathrm{A}}}{1+\gamma_{\mathrm{N}}\vec{I}\cdot\vec{v}_{r_{\mathrm{A}},i}}\vec{a}_{r_{\mathrm{A}},i}\cdot\vec{v}_{r_{\mathrm{A}},i} + \gamma_{\mathrm{E}}\vec{e}_{r_{\mathrm{E}},i}\cdot\vec{x}_{r_{\mathrm{E}},i}^{(k-1)} - \gamma_{\mathrm{H}}\vec{h}_{r_{\mathrm{H}},i}\cdot\vec{x}_{r_{\mathrm{H}},i}^{(k-1)}\right), \tag{5}$$

where the vectors $\vec{x}_{r_{\mathrm{E}},i}^{(k-1)}$ and $\vec{x}_{r_{\mathrm{H}},i}^{(k-1)}$ are the activations of all neurons in the map, where a lateral connection exists with neuron i of an excitatory or inhibitory type, respectively. Their fields are circular areas of radius, respectively, r_{E}, r_{H}. Vectors $\vec{e}_i$ and $\vec{h}_i$ are composed by

all connection strengths of the excitatory or inhibitory neurons projecting to i. The vectors $\vec{v}$ and $\vec{x}_i$, as before, are the input and the neural code. The scalars γ_{A}, γ_{E}, and γ_{H}, are constants modulating the contribution of afferents. The map is characterized by the matrices $\mathbf{A}, \mathbf{E}, \mathbf{H}$, whose columns are all vectors $\vec{a}$, $\vec{e}$, $\vec{h}$ for every neuron in the map. The scalar γ_{N} controls the setting of a push-pull effect in the afferent weights, allowing inhibitory effects without negative weight values. Mathematically, it represents dividing the response from the excitatory weights by the response from a uniform disc of inhibitory weights over the receptive field of neuron i. Vector $\vec{1}$ is just a vector of 1's of the same dimension of $\vec{x}_i$. The function f is any monotonic non-linear function limited between 0 and 1, in this model is implemented as a piece-wise linear sigmoid.

The final activation value of the neurons is assessed after a certain settling time K, typically about 10 time steps. It is easy to recognize the resemblance of (5) with the original formulation of von der Malsburg in (1). All connection strengths adapt according to the general Hebbian principle, but include a normalization mechanism that counterbalances the overall increase of connections of the pure Hebbian rule. The equations are the following:

$$\Delta\vec{a}_{r_{\mathrm{A}},i} = \frac{\vec{a}_{r_{\mathrm{A}},i} + \eta_{\mathrm{A}} x_i \vec{v}_{r_{\mathrm{A}},i}}{\left\|\vec{a}_{r_{\mathrm{A}},i} + \eta_{\mathrm{A}} x_i \vec{v}_{r_{\mathrm{A}},i}\right\|} - \vec{a}_{r_{\mathrm{A}},i}, \tag{6}$$

$$\Delta\vec{e}_{r_{\mathrm{E}},i} = \frac{\vec{e}_{r_{\mathrm{E}},i} + \eta_{\mathrm{E}} x_i \vec{x}_{r_{\mathrm{E}},i}}{\left\|\vec{a}_{r_{\mathrm{E}},i} + \eta_{\mathrm{E}} x_i \vec{x}_{r_{\mathrm{E}},i}\right\|} - \vec{e}_{r_{\mathrm{E}},i}, \tag{7}$$

$$\Delta\vec{h}_{r_{\mathrm{H}},i} = \frac{\vec{h}_{r_{\mathrm{H}},i} + \eta_{\mathrm{A}} x_i \vec{x}_{r_{\mathrm{H}},i}}{\left\|\vec{h}_{r_{\mathrm{H}},i} + \eta_{\mathrm{A}} x_i \vec{x}_{r_{\mathrm{H}},i}\right\|} - \vec{h}_{r_{\mathrm{H}},i}, \tag{8}$$

where $\eta_{\{\mathrm{A,E,H}\}}$ are the learning rates for the afferent, excitatory, and inhibitory weights. The normalization in equations (6), (7), (8), is a computational account of the biological phenomena of homeostatic plasticity (Turrigiano & Nelson, 2004).

The LISSOM model has been extensively used in vision studies, for modeling several low-level visual phenomena (Miikkulainen et al., 1997), the simulation of aftereffects (Bednar & Miikkulainen, 2000), perceptual grouping (Choe & Miikkulainen, 2004), face detection (Bednar & Miikkulainen, 2003), color response development (Bednar et al., 2005), motion direction selectivity (Bednar & Miikkulainen, 2006), and 3D object recognition (Plebe & Domenella, 2006; 2007). A comprehensive review of LISSOM applications is in (Miikkulainen et al., 2005). The LISSOM framework has been used by Plebe (2006) in a model of the ventral visual cortex, to simulate learning invariance in vision. Invariance is a kind of perceptual categorization, with the final goal of recognizing shapes as such, despite large changes in their appearance on the sensory surface. In this simulation it is simply learned, by detecting coincidences of activation patterns in lower visual maps, when belonging to same shapes. The model has been extended to the recognition of real objects (Plebe & Domenella, 2007), categorized by their visual features. In a more complex model, that includes visual and auditory pathways, Plebe et al. (2007b) have simulated the early acquisition of object names. The model was exposed to the coincidence of objects in its field of view, and the utterance of the category the object belonged to. A similar model was able to reproduce to some extent, some of the phenomena characteristic of early word learning, such as fast mapping and shape bias (Plebe et al., 2007a).

One difficulty in using LISSOM instead of SOM for studying categorization is in losing the interpretation of the role of each neuron in the map, that in the SOM can be achieved

immediately by labeling (see §4.3.2). The point is that LISSOM resembles a biological cortical map more closely than SOM, sharing with it the problem of the interpretation of neural firing, especially in higher-level maps. Research in this direction has recently drawn on the idea that the power of representing categories in cortical circuits lies in the combinations of activities of many columnar units. This concept is usually named "distributed coding" Hinton et al. (1986), but also "population coding", "vector coding" and "state space representation", in the formulation by Churchland (1989). In the current interpretation of population coding, a higher level map may code for a kind of object or fact, and is the concurrent level of firing of a population of cells in that map that represents a specific category of the kind. There is large evidence of population coding in higher cortical areas (Rolls & Tovee, 1995; Abbott et al., 1996; Pasupathy & Connor, 2002). In §4.3.1 we will use the population coding concept in analyzing the organization of the higher-level LISSOM maps in the model.

2.4 Theoretical issues in explaining categorization by self-organizing networks

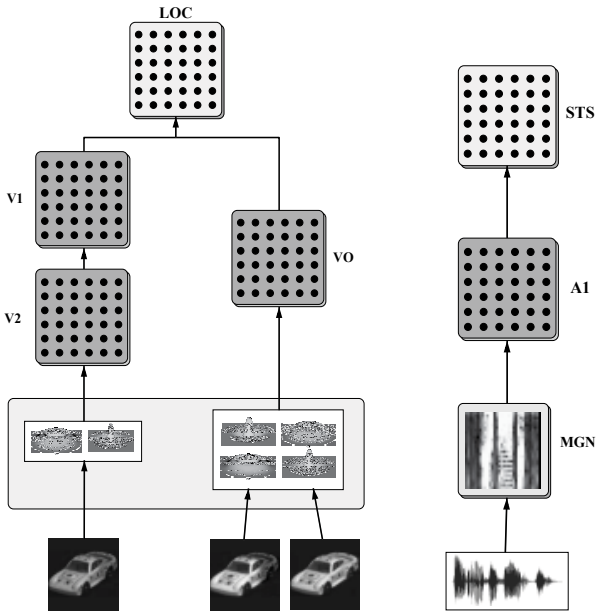
We will now discuss several critical issues in the enterprise of shedding light on human categorization by modeling the neural mechanism of coincidence detection and self-organization.

There is a classical argument against basing the organization of categories on feature associations, in that categories instead could form by distilling a few features from many, the ones typical of the category. Quoting from Gibson & Levin (1975,p.23), "[association] does not work for perceptual learning, because what is learned is not addition of something but rather extraction of something". It is arguable that associating, in the standard theory, would be equivalent to adding something, certainly detecting coincidence and organizing their representation accordingly has nothing to do with the notion of addition, while its net effect is typically of coding the discrimination of some feature.

A deeper challenge is against the whole assumption, here made, that early categorization is perceptually driven.

For Mandler (2004), an important feature of human cognition is conceptual analysis, that is said to be parallel to, and different from, ordinary perceptual learning, which is associative in essence. We do not deny that in building categories humans most probably rely on a variety of skills that cannot be ascribed to our putative mechanism, even in a pre-linguistic phase. We believe, however, that the coincidence mechanism plays a key role, for other reasons. The first reason is that coincidence not necessarily implies only perceptual elements. Regular conjunctions can be detected between signals elicited by stimuli and other signals recalled by activations of representations already coded, this is a necessary step for generalizing categories over a number of instances experienced in time. The second reason comes from developmental science evidence that casts doubt on the separation between perceptual and conceptual categorization (Eimas & Quinn, 1994; Rakison, 2003). As a result, Mandler's hypothesis that conceptual learning is different in nature from purely associative learning, not necessarily requires a different notion than that of coincidence detection and self-organized coding. One possibility is that conceptual analysis of experience is based on the same sort of coincidence detection and coding, but involving phenomena that emerge in widespread cortical circuits, with self-organization at a more global scale.

A more general criticism against the mechanism here proposed could be the objection that self-organization of perceptual regularities alone cannot act as the foundation for mental categories, because of an insufficiency of constraints. Keil et al. (1998) give the suggestive example that "virtually every washing machine ever encountered is white, yet we know that



(a) Visual pathway (b) Auditory pathway

Fig. 2. Scheme of the two pathways that make up the model.
All acronyms are listed in Tab. 1

whiteness is irrelevant in determining that an object is a washing machine.". This argument reveals to be poor when directed against the coincidence mechanism. In fact, it is entirely based on the assumption that perceptual features are equally neutral for the perceiver. Not only is this not the case, but coincidence detection and self-organizational coding is first and foremost the tool that allows the brain to rank the many features occurring with a category of objects or an event, according to how features are predictive of the category. Predictiveness is evaluated each time a next instance of the category is experienced, with consequent adjustment of the ranking of features. There is no doubt that early categories might include salient features that are not distinctive traits for the kind, in its normative semantics. We suspect that for many young children being white is indeed important for an object in order to be a washing machine, as important as being forbidden by their mothers. But this seems only to show that what is used to make a prediction depends on our global knowledge of the world, and that sometimes this knowledge could not be the best on the market.

3. Modeling categorization in the cortex

This section will introduce the model, the rationale for its design, describe each component, and discuss in detail the higher component of the model, where a SOM or a LISSOM module will be used and compared.

3.1 Overall structure of the model

The overall model is a combination of artificial cortical sheets that reproduce essentially the part of the brain involved in the categorization of objects with perceptual salience: the visual and the auditory pathways. Each of the two processing streams is fed by realistic stimuli, and therefore also comprises, in an approximate simulation, the components that transduce the external signals, and the subcortical components. An outline of the modules that make up the visual and auditory pathways is shown in Fig. 2, the names of the component are in Tab. 1, together with their parameters as in equation (5). The two higher cortical maps in the visual and the auditory paths, LOC and STS, will carry the best representation coded by the models on object visual features and word sound features. The way of combining the two representations is varied in the model, with the alternative of including a higher LISSOM-type artificial cortical map, or a standard SOM module. While in the visual and auditory pathways the criteria of adhering realistically to the structure of the human cortex is possible, in higher categorization correlation between functions and cortical areas becomes more fuzzy and uncertain. Categorization processes are by definition abstract representations, where what is coded at a neural level has no obvious correspondence with features in the sensorial inputs. Furthermore, in the mapping between words and objects, the native language acts in a complex way, with its own established category structure, that initially mismatches with the emerging mental categories of the developing mind, making it hard to trace back to the level of neural processes. It is therefore, justifiable to adopt at this level a very abstract neural architecture such as the SOM.

3.2 The visual pathway

The visual path is built upon and extends a previous model of visual object recognition (Plebe & Domenella, 2007). The visual system encompasses the ventral stream only, in the broad partition established by Ungerleider & Mishkin (1982), and commonly accepted in current vision science. They proposed the well known dichotomy of "what" and "where" pathways, in that the ventral pathway is specialized for object recognition whereas the dorsal pathway is specialized for spatial perception. While the "where" has been quite controversial and has been subject to other proposed interpretations (Goodale & Milner, 1992), there has always been a large consensus on the purpose of the "what" pathway. For the purposes of this model, therefore, limiting ourselves to the ventral path only, is a valid approximation. One of the simplifications used in the model is the processing of sensorial input coming from one side only. Stereopsis contributes to the perception of 3-dimensional shapes, and is therefore,

acronym	full name	size	r_A	r_E	r_H	γ_A	γ_E	γ_H	γ_N
LGN	Lateral Geniculated Nucleus	112	2.6	-	-	-	-	-	-
MGN	Medial Geniculated Nucleus	36	2.2	-	-	-	-	-	-
V1	Primary Visual Cortex	96	8.5	1.5	7	1.5	1.0	1.0	0.0
A1	Auditory Primary Cortex	24	6.5	1.5	5.0	1.7	0.9	0.9	0.0
V2	Secondary Visual Cortex	30	7.5	8.5	3.5	50.0	3.2	2.5	0.7
VO	Ventral Occipital	30	24.5	4	8	1.8	1.0	1.0	0.0
LOC	Lateral Occipital Complex	16	6.5	1.5	3.5	0.5	1.1	1.6	0.0
STS	Superior Temporal Sulcus	16	2.5	1.5	5.5	1.8	1.0	1.2	0.0
PFC	Pre-Frontal Cortex	24	2.5	1.5	5.5	1.5	3.2	4.1	0.1
ACM	Abstract Conceptual Map	18	-	-	-	-	-	-	-

Table 1. Legend of acronyms of maps used in the model, and their main parameters.

a piece that is missing in the model. It is, however, much less useful than 2-dimensional contours when there is a single, unmoving object in the foveal view of a scene. This is the case of the samples used in our experiments.
This pathway includes the thalamic module, that is not the object of this study and is therefore hardwired, according to what is known about its functions. The governing equations for the subcortical components are the following:

$$x^{(\odot)} = f\left(\left(\vec{l}_{r_\mathrm{A}} + \vec{m}_{r_\mathrm{A}}\right) \cdot \left(\vec{g}_{r_\mathrm{A}}^{(\sigma_\mathrm{N})} - \vec{g}_{r_\mathrm{A}}^{(\sigma_\mathrm{W})}\right)\right) \tag{9}$$

$$x^{(\circledcirc)} = f\left(\left(\vec{l}_{r_\mathrm{A}} + \vec{m}_{r_\mathrm{A}}\right) \cdot \left(\vec{g}_{r_\mathrm{A}}^{(\sigma_\mathrm{W})} - \vec{g}_{r_\mathrm{A}}^{(\sigma_\mathrm{N})}\right)\right) \tag{10}$$

$$x^{(\mathrm{R^+G^-}\odot)} = f\left(\vec{l}_{r_\mathrm{A}} \cdot \vec{g}_{r_\mathrm{A}}^{(\sigma_\mathrm{N})} - \vec{m}_{r_\mathrm{A}}\vec{g}_{r_\mathrm{A}}^{(\sigma_\mathrm{W})}\right) \tag{11}$$

$$x^{(\mathrm{R^+G^-}\circledcirc)} = f\left(\vec{l}_{r_\mathrm{A}} \cdot \vec{g}_{r_\mathrm{A}}^{(\sigma_\mathrm{W})} - \vec{m}_{r_\mathrm{A}}\vec{g}_{r_\mathrm{A}}^{(\sigma_\mathrm{N})}\right) \tag{12}$$

$$x^{(\mathrm{G^+R^-}\odot)} = f\left(\vec{m}_{r_\mathrm{A}} \cdot \vec{g}_{r_\mathrm{A}}^{(\sigma_\mathrm{N})} - \vec{l}_{r_\mathrm{A}}\vec{g}_{r_\mathrm{A}}^{(\sigma_\mathrm{W})}\right) \tag{13}$$

$$x^{(\mathrm{G^+R^-}\circledcirc)} = f\left(\vec{m}_{r_\mathrm{A}} \cdot \vec{g}_{r_\mathrm{A}}^{(\sigma_\mathrm{W})} - \vec{l}_{r_\mathrm{A}}\vec{g}_{r_\mathrm{A}}^{(\sigma_\mathrm{N})}\right) \tag{14}$$

where x is the activation of a neuron, without indexes for clarity. There are two distinct subcortical pathways, one achromatic, processed by equations (9), (10), and another sensitive to colors, limited here to medium and long wapath velengths. The equations are: (11), (12), (13), (14). The symbol $\odot$ refers to on-center type receptive fields, and symbol $\circledcirc$ to off-center receptive fields. The profile of all visual receptive fields is given by differences of two Gaussians $\vec{g}^{\,(\sigma_\mathrm{N})}$ and $\vec{g}^{\,(\sigma_\mathrm{W})}$, with $\sigma_\mathrm{N} < \sigma_\mathrm{W}$. This is an approximation of the combined contribution of gangliar cells and LGN (Dowling, 1987).
The lower cortical maps included in the model are V1, V2 and VO, the primary, secondary, and ventral-occipital visual areas. Their equations are the following:

$$\begin{aligned} x^{(\mathrm{V1})} = f\Bigg(&\gamma_\mathrm{A}^{(\mathrm{V1})}\left(\vec{a}_{r_\mathrm{A}}^{(\mathrm{V1}\leftarrow\odot)} \cdot \vec{x}_{r_\mathrm{A}}^{(\odot)} + \vec{a}_{r_\mathrm{A}}^{(\mathrm{V1}\leftarrow\circledcirc)} \cdot \vec{x}_{r_\mathrm{A}}^{(\circledcirc)}\right) + \\ &\gamma_\mathrm{E}^{(\mathrm{V1})}\vec{e}_{r_\mathrm{E}}^{(\mathrm{V1})} \cdot \vec{x}_{r_\mathrm{E}}^{(\mathrm{V1})} - \gamma_\mathrm{H}^{(\mathrm{V1})}\vec{h}_{r_\mathrm{H}}^{(\mathrm{V1})} \cdot \vec{x}_{r_\mathrm{H}}^{(\mathrm{V1})}\Bigg) \end{aligned} \tag{15}$$

$$x^{(\mathrm{V2})} = f\left(\gamma_\mathrm{A}^{(\mathrm{V2})}\vec{a}_{r_\mathrm{A}}^{(\mathrm{V2}\leftarrow V1)} \cdot \vec{x}_{r_\mathrm{A}}^{(V1)} + \gamma_\mathrm{E}^{(\mathrm{V2})}\vec{e}_{r_\mathrm{E}}^{(\mathrm{V2})} \cdot \vec{x}_{r_\mathrm{E}}^{(\mathrm{V2})} - \gamma_\mathrm{H}^{(\mathrm{V2})}\vec{h}_{r_\mathrm{H}}^{(\mathrm{V2})} \cdot \vec{x}_{r_\mathrm{H}}^{(\mathrm{V2})}\right) \tag{16}$$

$$\begin{aligned} x^{(\mathrm{VO})} = f\Bigg(&\gamma_\mathrm{A}^{(\mathrm{VO})}\Big(\vec{a}_{r_\mathrm{A}}^{(\mathrm{VO}\leftarrow\mathrm{R^+G^-}\odot)} \cdot \vec{x}_{r_\mathrm{A}}^{(\mathrm{R^+G^-}\odot)} + \vec{a}_{r_\mathrm{A}}^{(\mathrm{VO}\leftarrow\mathrm{R^+G^-}\circledcirc)} \cdot \vec{x}_{r_\mathrm{A}}^{(\mathrm{R^+G^-}\circledcirc)} + \\ &\vec{a}_{r_\mathrm{A}}^{(\mathrm{VO}\leftarrow\mathrm{G^+R^-}\odot)} \cdot \vec{x}_{r_\mathrm{A}}^{(\mathrm{G^+R^-}\odot)} + \vec{a}_{r_\mathrm{A}}^{(\mathrm{VO}\leftarrow\mathrm{G^+R^-}\circledcirc)} \cdot \vec{x}_{r_\mathrm{A}}^{(\mathrm{G^+R^-}\circledcirc)}\Big) + \\ &\gamma_\mathrm{E}^{(\mathrm{VO})}\vec{e}_{r_\mathrm{E}}^{(\mathrm{VO})} \cdot \vec{x}_{r_\mathrm{E}}^{(\mathrm{VO})} - \gamma_\mathrm{H}^{(\mathrm{VO})}\vec{h}_{r_\mathrm{H}}^{(\mathrm{VO})} \cdot \vec{x}_{r_\mathrm{H}}^{(\mathrm{VO})}\Bigg) \end{aligned} \tag{17}$$

In all equations the activation x has to be taken as the activation of a generic i-th neuron of that level, and all receptive fields have to be intended as referring to that neuron, the index i has been omitted for clarity. For the same reason, the indication of the radius r of the circular receptive fields has not been superscripted with the indication of the layer, for example in equation (16) all r_A should be read $r_\mathrm{A}^{(\mathrm{V2})}$, as the radii are different in each map.

The use of still pictures justifies the lack of a V3 area, whose main role is the processing of motion information (Gegenfurtner et al., 1997). In exploiting the modularity of the model, enacted by the correspondence with cortical maps, a simplification has been introduced by way of the separation of the processing of shape and color. Shape is elaborated through V1 and V2, by equations (15) and (16), and the processing of color is entrusted to VO, with equation (17).

There is evidence, in fact, that suggests that in the visual system no segregation of functions such as shape or color processing takes place, and that almost all visual cortical maps cooperate in analyzing form, color, motion and stereo information (Van Essen & DeYoe, 1994; Schiller, 1996). On the other hand, it is clear that visual areas are not equally involved in all aspects of object recognition. It is possible to identify specialization in one main function in certain maps, although the same maps might also marginally extend to other operations, and conversely, other maps may overlap in the elective function of that map as well. This is the case in what has been called the color center area by Zeki (1983a;b), who named it "V4". The precise localization of this area in humans, as well as its name, have been the object of a yet unsettled controversy. Sereno et al. (1995) redefined the borders of a color area in humans in the fusiform gyrus, and called it "V4v" ("v" for ventral), Hadjikhani et al. (1998) argued for an anterior area , which they called "V8". Here we are using the more general name of VO (Ventral Occipital), given by Wandell et al. (2005); Brewer et al. (2005) to the cluster of maps in the fusiform gyrus involved in color processing. V1 is the well-known primary visual cortex. It has been the most studied part of the brain (Hubel & Wiesel, 1962; 1959). One of its main functions is the organization of the map into domains of orientation tuned neurons (Blasdel, 1992; Vanduffel et al., 2002), which are fundamental for early shape analysis. For the reasons discussed above, our model discards the contributions of V1 to color processing (Landisman & Ts'o, 2002) and ocularity (Miller et al., 1989). The main projection from V1 is to its immediately anterior area, V2, and this connection is maintained in the model. V2 is almost as large as V1, but much less understood. A generally shared idea is that V2 is responsible for shape analysis at a level of complexity and scale larger than that of V1 (Kobatake & Tanaka, 1994). This is compatible with findings of V2 cells responding to end-lines and corners (Heider et al., 2000), gratings spirals and other complex shapes (Hegdé & Van Essen, 2000), Glass patterns Smith et al. (2007), and, with shared stronger evidence, angles (Ito & Komatsu, 2004; Anzai et al., 2007).

Shape and color processing paths converge in a higher visual map, with the following equation:

$$x^{(\mathrm{LOC})} = f\Bigg(\gamma_{\mathrm{A}}^{(\mathrm{LOC})}\left(\vec{a}_{r_{\mathrm{A}}}^{(\mathrm{LOC}\leftarrow V2)}\cdot\vec{x}_{r_{\mathrm{A}}}^{(V2)} + \vec{a}_{r_{\mathrm{A}}}^{(\mathrm{LOC}\leftarrow VO)}\cdot\vec{x}_{r_{\mathrm{A}}}^{(VO)}\right) + \gamma_{\mathrm{E}}^{(\mathrm{LOC})}\vec{e}_{r_{\mathrm{E}}}^{(\mathrm{LOC})}\cdot\vec{x}_{r_{\mathrm{E}}}^{(\mathrm{LOC})} - \gamma_{\mathrm{H}}^{(\mathrm{LOC})}\vec{h}_{r_{\mathrm{H}}}^{(\mathrm{LOC})}\cdot\vec{x}_{r_{\mathrm{H}}}^{(\mathrm{LOC})}\Bigg) \tag{18}$$

The LOC model map corresponds to the area in the human cortex thought to be crucial for the task of recognition in vision. Malach et al. (1995) first identified this region, an area located anterior to Brodmann's area 19, near the lateral occipital sulcus, extending into the posterior and mid fusiform gyrus and occipital-temporal sulcus, with an overall surface size similar to V1. They called it "lateral occipital complex", where the term "complex" denotes the uncertainty regarding whether this region is a single visual map or a cluster of several maps. Being an object of research that is still quite recent, a clear picture of LOC characteristics and its role in vision is still lacking, however, the impetus given by its discovery

has lead to an outstanding body of knowledge. Perhaps the most important idea, one that has obtained a certain amount of consensus, is that this area is involved in visual behavior in which recognition is the main task (Grill-Spector et al., 1998; 1999; Vuilleumier et al., 2002; Kourtzi et al., 2003; Weigelt et al., 2007).

3.3 Auditory pathway

A division in dorsal and ventral streams, similar to that in the visual channel, has been found in the auditory system as well, with two streams originating in the caudal and rostral parts of the auditory cortex, targeting spatial and non-spatial domains of the frontal lobe (Romanski & Goldman-Rakic, 1999). Here again, curiously, the "where" interpretation is debated (Belin & Zatorre, 2000), but not the "what" processing path. This is the only contribution taken into account in the auditory system of this model.

As in the ventral pathway, a simplifications used in the model is the processing of sensorial input coming from one side only. This is not too much of a limitation for the auditory system, since biaural hearing supports the "where" processing but not the "what" detection. An outline of the modules that make up this pathway is shown in Fig. 2(b). The auditory path includes as thalamic component the medial geniculate nucleus, with the following equation:

$$x^{(\square)}_{\tau,\omega} = \left| \sum_{t=t_0}^{t_M} v(t) w(t-\tau) e^{-j\omega t} \right|^2 \tag{19}$$

where the symbol $\square$ refers to the spectrotemporal representation of the auditory signal, the horizontal dimension τ is time, and the vertical dimension ω is frequency. Function $w(\cdot)$ in 19 is a short term temporal window, that performs a spectrogram-like response, similar to that given by the combination of cochlear and MGN processes (Brown, 2003).

The cortical modules in this pathway include the auditory primary cortex, and the superior temporal sulcus, the equations are the following:

$$x^{(\mathrm{A1})} = f\left(\gamma_{\mathrm{A}}^{(\mathrm{A1})} \vec{a}_{r_{\mathrm{A}}}^{(\mathrm{A1}\leftarrow\square)} \cdot \vec{x}_{r_{\mathrm{A}}}^{(\square)} + \gamma_{\mathrm{E}}^{(\mathrm{A1})} \vec{e}_{r_{\mathrm{E}}}^{(\mathrm{A1})} \cdot \vec{x}_{r_{\mathrm{E}}}^{(\mathrm{A1})} - \gamma_{\mathrm{H}}^{(\mathrm{A1})} \vec{h}_{r_{\mathrm{H}}}^{(\mathrm{A1})} \cdot \vec{x}_{r_{\mathrm{H}}}^{(\mathrm{A1})} \right) \tag{20}$$

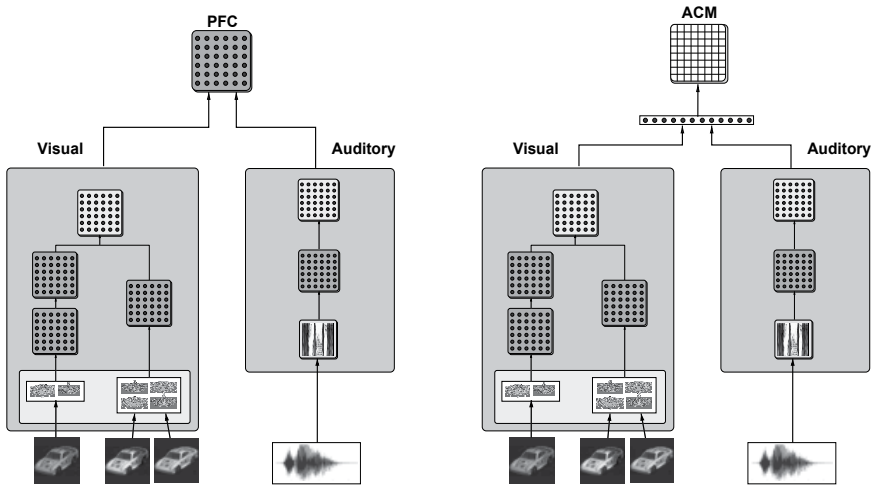
$$x^{(\mathrm{STS})} = f\left(\gamma_{\mathrm{A}}^{(\mathrm{STS})} \vec{a}_{r_{\mathrm{A}}}^{(\mathrm{STS}\leftarrow A1)} \cdot \vec{x}_{r_{\mathrm{A}}}^{(A1)} + \gamma_{\mathrm{E}}^{(\mathrm{STS})} \vec{e}_{r_{\mathrm{E}}}^{(\mathrm{STS})} \cdot \vec{x}_{r_{\mathrm{E}}}^{(\mathrm{STS})} - \gamma_{\mathrm{H}}^{(\mathrm{STS})} \vec{h}_{r_{\mathrm{H}}}^{(\mathrm{STS})} \cdot \vec{x}_{r_{\mathrm{H}}}^{(\mathrm{STS})} \right) \tag{21}$$

Very little is known about the computational organization of the auditory primary cortex, compared to the early visual maps just discussed (Linden & Schreiner, 2006). Our model discards biaural interaction, and preserves the main connectivity from single cochlear signals in the Medial Geniculate nucleus to A1. A large body of evidence points to an organization of A1 with a fundamental dependency on sound frequencies along one cortical dimension, and a distribution of neural responses to temporal properties (Atzori et al., 2001; Miller, Escab, Read & Schreiner, 2002; Winer et al., 2005), model A1 has frequency and time as map dimensions.

Equation (21) states the projection from the primary auditory cortex to STS. This is the model's correlate of a region in the cortical ventral auditory stream, on which there is accumulating evidence and a convergence of opinion on its role in representing and processing phonological information (Belin et al., 2000; 2004; Liebenthal et al., 2005; Hickok & Poeppel, 2007).

3.4 Higher categorial maps

The higher part of the model is implemented in two versions, as shown in Fig. 3. In version (a) the two higher cortical maps in the visual and the auditory paths, LOC and STS, are the



(a) PFC version (b) ACM version

Fig. 3. Scheme of the complete model, in two versions. In version (a) LOC and STS converge to PFC (PreFrontal Cortex), of type LISSOM. Version (b) aggregates LOC and STS contents in a single vector, presented to ACM (Abstract Conceptual Map), of type SOM.

afferents to a cortical map, still modeled with LISSOM, called PFC. In version (b) all neural activations in LOC and STS are aggregated in a single vector, that is fed to an abstract SOM type component, called ACM. Therefore there are the following two different equations for the higher cortical maps in the model:

$$x^{(\mathrm{PFC})} = f\left(\gamma_{\mathrm{A}}^{(\mathrm{PFC})}\left(\vec{a}_{r_{\mathrm{A}}}^{(\mathrm{PFC}\leftarrow LOC)} \cdot \vec{x}_{r_{\mathrm{A}}}^{(LOC)} + \vec{a}_{r_{\mathrm{A}}}^{(\mathrm{PFC}\leftarrow STS)} \cdot \vec{x}_{r_{\mathrm{A}}}^{(STS)}\right) + \gamma_{\mathrm{E}}^{(\mathrm{PFC})}\vec{e}_{r_{\mathrm{E}}}^{(\mathrm{PFC})} \cdot \vec{x}_{r_{\mathrm{E}}}^{(\mathrm{PFC})} - \gamma_{\mathrm{H}}^{(\mathrm{PFC})}\vec{h}_{r_{\mathrm{H}}}^{(\mathrm{PFC})} \cdot \vec{x}_{r_{\mathrm{H}}}^{(\mathrm{PFC})}\right) \tag{22}$$

$$w^{(\mathrm{ACM})} = \arg \min_{i\in\{1,\ldots,M\}} \left\{ \left\| \begin{bmatrix} \vec{x}_{r_\infty}^{(LOC)} \\ \vec{x}_{r_\infty}^{(STS)} \end{bmatrix} - \vec{x}_i^{(\mathrm{ACM})} \right\| \right\} \tag{23}$$

There are actually several areas where visual and auditory signals converge, and more than one area activated in categorization tasks, justifying the solution of an abstract architecture like the SOM as the final map, with equation (22). Its relevance in principle for the brain categorization mechanism has been discussed in §2.2.

The other assumption, made with equation (22), is to expect a specific area to be at least partially involved in aspects of abstraction and integration of object information typical of categorization, even if not the only area where this process takes place. This is a role which can be ascribed to biological PFC, Freedman et al. (2003) have suggested that PFC could also be a crucial component of high-level visual categorization. While ITC (Inferior Temporal Cortex) is more involved in the analysis of viewed shapes, lateral PFC could form a more abstract kind of categorization (Miller, Freedman & Wallis, 2002; Fuster, 2002; Huey et al., 2006). Needless to say, biological PFC deals with larger sets of object information than those used in the model, and these would include for example, the functional information pertaining to what objects can be used for.

The choice of ACM allows a more abstract, and powerful way of categorizing information processed by the visual and auditory pathways, at the price of losing the topological information pertaining to the LOC and STS maps, and population coding organization, that can be studied in the model PFC.

4. Results of the model

In this section we will describe how the model has been trained in this experiment, we will report on the functions developed in all the lower cortical maps of the model, and subsequently discuss the various aspects of categorization that emerged in the higher maps. For this purpose additional methods of analysis are required, that will be described. In the upper map, representative of the joint visual and linguistic aspects of object categorization, PFC and ACM results will be compared.

4.1 Simulation of inner and environmental experiences

The model has been exposed to a variety of stimuli, in different stages of its development, that to some extent parallel periods of human development from the pre-natal stage to that of early language acquisition. Initially only V1, VO and A1 maps are allowed to modify their synaptic weights, by equations (6), (7), and (8). The stimuli presented to V1 and VO are synthetic random blobs that mimic waves of spontaneous retinal activity, that are known to play a fundamental role in the ontogenesis of the visual system (Mastronarde, 1983; Katz & Shatz, 1996; Thompson, 1997; Gödecke & Bonhoeffer, 1996; Chapman et al., 1996). Blobs presented to V1 are elongated along random directions, to stimulate orientation selectivity. Blobs to VO are circular, with constant hues, and random size, position, and intensity. The A1 map is exposed to short trains of waves sweeping linearly around a central frequency. Time durations, central frequencies and sweeping intervals are changed randomly. The next period of development involves V2 and STS maps as well. The visual stimuli comprises pairs of elongated blobs with a coinciding end point, to enhance the experience of patterns that are slightly more complex

bird	boat	bottle
can	car	cheese
chewinggum	chicken	cigarettes
cup	dog	drink
dummy	frog	fruit
hanger	jam	jug
kitten	medicine	mug
pepper	piece	pig
plug	pot	pottery
ring	sauce	soap
spoon	sweets	tape
telephone	toast	tomato
tool	truck	

Fig. 4. The samples used as stimuli for the models, on the bottom all 100 objects of the COIL-100 image library, in their frontal view, on the top the 38 names of categories in which the objects has been classified.

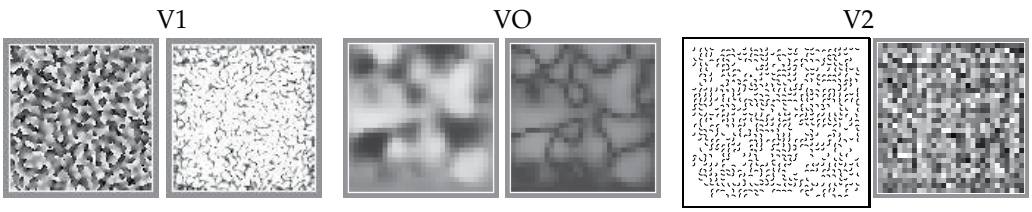


Fig. 5. Organization of neural responses in the lower maps of the model. The two maps on the far left show orientation selectivity in V1. Of these, the map at the far left shows the preferred orientation for each neuron, coded in gray level from black→horizontal to white→vertical. The map on the right contains the degree of neuronal selectivity in a gray scale from black→equally responsive to all orientations, to white→highly selective to a single orientation. The two middle maps represent the hue-responding domains in VO. The map on the far left encodes the preferred hue, from black→red to white→green, the rightmost map is again the degree of selectivity. The pair of maps on the far right show angle selectivity in V2. The map on the far left is overlaid with patterns inducing peak responses in the neurons, the map on the far right is the degree of selectivity.

than lines, such as corners. The auditory stimuli are synthesized waves of the 7200 most common English words (from `http://www.bckelk.uklinux.net/menu.html`), with length of 3 to 10 characters. All words are converted from text to waves using *Festival* software (Black & Taylor, 1997), with cepstral order 64 and a unified time window of 2.3 seconds. In the development stage that corresponds to that just after eye opening, natural images are used. In order to include the primary and most realistic difficulty in recognition, which is the identification of an object under different views, the COIL-100 collection has been used (Nayar & Murase, 1995), where for each of the 100 objects, 72 different views are available. In most experiments, unless otherwise stated, only 8 views per object have been used during the learning phase of the model, and all 72 views are used in the testing phases.
The last stage of the experiment simulates events in which an object is viewed and a word corresponding to its basic category is heard contemporaneously. The 100 objects have been grouped manually into 38 categories. Certain categories, such as `cup` or `medicine` have 5 exemplars in the object collection, while others, such as `telephone`, have only one exemplar. Each category word is converted from text to waves using the `en1` "Roger" male voice, and the `us1` female American speaker in the *Festival* software. Both male and female utterances are duplicated at standard and slower speeds, using the 1.3 value of the `Duration_Stretch` parameter in *Festival*. The 100 objects and the names of the categories can be seen in Fig. 4.

4.2 Emergence of organization in the lower maps

At the end of development, different organizations are found in the lower maps that enable the performance of processes that are essential to recognition, and that are similar to those found in corresponding brain areas. The model's V1 map organized orientation selectivity, with responsiveness of neurons to oriented segments arranged over repeated patterns of gradually changing orientations, broken by few discontinuities, as shown in Fig. 5.
This ordering resembles the one known to be found in biological primary cortex (Blasdel, 1992; Vanduffel et al., 2002). In the VO map of the model, most neurons respond to specific hues, regardless of intensity. This is one of the basic features of color processing. Color constancy is crucial in object recognition and is known to develop somewhere between two and four months of age (Dannemiller, 1989). The organization in the model's VO is shown in Fig. 5.

The kind of mapping found in A1 is typically tonotopic, and it encodes the dimensions of frequency and time sequences in a sound pattern. This is known to be the main ordering of neurons in biological A1 (Verkindt et al., 1995). The main organization found in the V2 map is responsiveness to angles, especially in the 60 and 150 degrees range, this is shown in Fig. 5. This kind of selectivity is one of the major phenomena recently discovered in biological V2 (Ito & Komatsu, 2004; Anzai et al., 2007). We refer to other published works for further details on the functions that emerged in V1 and VO (Plebe & Domenella, 2007) and V2 (Plebe, 2007).

4.3 Categorization and coding analysis

In the scope of this experiment the most interesting results are found in the three higher maps. It is here where we expect types of categorization to take place that concern both visual and word forms, and mostly conceptual categorization at the boundary between the two. In order to understand how a categorial representation about world objects can be carried by a sheet of artificial neurons, it is necessary to apply mathematical tools which will be introduced below.

4.3.1 Population coding

A common method of analysis has been carried out for the upper maps of LISSOM type to reveal possible distributions of neural activation as population coding of categories, as discussed in §2.3. Let us introduce the following function:

$$x_i(s) : S \in \mathcal{S} \to \mathbb{R}; \quad s \in S \in \mathcal{S}, \tag{24}$$

that gives the activation x of a generic neuron i in one of the higher maps, in response to the presentation of the stimulus s to the system. This stimulus is an instance of a class S, belonging to the set of all classes of stimuli $\mathcal{S}$ available in the experiment. For a class $S \in \mathcal{S}$ we can define the two sets:

$$X_{S,i} = \left\{ x_i(s_j) \mid s_j \in S \right\}; \quad \overline{X}_{S,i} = \left\{ x_i(s_j) \mid s_j \in S' \neq S \in \mathcal{S} \right\}. \tag{25}$$

We can then associate to the class S a set of neurons in the map, by ranking it with the following function:

$$r(S,i) = \frac{\mu_{X_{S,i}} - \mu_{\overline{X_{S,i}}}}{\sqrt{\frac{\sigma_{X_{S,i}}}{|X_{S,i}|} - \frac{\sigma_{\overline{X_{S,i}}}}{|\overline{X}_{S,i}|}}}, \tag{26}$$

where μ is the average and σ the standard deviation of the values in the two sets, and $|\cdot|$ is the cardinality of a set. Now the following relation can be established as the population code of a class S:

$$p(S) : \mathcal{S} \to \{ \langle i_1, i_2, \cdots, i_M \rangle \mid r(S,i_1) > r(S,i_2) > \cdots > r(S,i_M) \}, \tag{27}$$

where M is a given constant, typically one order of magnitude smaller than the number of neurons in the map, and α is a constant that is close, but smaller, than one. The population code $p(S)$ computed with (27) can be used to classify a stimulus s in an expected category:

$$c(s) = \arg\max_{S \in \mathcal{S}} \left\{ \sum_{j=1 \cdots M} \alpha^j x_{p(S)_j}(s) \right\}, \tag{28}$$

where $p(S)_j$ denotes the j-th element in the ordered set $p(S)$. It is possible to evaluate how the population code of a map is effective in discriminating a category S by measuring the fraction

of hits in classifying stimuli belonging to that category:

$$a(S) = \frac{|\{s \mid c(s) \in S\}|}{|S|}, \tag{29}$$

where $|\cdot|$ is the cardinality of a set.

4.3.2 SOM coding

We will provide here some of the mathematical details used in our interpretation of the results in the ACM map, that we consider most relevant to lexical categorization. The ACM is a SOM map, and at the end of its development can be analyzed by labeling its neurons with elements from the set of objects and names.

Since the only SOM map is ACM, and its purpose is to organize categories, the labeling function could be directly specified:

$$l^{(\mathrm{ACM})}(i) = \arg\max_{C \in \mathcal{C}} \left\{ \left| \left\{ c = \langle o, u \rangle \in C = \left(\bigcup_{O \in \mathcal{O}_C} O \right) \times U_C \mid i = w^{(\mathrm{ACM})}(c) \right\} \right| \right\}, \tag{30}$$

where $w_i^{(\mathrm{ACM})}(c)$ is computed by (23), when the utterance u is presented to the auditory path, and the object o to the visual path. The 38 object categories introduce a partition in the set of object's views $\mathcal{O}$, such that all sets of views in the partition, $O \in \mathcal{O}_C$ are of objects of that category C, while the set $U_C \subset \mathcal{U}$ collects all utterances naming the category C.

From the labeling functions the possibility of judging the correctness of the identification or the categorization of an object presented to the model immediately follows, by simply checking the label of the winner neuron in the ACM map. It is also possible to estimate the accuracy of the categorization simply by weighing the number of cases where the category or the object has been classified as the prevailing one in each neuron of the SOM, as done by the following equation:

$$a^{(\mathrm{ACM})}(C) = \frac{\left| \left\{ c \mid l^{(\mathrm{ACM})}(c) = w^{(\mathrm{ACM})}(c) \right\} \right|}{|C|}. \tag{31}$$

4.4 Categorization in the top visual and auditory maps

It is possible to begin evaluating the type of representation established in LOC, by using as stimuli view o of an object O, therefore specializing the equation (24) as:

$$x_i^{(\mathrm{LOC})}(o) : O \in \mathcal{O} \to \mathbb{R}; \quad o \in O \in \mathcal{O}, \tag{32}$$

where $x_i^{(\mathrm{LOC})}(o)$ is computed by (18), when the image o is presented to the visual path. In this case a category S of (24) is just an individual object, whose instances can be ascribed to the same entity, or mistaken for a different entity, if a specific point of view does not resemble the typical view of that object. Samples of population coding of individual objects are shown in Fig. 6. As can be seen from the comparison of the population coding map with LOC responses to different views of the same object, most coding neurons exhibit invariant responses. Invariance, the ability to recognize known objects despite large changes in their appearance on the sensory surface, is the most important property to fulfill the requirement for an object-recognition cortical area. It has been identified in human LOC by several studies (Grill-Spector et al., 1999; Vuilleumier et al., 2002; Kourtzi et al., 2003; Weigelt et al., 2007).

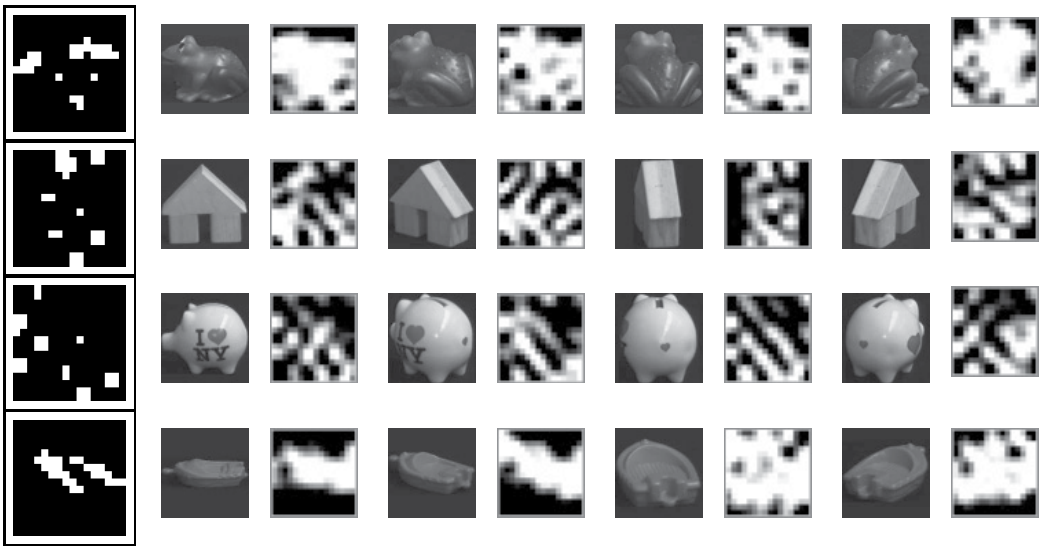


Fig. 6. Examples of population coding of objects in the LOC map. In each row the map on the far left displays the coding neurons. The following images in the row, are samples of views of the same object, and the corresponding response patterns in LOC.

Although the term invariance is commonly used in vision, a better term would be "degree of tolerance" with respect to classes of changes in appearance of the same object, since in no cortical map are responses absolutely invariant to transformations of the same object (Cox et al., 2005). There is also psychophysical evidence of inabilities to recognize objects viewed from less common perspectives, even in adult humans (Farah, 1990). In the model's LOC the supposed coding by population of individual objects is efficient in discriminating most of the views for many objects, although not in an absolute way. By means of equation (29), adapted to stimuli conditions described in (32), the discrimination power of LOC has been quantitatively assessed, and is shown in Tab. 2. In order to test the ability of the representation to cope with invariance, two versions of the model have been trained, exposing them to just a few views for each object only. During the test phase, all 72 views per object have been used. As a comparative figure, discrimination by chance of this task would be of 0.01.

In the STS map it is possible to evaluate the population coding of words, as heard in four different voices, therefore using stimuli that specialize equation (24) as:

$$x_i^{(\mathrm{STS})}(u) : U \in \mathcal{U} \to \mathbb{R}; \quad u \in U \in \mathcal{U}, \tag{33}$$

where $x_i^{(\mathrm{STS})}(u)$ is computed by (18), when the utterance u is presented to the auditory path.

# of views	discriminative power	
	average over all objects	standard deviation
4	0.611	0.334
8	0.638	0.315

Table 2. The model LOC's discriminative power of individual objects by population coding.

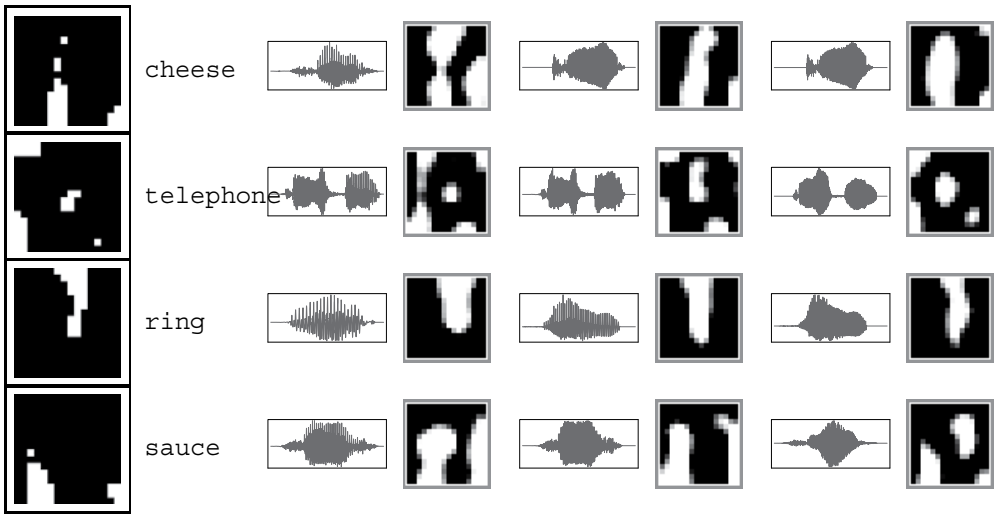


Fig. 7. Examples of population coding of word sounds in the STS map. In each row the leftmost map displays the coding neurons. The following images in the row are samples of sound of the same name, and corresponding response patterns in STS.

There are 38 classes U of sounds in $\mathcal{U}$, corresponding to the names of 38 object categories. Samples of population coding of names are shown in Fig. 7. In Tab. 3 the accuracy of STS

voices	discriminative power	
	average over all names	standard deviation
female	0.882	0.242
male	0.895	0.234
both	0.658	0.300

Table 3. The model STS's discriminative power of words heard by different voices.

in discriminating between the different names is shown. The development of STS has been experimented using three different sets of stimuli: female voices only, male voices only, or using the full set of voices. Listening to voices of a single gender facilitates the identification of names, as can be observed in children. As a comparative figure, discrimination by chance in this task would be 0.026.

4.5 Lexical categorization in PFC or ACM

Lexical categorization in the model takes place where the visual and the linguistic information meet, implemented with the two alternatives of PFC or ACM maps. In the case of PFC population coding is the computed specializing equation (24) as:

$$x_i^{(\mathrm{PFC})}(c) : C \in \mathcal{C} \to \mathbb{R}; \quad c = \langle o, u \rangle \in C = \left(\bigcup_{O \in \mathcal{O}_C} O \right) \times U_C, \tag{34}$$

where $x_i^{(\mathrm{PFC})}(c)$ is computed by (22), when the utterance u is presented to the auditory path, and the object o to the visual path. The 38 object categories introduce a partition in the set

of object's views $\mathcal{O}$, such that all sets of views in the partition, $O \in \mathcal{O}_C$ are of objects of that category C, while the set $U_C \in \mathcal{U}$ collects all utterances naming the category C.
Samples of population coding and actual activations are shown in Fig. 9. Shown are how both the population coding, and the map activations, do not preserve any explicit trace of visual features, and the activations of neurons seem to code the categories in a rather abstract way, as well as the peculiarities of the single samples. For each couple of waves and images, the activations in the PFC map exhibit an important overlap with the population coding of the corresponding category, with contingent additional or lacking activations, denoting the specific view and the specific voice.
The development of both PFC and ACM followed that of all the lower hierarchies in the model, and has been replicated, as in the case of STS, for single gender voices, and for the full set. The objects have been presented using 8 views, of the 72 available, for each. Tab. 4 displays the accuracy achieved at the end of the development, as for STS, the discrimination is quite more accurate when words are spoken by a person of a single gender. Nevertheless, both models achieve a remarkable ability of categorizing objects taking into account their names, even when thrown into a two gender speaking environments. The ACM version exhibits a better accuracy in every condition with respect to PFC, with absolute accuracy on most categories, confirming how the SOM architecture is suitable for simulating categorization.

5. Conclusion

The model here described demonstrates that the perceptual experiences of sound and lights are enough for a first emergence of categories of visual objects and their names, where sounds are uttered by people naming objects, and the lights are the views of those objects. All processes implemented in the model rely on fundamental mechanisms that enable detection of recurrent coincidences in stimuli features, and the self-organization of crucial representations of entities in the external world and of lexical categories.
A strength in the model is the attempt to adhere to the chain of computational processes taking place in the brain, especially in the cerebral cortex, and to rely on fundamental plasticity mechanisms that allow functions to emerge from sensorial experience. The model benefits

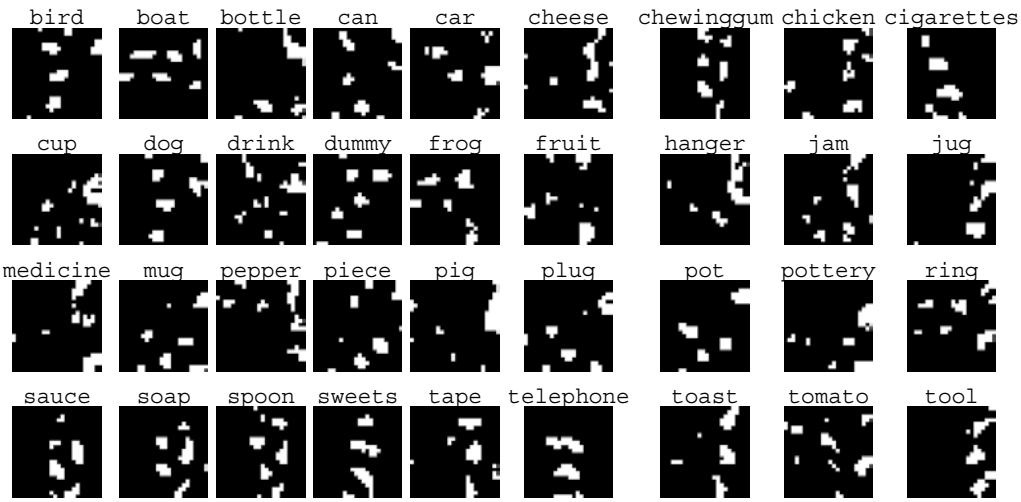


Fig. 8. Population coding in the PFC map for most categories.

category	PFC			ACM		
	all	male	female	all	male	female
bird	0.986	0.903	1.000	1.000	1.000	1.000
boat	0.304	0.198	0.844	0.972	0.965	1.000
bottle	0.450	1.000	0.564	1.000	1.000	1.000
can	0.764	1.000	0.833	0.750	1.000	1.000
car	0.316	0.423	1.000	0.996	1.000	1.000
cheese	1.000	1.000	0.963	1.000	1.000	1.000
chewinggum	0.500	1.000	0.676	1.000	1.000	1.000
chicken	0.750	1.000	0.972	1.000	1.000	1.000
cigarettes	1.000	1.000	1.000	1.000	1.000	1.000
cup	0.293	0.942	1.000	0.994	1.000	1.000
dog	0.986	1.000	1.000	1.000	1.000	1.000
drink	0.484	1.000	0.462	1.000	1.000	1.000
dummy	0.819	1.000	0.986	1.000	1.000	1.000
frog	0.743	0.958	1.000	1.000	1.000	1.000
fruit	0.650	0.560	1.000	1.000	1.000	1.000
hanger	0.576	1.000	0.764	0.875	1.000	1.000
jam	0.616	0.514	0.764	0.938	1.000	1.000
jug	0.389	1.000	0.486	0.500	1.000	0.500
kitten	0.424	0.722	1.000	1.000	1.000	1.000
medicine	0.969	1.000	1.000	1.000	1.000	1.000
mug	0.823	1.000	0.941	1.000	1.000	1.000
pepper	0.823	1.000	1.000	1.000	1.000	1.000
piece	0.935	0.994	0.997	1.000	1.000	1.000
pig	0.979	1.000	1.000	1.000	1.000	1.000
plug	0.521	0.889	0.611	1.000	1.000	1.000
pot	0.694	0.833	1.000	1.000	1.000	1.000
pottery	0.790	0.733	1.000	1.000	1.000	1.000
ring	0.868	1.000	1.000	1.000	1.000	1.000
sauce	0.984	1.000	1.000	1.000	1.000	1.000
soap	0.979	0.950	0.976	1.000	1.000	1.000
spoon	0.750	1.000	0.944	0.750	1.000	1.000
sweets	0.580	0.771	1.000	0.847	1.000	1.000
tape	0.750	0.361	0.806	1.000	1.000	1.000
telephone	0.750	1.000	1.000	0.500	1.000	1.000
toast	1.000	1.000	0.951	1.000	1.000	1.000
tomato	0.743	1.000	1.000	1.000	1.000	1.000
tool	1.000	1.000	0.986	1.000	1.000	1.000
truck	0.483	0.858	1.000	0.977	1.000	1.000
average	0.690	0.853	0.902	0.950	0.999	0.987

Table 4. Categorization accuracy of the two versions of the top map in the model, trained with different sets of voices.

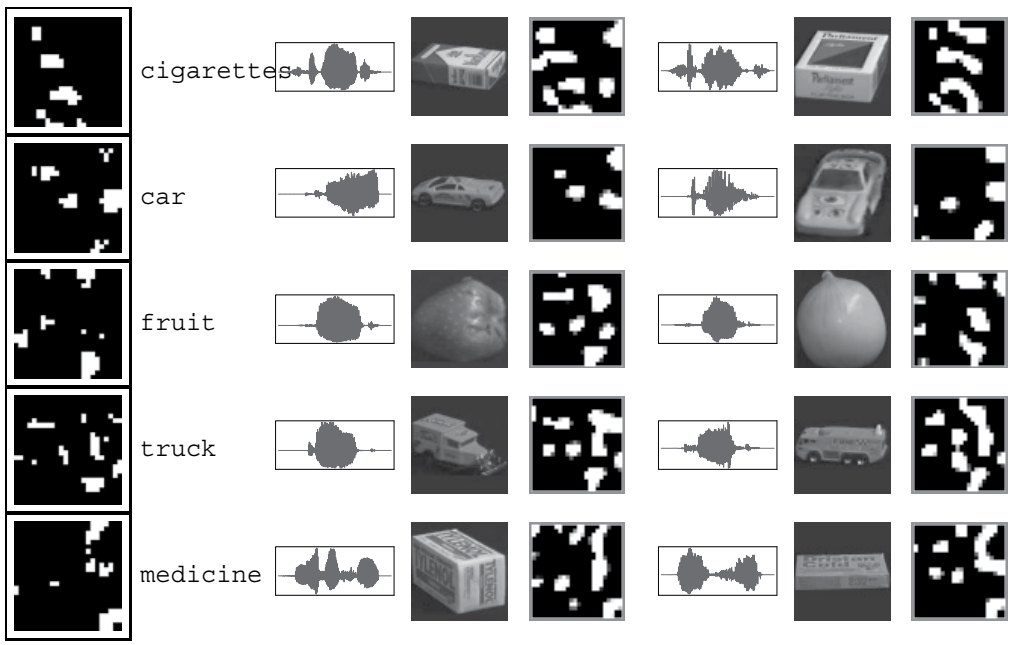


Fig. 9. Examples of population coding in the PFC map. In each row the leftmost map displays the coding neurons. Following images in the row are samples of a sound of the same name, a view of an object of the relevant class, and the corresponding response patterns in PFC. The first sample always has a male voice sound, and the second sample a female voice.

from this effort to different degrees, due to the different levels of knowledge we have of the involved cortical processes. Much more is known about visual recognition, and this explains why this path is much more detailed in the model. In vision, there is also a better knowledge on how the process is organized in the lower part of the path, but we have less information regarding the higher one. In the higher level of the model, where lexical categorization takes place, two different versions have been implemented and compared, one with a classical SOM map, and another that keeps the same LISSOM architecture closer to the cortical structure, even if in this case there is no strict correspondence between model map and a single brain area. We conceive the model here described as an important step in the effort of developing a brain-informed simulation of phenomena in the early development of fundamental human cognitive abilities, that rely on the same coincidence detection and self-organization mechanism. Its novelty is not just in demonstrating the emergence of concepts from the association of visual information and sounds, that has been already simulated in previous work, but rather in its showing how this could happen in neural structures that are similar to the cortical architecture of the human brain.

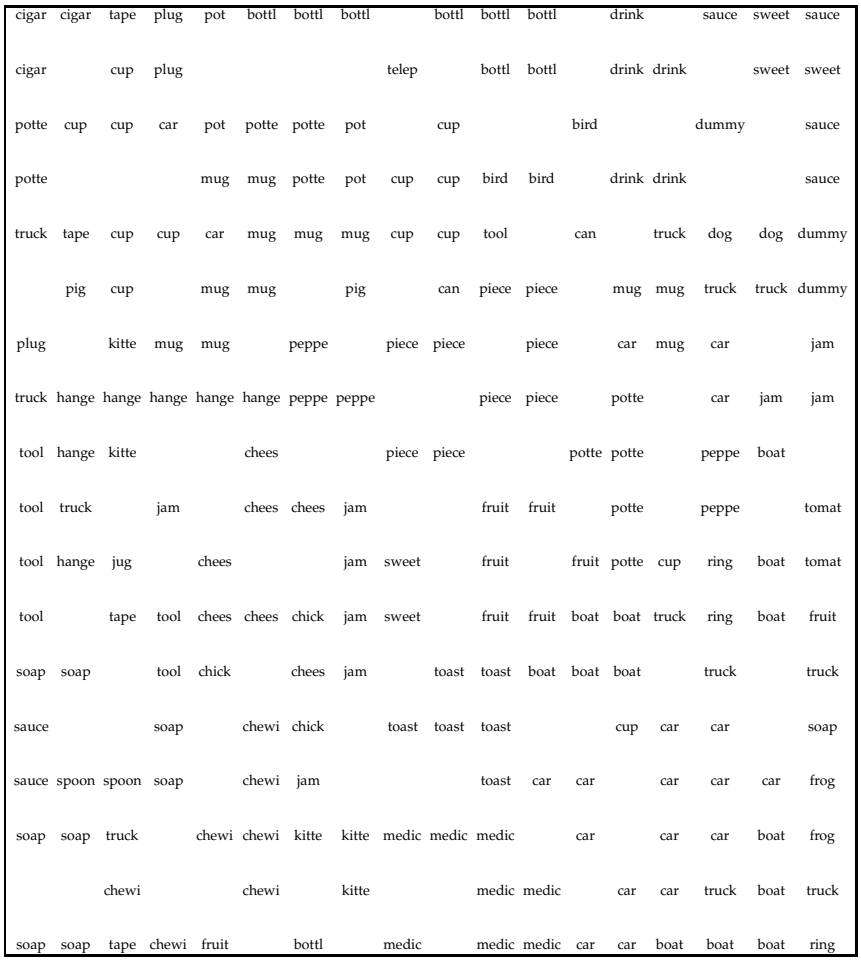


Fig. 10. Organization of categories in the ACM SOM map, in the case of the model trained with all voices. Category nouns are abbreviated for better readability, see Fig. 4 for full nouns.

6. References

Abbott, L. F., Rolls, E. & Tovee, M. J. (1996). Representational capacity of face coding in monkeys, *Cerebral Cortex* 6: 498–505.

Anzai, A., Peng, X. & Essen, D. C. V. (2007). Neurons in monkey visual area V2 encode combinations of orientations, *Nature Neuroscience* 10: 1313–1321.

Atzori, M., Lei, S., Evans, D. I. P., Kanold, P. O., Phillips-Tansey, E., McIntyre, O. & McBain, C. J. (2001). Differential synaptic processing separates stationary from transient inputs to the auditory cortex, *Neural Networks* 4: 1230–1237.

Ball, P. (1999). *The Self-Made Tapestry: Pattern Formation in Nature*, Oxford University Press, Oxford (UK).

Bednar, J. A., De Paula, J. B. & Miikkulainen, R. (2005). Self-organization of color opponent receptive fields and laterally connected orientation maps, *Neurocomputing*

65-66: 69–76.
Bednar, J. A. & Miikkulainen, R. (2000). Tilt aftereffects in a self-organizing model of the primary visual cortex, *Neural Computation* 12: 1721–1740.
Bednar, J. A. & Miikkulainen, R. (2003). Learning innate face preferences, *Neural Computation* 15: 1525–1557.
Bednar, J. A. & Miikkulainen, R. (2006). Joint maps for orientation, eye, and direction preference in a self-organizing model of v1, *Neurocomputing* 69: 1272–1276.
Belin, P., Fecteau, S. & Bédard, C. (2004). Thinking the voice: neural correlates of voice perception, *Trends in Cognitive Sciences* 8: 129–135.
Belin, P. & Zatorre, R. J. (2000). 'what', 'where' and 'how' in auditory cortex, *Nature Neuroscience* 3: 965–966.
Belin, P., Zatorre, R. J., Lafaille, P., Ahad, P. & Pike, B. (2000). Voice-selective areas in human auditory cortex, *Cognitive Brain Research* 403: 309–312.
Black, A. W. & Taylor, P. A. (1997). The festival speech synthesis system: System documentation, *Technical Report HCRC/TR-83*, Human Communciation Research Centre, University of Edinburgh, Edinburgh, UK.
Blasdel, G. G. (1992). Orientation selectivity, preference, and continuity in monkey striate cortex, *Journal of Neuroscience* 12: 3139–3161.
Brewer, A. A., Liu, J., Wade, A. R. & Wandell, B. A. (2005). Visual field maps and stimulus selectivity in human ventral occipital cortex, *Nature Neuroscience* 8: 1102–1109.
Brown, M. C. (2003). Audition, *in* L. R. Squire, F. Bloom, S. McConnell, J. Roberts, N. Spitzer & M. Zigmond (eds), *Fundamental Neuroscience*, Academic Press, New York, pp. 699–726.
Chapman, B., Stryker, M. P. & Bonhoeffer, T. (1996). Development of orientation preference maps in ferret primary visual cortex, *Journal of Neuroscience* 16: 6443–6453.
Chklovskii, D. B. & Koulakov, A. A. (2004). Maps in the brain: what can we learn from them?, *Annual Review of Neuroscience* 27: 369–392.
Choe, Y. & Miikkulainen, R. (2004). Contour integration and segmentation with self-organized lateral connections, *Biological Cybernetics* 90: 75–88.
Churchland, P. M. (1989). *A Neurocomputational perspective: The Nature of Mind and the Structure of Science*, MIT Press, Cambridge (MA).
Cox, D. D., Meier, P., Oertelt, N. & DiCarlo, J. J. (2005). 'breaking' position-invariant object recognition, *Nature Neuroscience* 8: 1145–1147.
Dannemiller, J. L. (1989). A test of color constancy in 9- and 20-weeks-old human infants following simulated illuminant changes, *Developmental Psychology* 25: 171–184.
Dayan, P. & Abbott, L. F. (2001). *Theoretical Neuroscience*, MIT Press, Cambridge (MA).
Dowling, J. E. (1987). *The Retina: An Approachable Part of the Brain*, Cambridge University Press, Cambridge (UK).
Eimas, P. D. & Quinn, P. C. (1994). Studies on the formation of perceptually based basic-level categories in young infants, *Child Development* 3: 903–917.
Farah, M. J. (1990). *Visual Agnosia: Disorders of Object Recognition and What They Tell Us About Normal Vision*, MIT Press, Cambridge (MA).
Fodor, J. (1981). *Representations: Philosofical Essay on the Foundation of Cognitive Science*, MIT Press, Cambridge (MA).
Fodor, J. (1983). *Modularity of Mind: and Essay on Faculty Psychology*, MIT Press, Cambridge (MA).
Freedman, D. J., Riesenhuber, M., Poggio, T. & Miller, E. K. (2003). Visual categorization

and the primate prefrontal cortex: Neurophysiology and behavior, *Journal of Neurophysiology* 88: 929–941.

Fuster, J. M. (2002). Frontal lobe and cognitive development, *Journal of Neurocytology* 31: 373–385.

Gegenfurtner, K. R., Kiper, D. C. & Levitt, J. B. (1997). Functional properties of neurons in macaque area V3, *Journal of Neurophysiology* 77: 1906–1923.

Gibson, E. J. & Levin, H. (1975). *The psychology of reading*, MIT Press, Cambridge (MA).

Gödecke, I. & Bonhoeffer, T. (1996). Development of identical orientation maps for two eyes without common visual experience, *Nature* 379: 251–254.

Goodale, M. A. & Milner, A. D. (1992). Separate visual pathways for perception and action, *Trends in Neuroscience* 15: 20–25.

Grill-Spector, K., Kushnir, T., Edelman, S., Avidan-Carmel, G., Itzchak, Y. & Malach, R. (1999). Differential processing of objects under various viewing conditions in the human lateral occipital complex, *Neuron* 24: 187–203.

Grill-Spector, K., Kushnir, T., Edelman, S., Itzchak, Y. & Malach, R. (1998). Cue-invariant activation in object-related areas in the human occipital lobe, *Neuron* 21: 191–202.

Hadjikhani, N., Liu, A. K., Dale, A. M., Cavanagh, P. & Tootell, R. B. H. (1998). Retinotopy and color sensitivity in human visual cortical area v8, *Nature Neuroscience* 1: 235–241.

Haken, H. (1978). *Synergetics – An Introduction, Nonequilibrium Phase Transitions and Self-organization in Physics, Chemistry and Biology*, Springer-Verlag, Berlin.

Hegdé, J. & Van Essen, D. C. (2000). Selectivity for complex shapes in primate visual area V2, *Journal of Neuroscience* 20: 4117–4130.

Heider, B., Meskenaite, V. & Peterhans, E. (2000). Anatomy and physiology of a neural mechanism defining depth order and contrast polarity at illusory contours, *The European Journal of Neuroscience* 12: 4117–4130.

Hickok, G. & Poeppel, D. (2007). The cortical organization of speech processing, *Nature Reviews Neuroscience* 8: 393–402.

Hinton, G. E., McClelland, J. L. & Rumelhart, D. E. (1986). Distributed representations, *in* D. E. Rumelhart & J. L. McClelland (eds), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1, MIT Press, Cambridge (MA), pp. 77–109.

Hubel, D. H. & Wiesel, T. N. (1959). *Brain and Visual Perception: The Story of a 25-Year Collaboration*, Oxford University Press, Oxford (UK).

Hubel, D. & Wiesel, T. (1962). Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex, *Journal of Physiology* 160: 106–154.

Huey, E. D., Krueger, F. & Grafman, J. (2006). Representations in the human prefrontal cortex, *Current Directions in Psychological Science* 15: 167–171.

Hume, D. (1739). *A Treatise of Human Nature*, John Noon, London. Vol 1,2.

Ito, M. & Komatsu, H. (2004). Representation of angles embedded within contour stimuli in area V2 of macaque monkeys, *Journal of Neuroscience* 24: 3313–3324.

Katz, L. & Shatz, C. (1996). Synaptic activity and the construction of cortical circuits, *Science* 274: 1133–1138.

Keil, F. C., Smith, W. C., Simons, D. J. & Levin, D. T. (1998). Two dogmas of conceptual empiricism: implications for hybrid models of the structure of knowledge, *Cognition* 65: 103–135.

Kobatake, E. & Tanaka, K. (1994). Neuronal selectivities to complex object features in the ventral visual pathway of the macaque cerebral cortex, *Journal of Neurophysiology* 71: 856–867.

Kohonen, T. (1982). Self-organizing formation of topologically correct feature maps, *Biological Cybernetics* 43: 59–69.

Kohonen, T. (1995). *Self-Organizing Maps*, Springer-Verlag, Berlin.

Kohonen, T. (1998). The self-organizing map, a possible model of brain maps, *in* K. H. Pribram (ed.), *Brain and Values: Is A Biological Science of Values Possible?*, Psychology Press, East Sussex (UK), pp. 207–236.

Kohonen, T. & Hari, R. (2000). Where the abstract feature maps of the brain might come from, *Trends in Neurosciences* 22: 135–139.

Kourtzi, Z., Erb, M., Grodd, W. & Bülthoff, H. H. (2003). Representation of the perceived 3-d object shape in the human lateral occipital complex, *Cerebral Cortex* 13: 911–920.

Landisman, C. E. & Ts'o, D. Y. (2002). Color processing in macaque striate cortex: Relationships to ocular dominance, cytochrome oxidase, and orientation, *Journal of Neurophysiology* 87: 3126–3137.

Li, P., Farkas, I. & MacWhinney, B. (2004). Early lexical development in a self-organizing neural network, *Neural Networks* 17: 1345–1362.

Liebenthal, E., Binder, J. R., Spitzer, S. M., Possing, E. T. & Medler, D. A. (2005). Neural substrates of phonemic perception, *Cerebral Cortex* 15: 1621–1631.

Linden, J. F. & Schreiner, C. E. (2006). Columnar transformations in auditory cortex? a comparison to visual and somatosensory cortices, *Cerebral Cortex* 13: 83–89.

Locke, J. (1690). *An essay concerning human understanding*, Meridian Books, Cleveland.

Malach, R., Reppas, J. B., Benson, R. R., Kwong, K. K., Jiang, H., Kennedy, W. A., Ledden, P. J., Brady, T. J., Rosen, B. R. & Tootell, R. B. (1995). Object-related activity revealed by functional magnetic resonance imaging in human occipital cortex, *Proceedings of the Natural Academy of Science USA* 92: 8135–8139.

Mandler, J. M. (2004). *The Foundations of Mind*, Oxford University Press, Oxford (UK).

Mastronarde, D. N. (1983). Correlated firing of retinal ganglion cells: I. spontaneously active inputs in X- and Y-cells, *Journal of Neuroscience* 14: 409–441.

Mayor, J. & Plunkett, K. (2010). A neurocomputational account of taxonomic responding and fast mapping in early word learning, *Psychological Review* 117: 1–31.

Miikkulainen, R. (1997). Dyslexic and category-specific aphasic impairments in a self-organizing feature map model of the lexicon, *Brain and Language* 59: 334–366.

Miikkulainen, R., Bednar, J. A., Choe, Y. & Sirosh, J. (1997). Self-organization, plasticity, and low-level visual phenomena in a laterally connected map model of the primary visual cortex, *in* R. L. Goldstone, P. G. Schyns & D. L. Medin (eds), *Psychology of Learning and Motivation*, Vol. 36, Academic Press, New York, pp. 257–308.

Miikkulainen, R., Bednar, J., Choe, Y. & Sirosh, J. (2005). *Computational maps in the visual cortex*, Springer-Science, New York.

Miller, E. K., Freedman, D. J. & Wallis, J. D. (2002). The prefrontal cortex: Categories, concepts and cognition, *Philosophical Transactions: Biological Sciences* 357: 1123–1136.

Miller, K. D., Keller, J. B. & Stryker, M. P. (1989). Ocular dominance column development: Analysis and simulation, *Science* 245: 605–615.

Miller, L. M., Escab, M. A., Read, H. L. & Schreiner, C. E. (2002). Spectrotemporal receptive fields in the lemniscal auditory thalamus and cortex, *Journal of Neurophysiology* 87: 516–527.

Mountcastle, V. (1957). Modality and topographic properties of single neurons in cats somatic sensory cortex, *Journal of Neurophysiology* 20: 408–434.

Nayar, S. & Murase, H. (1995). Visual learning and recognition of 3-d object by appearence,

International Journal of Computer Vision 14: 5–24.

Pasupathy, A. & Connor, C. E. (2002). Population coding of shape in area v4, *Nature Neuroscience* 5: 1332–1338.

Plebe, A. (2006). Learning visual invariance, *in* M. Verleysen (ed.), *ESANN 2006 – 14th European Symposium on Artificial Neural Networks*, d-side Publications, Evere (BE), pp. 71–76.

Plebe, A. (2007). A model of angle selectivity development in visual area V2, *Neurocomputing* 70: 2060–2066.

Plebe, A., De la Cruz, V. & Mazzone, M. (2007a). Artificial learners of objects and names, *in* Y. Demiris, B. Scassellati & D. Mareschal (eds), *Proceedings of the 6th International Conference on Development and Learning*, IEEE, pp. 300–305.

Plebe, A., De la Cruz, V. & Mazzone, M. (2007b). Simulating the acquisition of object names, *in* P. Buttery, A. Villavicencio & A. Korhonen (eds), *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, Association for Computational Linguistics, Stroudsburg (PA), pp. 57–64.

Plebe, A. & Domenella, R. G. (2006). Early development of visual recognition, *BioSystems* 86: 63–74.

Plebe, A. & Domenella, R. G. (2007). Object recognition by artificial cortical maps, *Neural Networks* 20: 763–780.

Quartz, S. R. & Sejnowski, T. J. (1997). The neural basis of cognitive development: a constructivist manifesto, *Behavioral and Brain Science* 20: 537–596.

Rakison, D. H. (ed.) (2003). *Early Category And Concept Development*, Oxford University Press, Oxford (UK).

Ritter, H. & Kohonen, T. (1989). Self-organizing semantic maps, *Biological Cybernetics* 61: 241–254.

Rolls, E. (1992). Neurophysiological mechanisms underlying face processing within and beyond the temporal cortical visual areas, *Philosophical transactions of the Royal Society B* 335: 11–21.

Rolls, E. & Tovee, M. J. (1995). Sparseness of the neuronal representation of stimuli in the primate temporal visual cortex, *Journal of Neurophysiology* 73: 713–726.

Romanski, L. M. & Goldman-Rakic, P. S. (1999). Dual streams of auditory afferents target multiple domains in the primate prefrontal cortex, *Nature Neuroscience* 2: 1131–1136.

Schiller, P. H. (1996). On the specificity of neurons and visual areas, *Behavioural Brain Research* 76: 21–35.

Sereno, M., Dale, A., Reppas, J., Kwong, K., Belliveau, J., Brady, T., Rosen, B. & Tootell, R. (1995). Borders of multiple visual areas in human revealed by functional magnetic resonance imaging, *Science* 268: 889–893.

Sirosh, J. & Miikkulainen, R. (1997). Topographic receptive fields and patterned lateral interaction in a self-organizing model of the primary visual cortex, *Neural Computation* 9: 577–594.

Smith, M. A., Kohn, A. & Movshon, J. A. (2007). Glass pattern responses in macaque V2 neurons, *Journal of Neurophysiology* 97: 4284–4295.

Thompson, I. (1997). Cortical development: A role for spontaneous activity?, *Current Biology* 7: 324–326.

Turrigiano, G. G. & Nelson, S. B. (2004). Homeostatic plasticity in the developing nervous system, *Nature Reviews Neuroscience* 391: 892–896.

Ungerleider, L. & Mishkin, M. (1982). Two cortical visual systems, *in* D. J. Ingle, M. A. Goodale

& R. J. W. Mansfield (eds), *Analysis of visual behavior*, MIT Press, Cambridge (MA), pp. 549–586.

Van Essen, D. C. & DeYoe, E. A. (1994). Concurrent processing in the primate visual cortex, *in* M. S. Gazzaniga (ed.), *The Cognitive Neurosciences*, MIT Press, Cambridge (MA).

Vanduffel, W., Tootell, R. B., Schoups, A. A. & Orban, G. A. (2002). The organization of orientation selectivity throughout the macaque visual cortex, *Cerebral Cortex* 12: 647–662.

Verkindt, C., Bertrand, O., Echallier, F. & Pernier, J. (1995). Tonotopic organization of the human auditory cortex: N100 topography and multiple dipole model analysis, *Electroencephalography and Clinical Neurophisiology* 96: 143–156.

von der Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striate cortex, *Kibernetic* 14: 85–100.

Vuilleumier, P., Henson, R. N., Driver, J. & Dolan, R. J. (2002). Multiple levels of visual object constancy revealed by event-related fmri of repetition priming, *Nature Neuroscience* 5: 491–499.

Wallis, G. & Rolls, E. (1997). Invariant face and object recognition in the visual system, *Progress in Neurobiology* 51: 167–194.

Wandell, B. A., Brewer, A. A. & Dougher, R. F. (2005). Visual field map clusters in human cortex, *Philosophical transactions of the Royal Society of London* 360: 693–707.

Weigelt, S., Kourtzi, Z., Kohler, A., Singer, W. & Muckli, L. (2007). The cortical representation of objects rotating in depth, *Journal of Neuroscience* 27: 3864–3874.

Westermann, G., Mareschal, D., Johnson, M. H., Sirois, S., Spratling, M. S. & Thomas, M. S. C. (2007). Neuroconstructivism, *Developmental Science* 10: 75–83.

Winer, J. A., Miller, L. M., Lee, C. C. & Schreiner, C. E. (2005). Auditory thalamocortical transformation: structure and function, *Neuron* 28: 255–263.

Zeki, S. (1983a). Colour coding in the cerebral cortex: The reaction of cells in monkey visual cortex to wavelenghts and colours, *Neuroscience* 9: 741–765.

Zeki, S. (1983b). Colour coding in the cerebral cortex: the responses of wavelength-selective and colour-coded cells in monkey visual cortex to changes in wavelength composition, *Neuroscience* 9: 767–781.

34

Applying SOFM and Its FPGA Implementation on Event Processing of PET Block Detector

Dongming Hu
Siemens Medical Solutions, Molecular Imaging
USA

1. Introduction

This chapter introduces an application case of self-organizing feature map (SOFM) on event processing of Positron Emission Tomagraph (PET) detector. The Siemens Inveon Dedicated PET scanner uses an array of 64 20x20 segmented LSO detectors in a tomographic arrangement to allow PET imaging of small animals. A detector consists of 20x20 LSO crystal pixels coupled to a Position Sensitive Photomultiplier Tube (PSPMT) through tapered light guide (Fig. 1 (a)). The 511 kev gamma ray enters LSO pixel, deposits its energy and emits visible light through scintillation process. The light is guided through light guide and detected by PSPMT. The position of scintillation (X, Y) is calculated by Anger logic. The flood image (position profile) of an Inveon PET detector is shown in Fig. 1 (b). The white clouds indicate the accumulation of scintillation events.
The matching and calibration of the system electronics to the detector arrays involves three stages. The first stage identifies the individual crystal elements (pixel) from the raw X and Y values. That is to build a mapping relation between the X, Y value to the crystal index. The current implementation of crystal identification uses a hill-climbing algorithm to find the exact position of every crystal by locating the peaks in position profile. Crystal Lookup Tables (CLTs) are generated as the results of the crystal identification process. Although this method achieves around 95% accuracy, it involves intense human interactions for crystal look-up table corrections. The complex algorithm can only be implemented on a PC, which makes the crystal identification an off-line procedure.

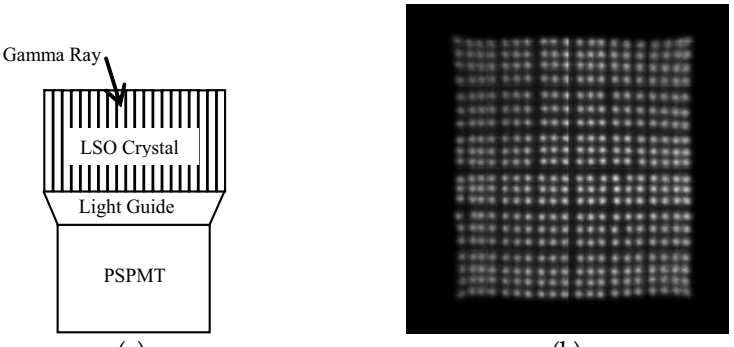


Fig. 1. Inveon PET detector and calibration. (a) Inveon PET detector. (b) Position profile of flood image

This chapter introduces the study on a neural network based algorithm for crystal identification. A modified unsupervised SOFM is trained by the incoming scintillation events to construct a CLT. The algorithm is implemented in a Field Programmable Gate Array (FPGA) chip in the Siemens Inveon Event Processing Module (EPM) electronics, which significantly reduces the training time and brings feasibility to detector real-time training. The preliminary training result shows that SOFM can be used effectively in CLT construction with excellent accuracy. The simulations show that SOFM has a good performance on real-time crystal peak tracking, which brings a good solution to eliminate crystal identification errors caused by gain drifting of PMT or Avalanche Photo Diode (APD).

2. Neural network structure (Hagan et al., 1996)

A block position profile from which a CLT is built contains millions of events, randomly detected by all pixels in a detector block. A supervised neural network is not practical in this case. However, a self-organizing (unsupervised) network can solve the problem effectively. Kohonen's self-organizing feature map is a competitive network that contains a weight vector matrix W (neurons) and a competitive layer C (Fig. 2) .

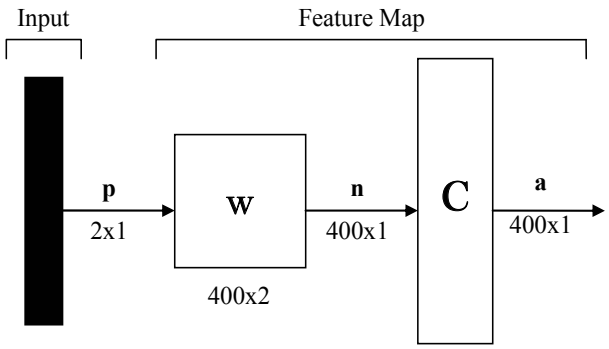


Fig. 2. Structure of self-organizing feature map

The SOFM first determines the winning neuron from n using the competitive layer.

$$\mathbf{a} = \mathbf{compet}\ (\mathbf{n}) \tag{1}$$

$$\mathbf{n} = \mathbf{Wp}$$

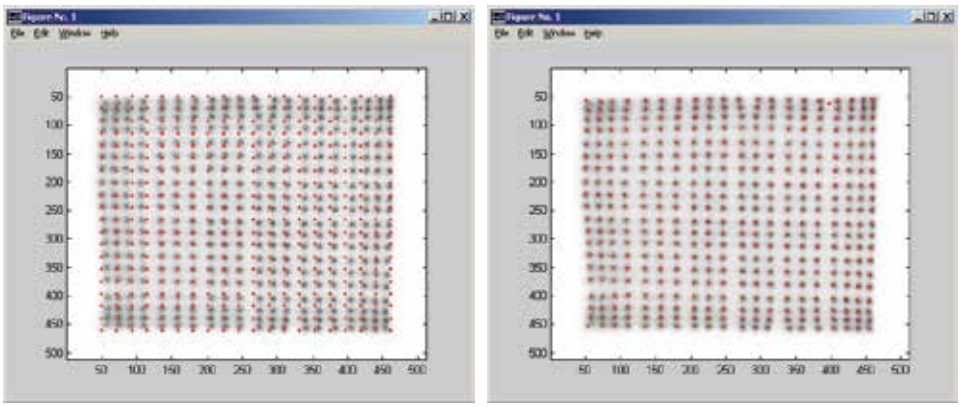
where **p** is input vector with dimension of 2x1 representing the (X, Y) position of an event. **W** is weight matrix with dimension of 400x2 representing neurons positions. Each neuron stands for a detector pixel. So for an Inveon 20x20 detector, the number of neurons is 400. In the normalized case, **Wp** can be regarded as the distances between **p** and neurons weight vectors **W**. Vector **a** is the output from the competitive layer indicating the winning neuron. Only the neuron with the closest weight vector to **p** wins the competition each time.
Next, the weight vectors for all neurons within a certain neighborhood of the winning neuron are updated using the Kohonen rule,

$$\mathbf{w}(q) = (1 - \alpha)\mathbf{w}(q\text{-}1) + \alpha\, \mathbf{p}(q) \tag{2}$$

where α is learning rate, q is training index number, **w** is the neuron in the neighborhood of the winning neuron, which has the same dimension of **p** . Here, neighborhood 1 is used, so only the winning neuron is updated.
When a vector **p** is presented, the position of the winning neuron will move a small distance toward **p**. The result is that, after many presentations of **p** from incoming events, the neuron will have learned the vectors that cluster together, which indicates a pixel location in position profile. After training, the positions of the neurons in the weight matrix represent the position of the detector pixels. A CLT can then be easily built from the neurons.

3. Simulation on SOFM training (Hu et al., 2006)

The SOFM training is first simulated using a Matlab program to verify the algorithm feasibility. The 400 neurons are evenly distributed in the interested area as initial weight vectors. Fig. 3 (a) shows a detector position profile and the neurons' initial positions (round dots). The initial weight vectors can be randomly arranged. However, since we know that the detector pixels are assembled in a grid pattern, applying this constraint in the arrangement of initial weight vectors will speed up the self-organizing process and make it reliable.
The position profile is histogrammed from a list mode file that records the sequence of all events, randomly detected by all pixels of the detector. The list mode file is acquired with a wide open energy window. Reading the (X, Y) position of an event sequentially from the list mode file as an input vector ensures that a training vector is randomly picked up from the position profile. In this simulation, three million events are used as training vectors, which indicates the weight matrix is updated 3 million times. In the Matlab program, the training algorithm is modified and the learning rate is varied over time to achieve better training solutions.
The training result is shown in Fig. 3 (b). In this simulation, most of the neurons find the right pixel location. Only two neurons are misplaced among 400 neurons, which yield 99.5% location accuracy. The total training time for 3 million events is around 8 hours on a Pentium IV 1.6GHz PC. All the neurons are involved in the competition, so training time increases with the number of neurons.



(a) (b)

Fig. 3. Matlab simulation on SOFM training. (a) Position profile and initial positions of neurons in simulation. (b)Matlab training results

4. FPGA implementation

To make the SOFM a practical method for crystal identification, the issue of long training time must be addressed. Since the SOFM algorithm is pretty compact and simple, it can be implemented in the FPGA chip in the Inveon EPM electronics to accelerate the training process. The schematic of the FPGA implementation is shown in Fig. 4. It includes a large scale FPGA chip and several Static Random Access Memory (SRAM) chips. The FPGA implementation consists of a front end module, a SRAM addressing module, a competition pipeline and a neuron training module. A position profile is acquired first with a short period of time. Initial weight vectors of the 400 neurons based on the position profile are calculated and downloaded to the SRAM before training. The initial neurons are evenly distributed in the interested area of the position profile as discussed before. Since multiplication is involved in competition and learning calculations, and integer does not provide enough resolution, a fixed point data format with 9 integer bits and 7 fractional bits is used. So the (X, Y) position of each neuron occupies 4 bytes, with range of 0 - 512 and resolution of 2^{-7}.

A lookup table is also generated based on the initial weight vectors of the neurons, and stored in the SRAM. The lookup table defines regions indexed by the (X, Y) value of the event. Only the neurons in the same region rather than all 400 neurons compete to each other. The adjacent regions overlap each other to ensure that all necessary neurons are picked up for competition when an event comes in. The lookup table reduces the size of the competitive layer and the pipeline processing time. So the required FPGA resource and processing cycle does not increase with the number of neurons.

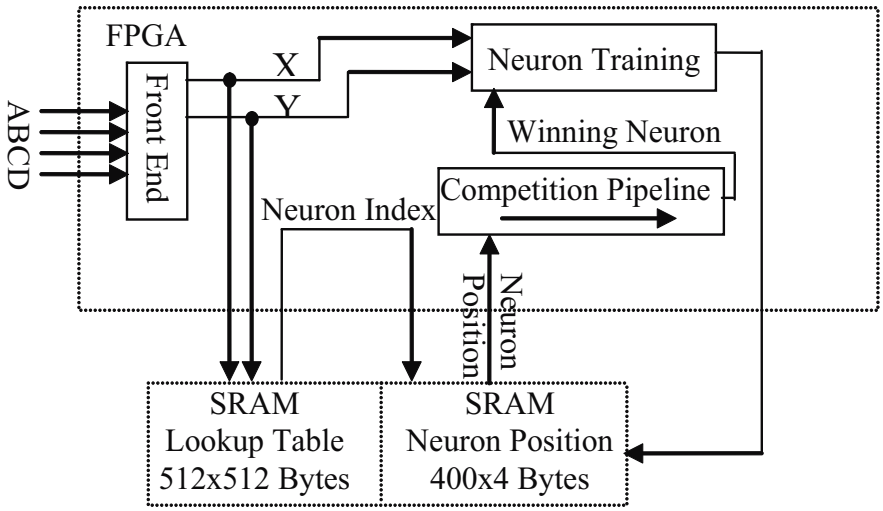


Fig. 4. SOFM FPGA implementation schematic

When an event triggers, the digitized ABCD values are sampled in the front end, and the (X, Y) position of the event is calculated by Anger logic. The (X, Y) value is used as an address to retrieve the region number from the lookup table in the SRAM. The positions of all the neurons in the region are read from the SRAM and sent to the competition pipeline sequentially. The FPGA contains multiple 18 bit x 18 bit multiplier blocks. Due to the pipeline implementation, only one multiplier is needed. The winner of the competition is the

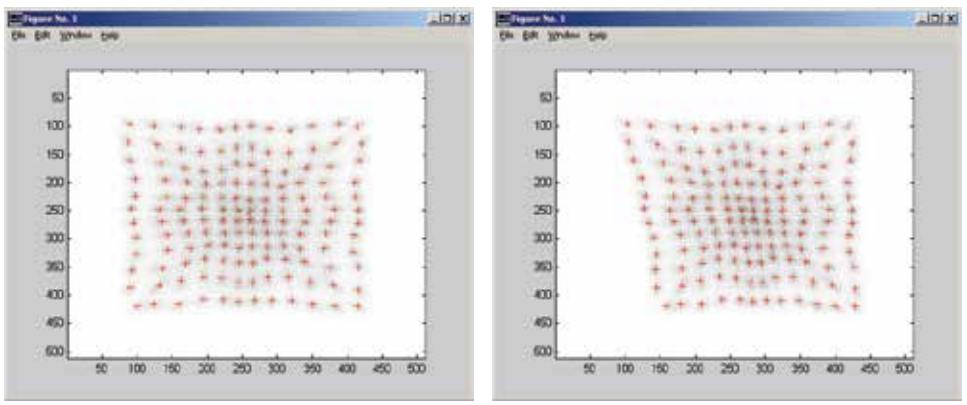
neuron to be updated by the (X, Y) value of the incoming event using the Kohonen rule. Each time an event happens, only one neuron is updated. The updated neuron is then stored back to the SRAM. And the training logic is ready for the next event. The neurons' positions are uploaded from the SRAM after the training is completed. A CLT is built based on the peak positions represented by the neurons' positions.

During training, a 511keV weak source is placed in front of the detector. In 3 minutes, the SOFM is trained by more than 4.5 million events. The FPGA training logic is running at a clock with a period of 16ns. A training cycle only consumes 688ns. If a hot source is used, the training can be accomplished in seconds. In this training, all neurons find their corresponding pixel location.

5. Real-time crystal peak tracking (Hu et al., 2007)

Using the FPGA implementation, the neuron positions can be always trained or updated in the acquisition. Once the neurons are trained to the positions of clustered events, which indicate the positions of crystal peaks, they are trapped around these positions in the further training. When the positions of clustered events move slowly due to the drifting of gains, the neurons are capable of tracking the position changes of crystal peaks in real-time.

The feasibility of real-time crystal peak tracking by SOFM is verified through simulations on an APD detector, the SOFM is first trained to get the correct neuron positions. Then the bottom-left corner of the position profile in Fig. 5 (a) is pushed to the right slowly to simulate APD gain drifting. Fig. 5(b) shows that the position profile of the detector is changed because of the gain drifting. If the original crystal lookup table is used for crystal decoding, wrong crystal id is generated and the resolution of the scanner is degraded. In the simulation, the neurons are trained simultaneously while the gain is drifting. Fig. 5(b) shows the drifted position profile, and the responses of the neurons. It can be seen that the neurons are capable of tracking the position changes of the crystal peaks in real-time. A dynamic crystal lookup table can be generated based on the updated neuron positions. It brings a good solution to eliminate crystal decoding errors due to the APD gain drifting caused by temperature variations.



(a) (b)

Fig. 5. Matlab simulation on SOFM training. (a) SOFM training results for an APD detector. (b) SOFM neuron response to APD gain drifting

6. Conclusions

The SOFM based algorithm for CLT building yields higher pixel identification accuracy than the currently used method. The algorithm is compact and simple, and can be easily implemented in the FPGA chip present in the Inveon EPM electronics. The required FPGA resources do not increase with the number of pixels in the detector. The successful FPGA implementation makes this algorithm practical in training time. By applying FPGA implementation, the SOFM is capable of tracking slow position changes of crystal peaks in real-time, which brings a good solution to eliminate crystal identification errors caused by gain drifting, especially the gain changes of APD detectors due to the temperature variations.

7. References

Hagan, M.T.; Demuth, H. B. & Beale M. (1996). *Neural Network Design,* PWS Publishing Company, ISBN 0-5349-4332-2, Boston, USA.

Hu, D.; Atkins, B. & Lenox, M. (2006). A Neural Network Based Algorithm for Building Crystal Lookup Table of PET Block Detector, *2006 IEEE Nuclear Science Symposium Conference Record*, pp. 2458-2461, ISBN 1-4244-0561-0, San Diego CA, USA, Oct. 2006, IEEE, San Diego

Hu, D. & Gremillion, T. (2007). Verification of Neural Network Based Algorithm for Crystal Identification of PET Block Detector, *2007 IEEE Nuclear Science Symposium Conference Record*, pp. 2847-2850, ISBN 1-4244-0923-3, Honolulu HI, WSA, Oct. 2007, IEEE, Honolulu

35

Forced Accretion and Assimilation Based on Self-Organizing Neural Network

Cheng-Yuan Liou and Wei-Chen Cheng
Department of Computer Science and Information Engineering
National Taiwan University, Taiwan
Republic of China

1. Introduction

The high level abstraction is developed layer after layer. This abstraction and generalization are important in language evolution and the study of mental processes. This chapter presents a self-organizing neural network based on cascading a series of layered perceptrons that resemble the development of the deep layers of the brain. This neural network is named 'SOM perceptron'. It maps all received patterns in a single class into a point in the deep layer space and maps different classes into different points. These widely separated class points can facilitate the abstract categorization and analogy operated in the mental process. Categorizing similar objects into small categories and different objects into different small categories can simplify a problem and reduce the mental load. Those small categories that contain similar objects will be successively combined into large categories to further reduce the load. This categorization process is a kind of divide and conquer process. This work emulates such process in successive layers. This design also resembles the formation of the corlica sensory representation. It preserves the topological relations among classes. It constructs a layered feed-forward network. Each layer attempts to split different classes and concentrate the same class patterns. It is trained using the discrimination differences between classes and trained independently layer after layer in a bottom-up manner. The class labels are not used in the training process to relieve the loading complexity of weights.

A bottom-up training method for the construction of the SOM perceptron Liou et al. (2000); Liou & Cheng (2008) is introduced in this chapter. This constructed SOM perceptron can be used as the mapping function to split different classes. It can be applied to split multiple classes.

The construction ideas are discussed in this section. There are four issues concerning the method. The first issue is the reduction of the number of hidden representations. The second one discusses a design method for the weights, such that the accomplishment of the SOM perceptron is guaranteed. The third one is why we prefer a bottom-up construction for the MLP and why we do not apply the BP to training it. The fourth issue discusses the different functions of front layers and rear layers of MLP under the BP training. These four issues will be presented in four subsections in this section. The method and architecture of the SOM perceptron are described in the next section. Experiments are included in the third section. Several related works are discussed in the last section.

We start with an introduction of the hidden representation. Let the set of all patterns be $X = \{\mathbf{x}^p, p = 1,\ldots,P\}$. Each pattern $\mathbf{x}^p$ is a D-dimensional column vector. The label function,

$C : R^D \rightarrow N$, maps each pattern, $\mathbf{x}^p$, to its class identity number, c_p. Let the set U_{c_i} contain all pattern pairs that belong to the same class c_i,

$$U_{c_i} = \{(\mathbf{x}^p, \mathbf{x}^q); C(\mathbf{x}^p) = C(\mathbf{x}^q) = c_i\}. \quad (1)$$

Let the set V_{c_i,c_j} contain all pattern pairs that belong to different classes,

$$V_{c_i,c_j} = \left\{(\mathbf{x}^p, \mathbf{x}^q); C(\mathbf{x}^p) = c_i, C(\mathbf{x}^q) = c_j, c_i \neq c_j\right\}. \quad (2)$$

Suppose there are L hidden layers in the network, $\{m = 1,2,\ldots,L\}$. Let the column vector $\mathbf{y}^{(p,m)}$ be the output vector of all neurons in the mth layer when the pattern $\mathbf{x}^p$ is fed to the input layer. $\mathbf{y}^{(p,m)}$ is the hidden (or internal) representation of $\mathbf{x}^p$ in the mth layer. We set $\mathbf{y}^{(p,0)} = \mathbf{x}^p$ for $m = 0$. Let n_m denote the total number of neurons in the mth layer. The collection of all hidden representations (HRs) of the mth layer is

$$\mathbf{Y}^m = \left\{\mathbf{y}^{(p,m)}, p = 1,\ldots,P\right\}.$$

A HR may be the same for different patterns, that is, a many-to-one mapping, $\mathbf{y}^{(p,m)} = \mathbf{y}^{(q,m)}$ for $\mathbf{x}^p \neq \mathbf{x}^q$. Let $\|\mathbf{Y}^m\|$ be the total number of distinct HRs in the set $\mathbf{Y}^m$.

1.1 First issue: reduced number of hidden representations

The HRs, $\mathbf{Y}^m$, have been studied in Liou & Yu (1995). All patterns have their HRs in each layer. These HRs are the output vectors of the hidden layers for all input patterns. They are all binary codes when the hard-limiting activation function is applied to all neurons. So, $\mathbf{y}^{(p,m)}$ is a binary code. The n_m decision hyperplanes of the neurons in the mth layer divide the n_{m-1} dimensional space of the preceding $(m-1)$th layer into nonoverlapped small decision areas and code these areas with binary codes. Let A^m be the collection of all areas, $A^m = \{\mathbf{a}_i^m, i = 1\ldots\|A^m\|\}$. The total number of these areas is less than or equal to $\Sigma_{k=0}^{n_{m-1}}\binom{n_m}{k}$, $\|A^m\| \leq \Sigma_{k=0}^{n_{m-1}}\binom{n_m}{k}$ Mirchandini & Cao (1989). Each area has a convex polyhedral shape.

Each of the codes, $\mathbf{y}^{(p,m)} \equiv \mathbf{a}_i^m$, represents all patterns (or HRs) contained in a single divided polyhedral area, $\{\mathbf{y}^{(p,m-1)}; \mathbf{y}^{(p,m-1)} \in \mathbf{a}_i^m\}$. This could be a many-to-one mapping. Each area is a small category containing many similar patterns. Note that certain areas may not contain any representation, $\|A^m\| \geq \|\mathbf{Y}^m\|$. According to the study in Liou & Yu (1995), the total number of HRs will be much reduced, generally, in a layer that is far from the input layer. This means

$$\left\|\mathbf{Y}^L\right\| \ll \ldots \ll \left\|\mathbf{Y}^2\right\| \ll \left\|\mathbf{Y}^1\right\| \ll P.$$

The reduced number in the mth layer, $\left\|Y^{m-1}\right\| - \|Y^m\|$, is the pattern complexity resolved in the mth layer. In many cases, $\left\|Y^{m-1}\right\| >> \|Y^m\|$. This reduction is very useful for the abstraction and isolation of the whole class patterns. Ideally, this number can be reduced to the number of classes, $\left\|\mathbf{Y}^L\right\| = \text{'}total\ number\ of\ classes\text{'} = \|C\|$. This makes the design possible for the SOM perceptron. Figure 1 illustrates the design idea.

1.2 Second issue: constructive weight design for SOM perceptron

The method in Liou & Yu (1994) provided a weight design for each layer, see program in web Liou (2000a). This design can be used in the SOM perceptron. This guarantees the accomplishment of the SOM perceptron. According to the design, the upper bound of

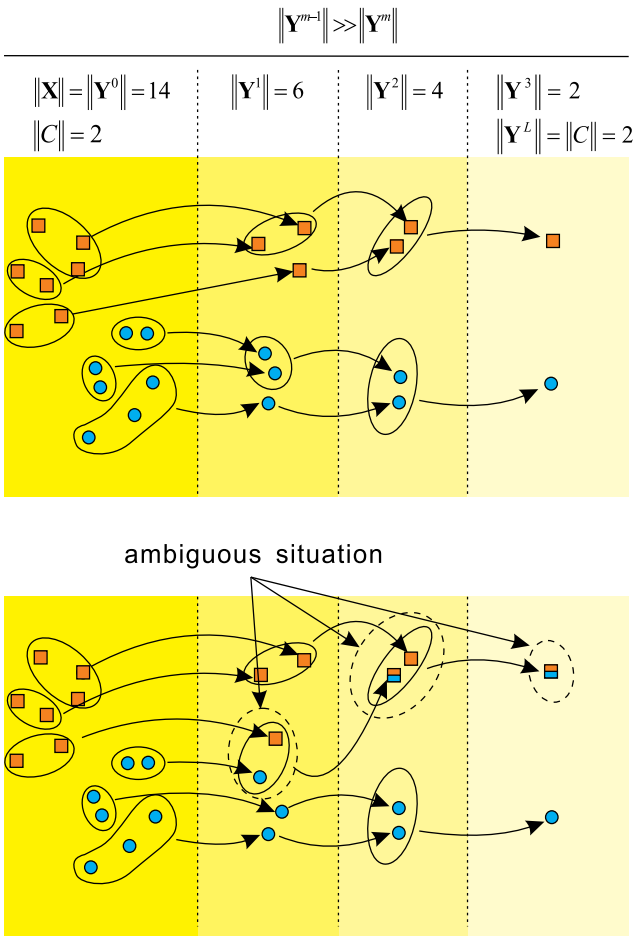


Fig. 1. The top diagram plots the reduction and the bottom plots an ambiguity representation in the second hidden layer.

the number of neurons in the mth layer required for solving a general-position two-class classification problem is

$$\left\lceil \frac{\|\mathbf{Y}^{m-1}\|}{n_{m-1}} \right\rceil \geq n_m.$$

For the number of neurons in the first hidden layer, n_1, the bound is $\left\lceil \frac{P}{D} \right\rceil \geq n_1$. With this weight design, the reduced number in the last layer L is guaranteed, that is $\|\mathbf{Y}^L\| = \|C\|$.

The method in Liou & Yu (1994) also showed a constructive weight design for the MLP. To illustrate this method, we show a general-position two-class classification problem. This problem can be solved with two hidden layers. This construction is very different from BP algorithms. It solves the complexity, $\Sigma_{k=0}^{n_{m-1}} \binom{n_m}{k}$, in the succeeding MLP layers.

Figure 2(a) illustrates the design for a two-class problem, $c_1 = 1$ and $c_2 = 2$, in a two dimensional space, $D = n_0 = 2$. In this D space, a center line of a strip, $\overline{\mathbf{x}^p \mathbf{x}^q}$, is allocated for two near patterns, $\mathbf{x}^p$ and $\mathbf{x}^q$, that are in a same class c_1, $\mathbf{x}^p \in c_1$ and $\mathbf{x}^q \in c_1$. We assume

that c_1 contains fewer number of patterns than that of c_2. Then, this center line is split into two parallel lines, line $\overline{a}$ and line $\overline{b}$. They are in the two opposite sides of this center line and parallel to the center line, $\overline{a} \| \overline{\mathbf{x}^p \mathbf{x}^q} \| \overline{b}$. For $\overline{a}$, pick a pattern $\mathbf{x}^r$, $\mathbf{x}^r \in c_2$, where $\mathbf{x}^r$ is closest to $\overline{\mathbf{x}^p \mathbf{x}^q}$. $\mathbf{x}^r$ and $\overline{a}$ are in the same side of $\overline{\mathbf{x}^p \mathbf{x}^q}$. Plot a parallel line $\overline{a}^r$, $\overline{a}^r \| \overline{\mathbf{x}^p \mathbf{x}^q}$, that passes the pattern $\mathbf{x}^r$. Pick a pattern $\mathbf{x}^s$, $\mathbf{x}^s \in c_1$, that is in between the two lines, $\overline{a}^r$ and $\overline{\mathbf{x}^p \mathbf{x}^q}$, and is the closest pattern to the line $\overline{a}^r$. Plot a parallel line, $\overline{a}^s$, $\overline{a}^s \| \overline{\mathbf{x}^p \mathbf{x}^q}$, that passes the pattern $\mathbf{x}^s$. Plot a decision border line, $\overline{a}^{rs}$, right in between the two parallel lines, $\overline{a}^r$ and $\overline{a}^s$. $\overline{a}^{rs}$ has wide margin between the pair $(\mathbf{x}^r, \mathbf{x}^s)$.

The two patterns, $\mathbf{x}^r$ and $\mathbf{x}^s$, serve as the margin-limiting stops of the region in between the two lines, $\overline{a}^r$ and $\overline{a}^s$. The decision border line $\overline{b}^{uv}$ for $\overline{b}$ can be accomplished in a similar way on the other side of $\overline{\mathbf{x}^p \mathbf{x}^q}$. These two decision lines are used as the two neurons in the mth layer. All patterns in between the two lines $\overline{a}^{rs}$ and $\overline{b}^{uv}$ belong to the same class c_1. They are well isolated from the patterns in the other class c_2. The stops $\mathbf{x}^r$ and $\mathbf{x}^s$ could be different from the support vectors in support vector machine (SVM) Boser et al. (1992).

The number of patterns in between the two lines $\overline{a}^{rs}$ and $\overline{b}^{uv}$ are one of the two factors of the two neurons. The other factor is the width between these two decision lines. These two factors are useful in the determination of the significance of these two neurons. Those neurons with large factors are preferable and will be preserved with high priority in many training operations. Small factor neurons will be eliminated occasionally.

An example of the typical decision regions is illustrated in Figure 2(b). The decision regions contain four bar-like strips. There exists physiological evidences on receptive fields, $D = 2$, for such bar-like strips, Daugman (1980); Dobbins et al. (1987).

There are many techniques to pick the center patterns $\mathbf{x}^p$ and $\mathbf{x}^q$ to build a strip. One way to do this is to select all patterns, $\{\mathbf{x}^p, \mathbf{x}^q, \mathbf{x}^r, \mathbf{x}^s, \mathbf{x}^u$ and $\mathbf{x}^v\}$, in a predefined neighborhood region. The size of this region can be tuned during the training process. One may include a penalty cost to set the borders $\overline{a}^{rs}$ and $\overline{b}^{uv}$ in a way similar to that for SVM.

Note that both the number of neurons and the number of layers in certain operations of the tiling algorithm [Mezard and Nadal, 1989] are highly sensitive to the setting of the origin, the absolute coordinates, of the patterns. Many other neural networks, such as SOM, are also sensitive to the setting of the origin. The relative distances among patterns are used in the method in Liou & Yu (1994). This relative distance gives the classification quality. To our knowledge, the performance of this method exceeds that of SVM. The relative distances will be used in this work.

Figure 2(c) illustrates an example for the general-position two-class classification problem. A single 'AND' function is used for a neuron in the second hidden layer to represent the patterns in one individual strip. A global 'OR' function is used in the output neuron to represent all patterns in class c_1 that are in all strips. To our knowledge, this is the simplest MLP architecture in many aspects. Any difficult isolated polyhedral region that is encompassed by several neurons can be included in this archietecture by adding an extra 'OR' weight connecting this region. The performance of this simple architecture exceeds that of tiling algorithm.

1.3 Third issue: front layers must be trained correctly to get perfect performance for MLP

The reason why we prefer a constructive way from the bottom layer is based on the work Liou & Yu (1995). It introduced a layered binary tree, named 'AIR' tree, that can trace the error neurons in a latent hidden layer that is far from the output layer and close to the input layer, see program in Liou (2000b). The error shows that certain mixed patterns from both

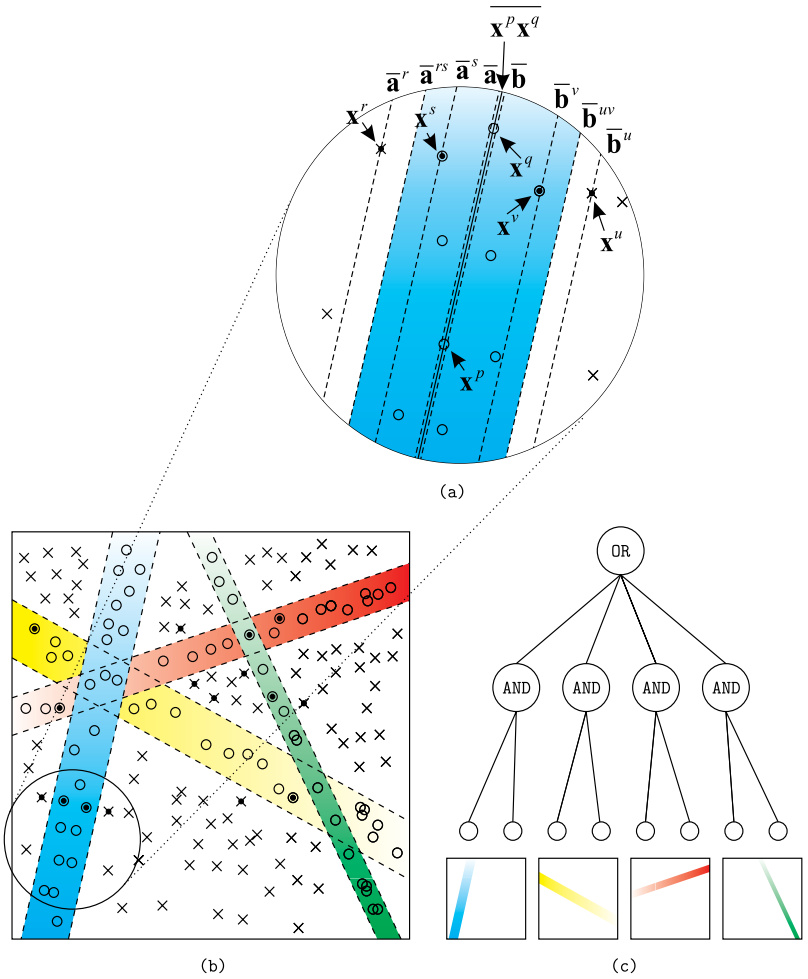


Fig. 2. The concept of the weight design method in Liou & Yu (1994).

classes are represented in a same code. This means that certain area code $\mathbf{y}^{(p,m)} \equiv \mathbf{a}_i^m \in A^m$ represents different class patterns, $\mathbf{y}^{(p,m)} = \mathbf{y}^{(q,m)}$ for $(\mathbf{x}^p, \mathbf{x}^q) \in V_{c_i,c_j}$. Any area that contains the mixed patterns, $(\mathbf{x}^p, \mathbf{x}^q) \in V_{c_i,c_j}$, is named 'ambiguity' area. An area that contains patterns belong to a single class is named 'unambiguity' area. The joint nodes of the tree are the HR codes. This tree exposes the latent error areas and their neurons. According to this tree, any BP algorithms cannot correct the latent error by adjusting the weights in its succeeding layers that near the output layer. The front layers must be trained correctly in order to send discriminated signals (HRs) to their succeeding layers. This suggests that one has to accomplish the MLP layer after layer in a bottom-up construction way.

Figure 3 illustrates an example tree for a two-class problem with the MLP network, $n_0 = 2$, $n_1 = 3$, $n_2 = 3$, $n_3 = 1$. There are 19 patterns in the class c_1 and 12 patterns in the class c_2, $P = 31$. In the input space, $n_0 = 2$, the three decision lines of the first hidden layer divide the input space into $\|A^1\| = 7$, $\|A^1\| = \Sigma_{k=0}^{n_0}\binom{n_1}{k}$, decision areas, $A^1 = \{\mathbf{a}_i^1, i = 1,2,3,5,6,7\}$. The input space is divided into 7 areas and coded with 7 binary codes shown in the bottom layer

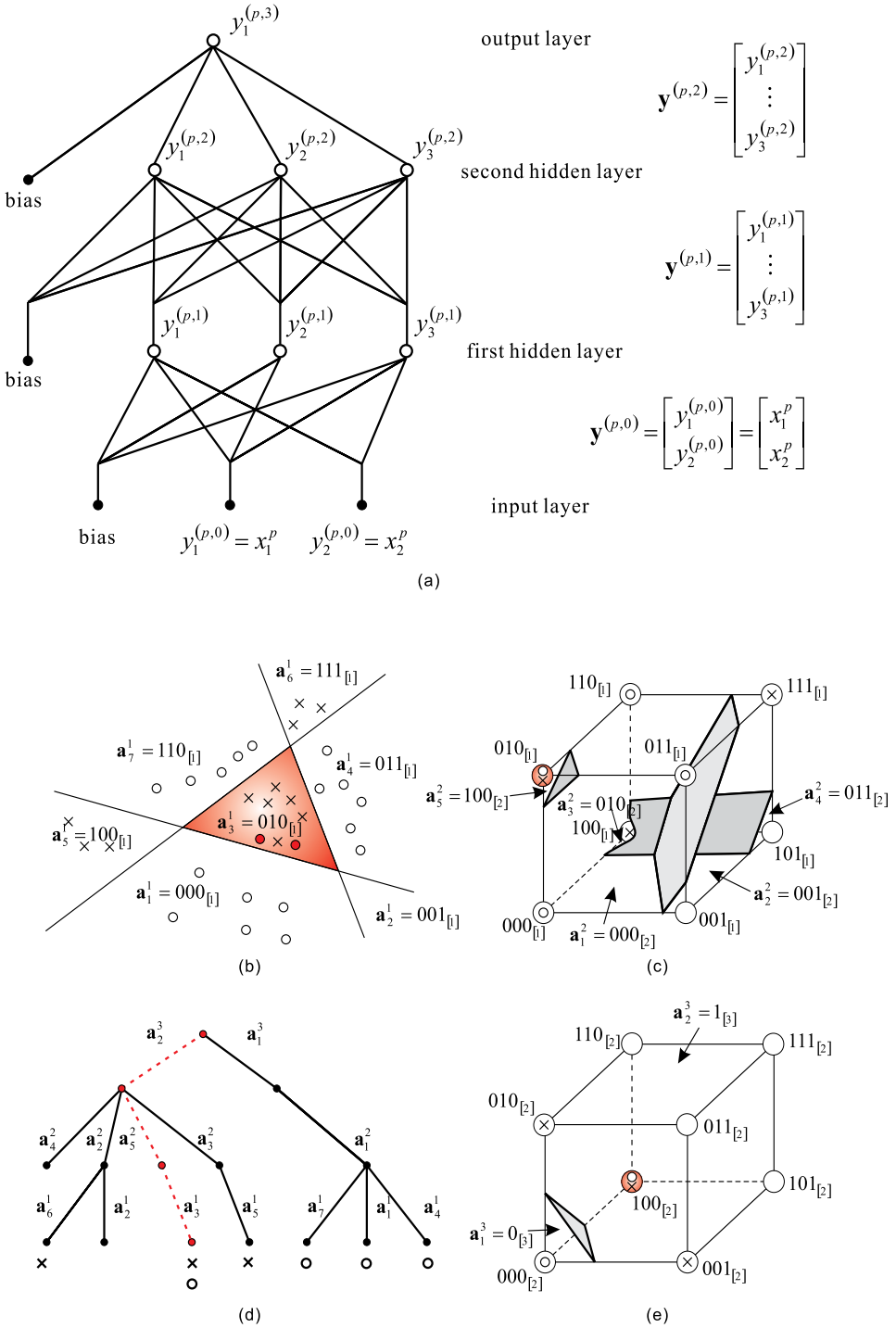


Fig. 3. The AIR tree for MLP Liou & Yu (1995).

of the tree. Note that these seven codes are symbols and are not binary numbers, for example $a_7^1 = 110_{[1]} \neq 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$. Each code represents all patterns in a single decision area. One of the areas is void, $\mathbf{a}_2^1$, and contains no pattern. No input pattern can activate this HR code. One area, $\mathbf{a}_3^1$, contains mixed patterns from both classes, this area is an ambiguity area. There exists one latent error and it can be traced by the tree, see the dash line in the figure. This kind error can be fixed locally using the patterns in the ambiguity area by adding extra neurons in this area. This error cannot be resolved by adjusting the weights in the layers n_2 and n_3 using any BP algorithms. When one deletes a significant neuron, many ambiguity areas will be generated. A less significant neuron will generate less number of ambiguity areas when it is deleted. One of two neurons can be pruned when both produce the same (or different) responses for all patterns. A neuron has the same response for all patterns can be pruned. One can build the tree for any training algorithms as an online monitor to visualize the latent contents. With this tree, one can see and understand how a MLP solve a problem. This tree can facilitate many applications.
Note that the latent errors cause the global mean square error (MSE) of the whole network stays significantly high constant for certain unpredictable period of time during the BP training. The BP may force the outputs of the error neurons to zero to suppress (or cover) the errors. This will do more harm than help to resolve the ambiguity areas. This is the major reason for slow learning and convergence to local minima of the BP algorithm. The presence of a high constant MSE is an important pointer of when we should fix the error neurons.
As for a single neuron, there exists at most one high constant MSE period Liou et al. (2005) Huang & Liou (2010) and the duration of that period is roughly proportion to the total number of boolean functions available near the origin point of input space.

1.4 Fourth issue: different functions of front layers and rear layers in MLP under BP training

The study in the work Liou & Yu (1995) further identifies the function of the front layers during the supervised BP training. It shows that categorization into different classes is the main function for those front layers. This means that the detailed identity of each class, the class label, is not used in the categorization. This suggests that the front layers can be successfully trained by using the discrimination differences between classes as the object function. The discrimination differences may come from parts of the distributed representation as discussed by Elman (1991).

Note that the most important conclusion of the work, Liou & Yu (1995), is that the MLP must be accomplished in a bottom-up manner in order to get perfect performance. It is hopeless to use the BP algorithm to get a global solution for a medium-sized MLP network. Any BP algorithm will converging to a local minimum solution. This work also shows that the MLP with perfect performance is a certain kind of " logical machine " that build a logical map for the events in the training dataset. This map can facilitate the comprehension of the events.

The 'SIR' method in Liou et al. (2000) provides the categorization object function based on the discrimination differences between classes, see program in web Liou (2000c). The front layers can be trained layer after layer using this object function starting from the first hidden layer. Perfect categorization and production of right signals can be accomplished for each layer Liou & Cheng (2008). These front layers are served, suitably, as the SOM perceptron. The SOM perceptron will utilize the discrimination differences between classes to train the front layers. It will not use class labels in its training process. Labels are not used as the

supervised primitives. Rather, the categorization is accreted under the side direction of those discrimination differences. It carry the similar notion as that in Elman (1991).
The work Liou & Yu (1995) also identifies the function of rear layers that are near the output layer. It shows that labeling with class labels is the main function. These front and rear functions comprise the supervised BP for the MLP. We will include a labeling sector that contains several layers after the SOM perceptron. The object function for the labeling sector is the class labels. One can apply any wide margin techniques to train this sector.
We will use the discrimination differences between classes to train each front layer starting from the first hidden layer. Any ambiguity area should be resolved for each layer. All ambiguity errors can be traced using the AIR tree and fixed locally and independently. This can be done either by the weight design method Liou & Yu (1994) for those error neurons or by the retraining method Liou & Yu (1995). For a severe ambiguity area, one may insert additional neurons and train them using the HRs in the ambiguity area. Note that any added neuron will not destroy the unambiguity area.
A second hidden layer is added to the first hidden layer when the outputs of the first hidden layer cannot produce well isolated HRs for each class. When a hidden layer can produce well isolated signals for different classes, it will be served as the last front layer and as the output of the SOM perceptron. We expect that the number of reduced HRs of the last layer will be equal to the number of classes, $\left\|\mathbf{Y}^L\right\| = \|C\|$.

2. Method

Figure 4 illustrates the SOM perceptron and the labeling sector. The SOM perceptron consists of layered neurons.
The relative distance between two patterns will be used in this work. For the pair patterns in the same class, $(\mathbf{x}^p, \mathbf{x}^q) \in U_{c_i}$, each layer is trained by using the energy function Liou et al. (2000),Liou & Cheng (2008),

$$E^{att}(\mathbf{x}^p, \mathbf{x}^q) = \frac{1}{2}\left\|\mathbf{y}^{(p,m)} - \mathbf{y}^{(q,m)}\right\|^2, \tag{3}$$

to reduce the distance between their output vectors, $\left\|\mathbf{y}^{(p,m)} - \mathbf{y}^{(q,m)}\right\|$. For the pair, $(\mathbf{x}^p, \mathbf{x}^q) \in V_{c_i,c_j}$, each layer is trained by using the energy function,

$$E^{rep}(\mathbf{x}^p, \mathbf{x}^q) = \frac{-1}{2}\left\|\mathbf{y}^{(p,m)} - \mathbf{y}^{(q,m)}\right\|^2, \tag{4}$$

to increase the distance between their output vectors. The discrimination information between classes is implicitly used in these two energies. They comprise the self-organizing principle for evolving the HRs on each layer space. We expect that these energies can maximally utilize all inherent discrimination differences among class patterns to separate the classes. Note that the class labels are not used in these two object functions. The labels will be used only in the labeling sector.
The network is constructed layer after layer, starting from $L = 1$. A new hidden layer is added, $L^{new} = L^{old} + 1$, whenever L^{old} layers cannot accomplish the isolation. All weights of the trained L^{old} layers are fixed during the training of the added layer, $m = L^{old} + 1$.
The weight matrix which connects the output of the $(m-1)$th layer and the input of the mth layer, is denoted by W^m. The W^1 connects the input layer and the first hidden layer. In this paper, 'module W^m' is used for representing the weights of the mth layer. Applying

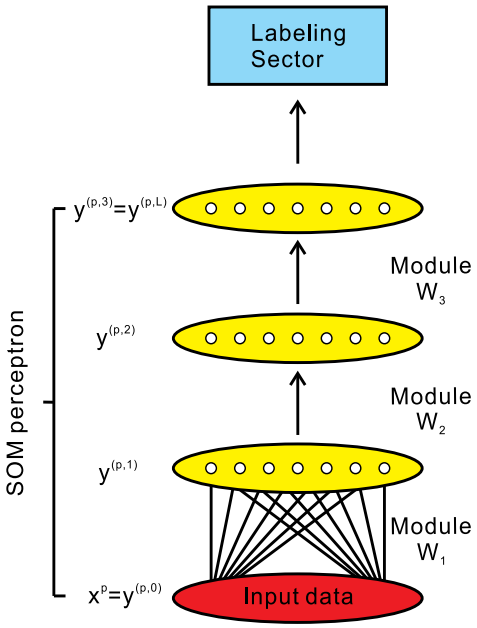


Fig. 4. The SOM perceptron and labeling sector.

the gradient descent method to the added layer, the two energies can be reduced efficiently during training iterations. The successfully trained network is used as the SOM perceptron to map the pattern, $\mathbf{x}^p$, to the output space, $\mathbf{y}^{(p,L)}$.

Algorithm

Each time a new hidden layer, L^{new}=$L^{old}+1$, is added, its weights are adjusted by the gradient descent method based on the energies (3) and (4). The weights of all trained layers, L^{old}, are fixed. Suppose there are two classes, $\{c_1=1, c_2=2\}$. The training algorithm is in below.

1. For each added layer W^m, W^m from W^1 to W^L
2. For limited epochs
3. Pick any two patterns in the same class, $\mathbf{x}^p$ and $\mathbf{x}^q$, which satisfy the following condition

$$(\mathbf{x}^p,\mathbf{x}^q)=\underset{\{(\mathbf{x}^i,\mathbf{x}^j)\in U_1 \text{ or } (\mathbf{x}^i,\mathbf{x}^j)\in U_2\}}{\arg\max}\left\|\mathbf{y}^{(i,m)}-\mathbf{y}^{(j,m)}\right\|^2. \tag{5}$$

Among the pair patterns in the same class, either in U_1 or in U_2, the two patterns $(\mathbf{x}^p,\mathbf{x}^q)$ have the longest distance in the output space of the mth layer.

4. Find the pair patterns, $\mathbf{x}^r$ and $\mathbf{x}^s$ in different classes, which satisfy

$$(\mathbf{x}^r,\mathbf{x}^s)=\underset{(\mathbf{x}^i,\mathbf{x}^j)\in V_{1,2}}{\arg\min}\left\|\mathbf{y}^{(i,m)}-\mathbf{y}^{(j,m)}\right\|^2. \tag{6}$$

The pair patterns $(\mathbf{x}^r,\mathbf{x}^s)$ have the shortest distance in the output space of the mth layer.

5. Adjust the module W^m by

$$\begin{aligned} \nabla W^m &\leftarrow \eta^{att}\frac{\partial E^{att}(\mathbf{x}^p,\mathbf{x}^q)}{\partial W^m}+\eta^{rep}\frac{\partial E^{rep}(\mathbf{x}^r,\mathbf{x}^s)}{\partial W^m} \\ W^m &\leftarrow W^m-\nabla W^m, \end{aligned} \tag{7}$$

where η^{att} and η^{rep} are learning rates.

The gradients of E^{att} and E^{rep} in (7) are

$$\begin{aligned} \frac{\partial E^{att}(\mathbf{x}^p,\mathbf{x}^q)}{\partial W^m} = &+\begin{bmatrix} \left(y_1^{(p,m)}-y_1^{(q,m)}\right)\left(1-\left(y_1^{(p,m)}\right)^2\right) \\ \vdots \\ \left(y_{n_m}^{(p,m)}-y_{n_m}^{(q,m)}\right)\left(1-\left(y_{n_m}^{(p,m)}\right)^2\right) \end{bmatrix}\left[y_1^{(p,m-1)},\ldots,y_{n_{m-1}}^{(p,m-1)},-1\right] \\ &-\begin{bmatrix} \left(y_1^{(p,m)}-y_1^{(q,m)}\right)\left(1-\left(y_1^{(q,m)}\right)^2\right) \\ \vdots \\ \left(y_{n_m}^{(p,m)}-y_{n_m}^{(q,m)}\right)\left(1-\left(y_{n_m}^{(q,m)}\right)^2\right) \end{bmatrix}\left[y_1^{(q,m-1)},\ldots,y_{n_{m-1}}^{(q,m-1)},-1\right] \end{aligned}$$

and

$$\begin{aligned} \frac{\partial E^{rep}(\mathbf{x}^p,\mathbf{x}^q)}{\partial W^m} = &-\begin{bmatrix} \left(y_1^{(p,m)}-y_1^{(q,m)}\right)\left(1-\left(y_1^{(p,m)}\right)^2\right) \\ \vdots \\ \left(y_{n_m}^{(p,m)}-y_{n_m}^{(q,m)}\right)\left(1-\left(y_{n_m}^{(p,m)}\right)^2\right) \end{bmatrix}\left[y_1^{(p,m-1)},\ldots,y_{n_{m-1}}^{(p,m-1)},-1\right] \\ &+\begin{bmatrix} \left(y_1^{(p,m)}-y_1^{(q,m)}\right)\left(1-\left(y_1^{(q,m)}\right)^2\right) \\ \vdots \\ \left(y_{n_m}^{(p,m)}-y_{n_m}^{(q,m)}\right)\left(1-\left(y_{n_m}^{(q,m)}\right)^2\right) \end{bmatrix}\left[y_1^{(q,m-1)},\ldots,y_{n_{m-1}}^{(q,m-1)},-1\right]. \end{aligned}$$

Figure 5 illustrates an example of the algorithm. In this figure, the intra-class pattern pairs are $U_1=\left\{\left(\mathbf{x}^1,\mathbf{x}^2\right),\left(\mathbf{x}^1,\mathbf{x}^3\right),\left(\mathbf{x}^2,\mathbf{x}^3\right)\right\}$ and $U_2=\left\{\left(\mathbf{x}^4,\mathbf{x}^5\right)\right\}$. The inter-class pattern pairs are $V_{1,2}=\left\{\left(\mathbf{x}^1,\mathbf{x}^4\right),\left(\mathbf{x}^1,\mathbf{x}^5\right),\left(\mathbf{x}^2,\mathbf{x}^4\right),\left(\mathbf{x}^2,\mathbf{x}^5\right),\left(\mathbf{x}^3,\mathbf{x}^4\right),\left(\mathbf{x}^3,\mathbf{x}^5\right)\right\}$. The intra-class pair has the maximal distance in the output space is

$$\left(\mathbf{x}^1,\mathbf{x}^2\right)=\underset{\{(\mathbf{x}^i,\mathbf{x}^j)\in U_1\ or\ (\mathbf{x}^i,\mathbf{x}^j)\in U_2\}}{\arg\max}\left\|\mathbf{y}^{(i,1)}-\mathbf{y}^{(j,1)}\right\|,$$

and the inter-class pair has the minimal distance is

$$\left(\mathbf{x}^3,\mathbf{x}^4\right)=\underset{\{(\mathbf{x}^i,\mathbf{x}^j)\in V_{1,2}\}}{\arg\min}\left\|\mathbf{y}^{(i,1)}-\mathbf{y}^{(j,1)}\right\|.$$

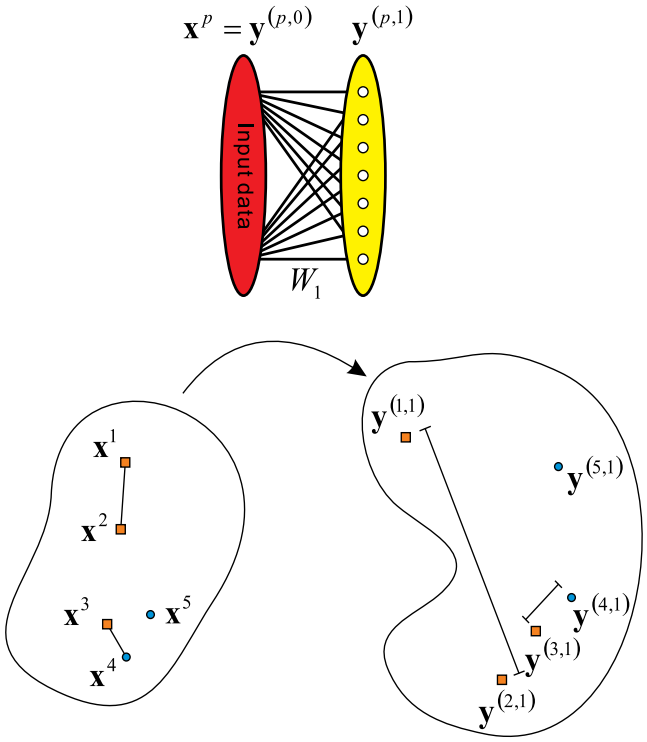


Fig. 5. A snapshoot of the first hidden layer of the algorithm.

The two energies are

$$E^{rep}\left(\mathbf{x}^3,\mathbf{x}^4\right) = \frac{-1}{2}\left\|\mathbf{y}^{(3,1)} - \mathbf{y}^{(4,1)}\right\|^2$$

and

$$E^{att}\left(\mathbf{x}^1,\mathbf{x}^2\right) = \frac{1}{2}\left\|\mathbf{y}^{(1,1)} - \mathbf{y}^{(2,1)}\right\|^2.$$

3. Experimental analysis

Two artificial datasets are used in the simulations. One is a two-class problem and the other is a three-class problem. Four real world datasets are also used in the simulations.

3.1 Two-class problem

Figure 6(b) plots the trained result for the two-class patterns, $c_i \in \{1,-1\}$, in the 2D plane, $n_0 = 2$. The border of these two-class patterns is $(x_1)^3 + 0.1x_1 = x_2$. Pattern points with the same color are in the same class. There are five neurons in each layer, $\{n_m = 5, m \in \{1,\ldots,L\}\}$. The SOM perceptron is trained layer after layer until it produces well isolated HRs for each class. We set the isolation condition for inter-class HRs as

$$\min_{(\mathbf{x}^p,\mathbf{x}^q)\in V_{1,2}} \left\|\mathbf{y}^{(p,L)} - \mathbf{y}^{(q,L)}\right\|^2 \approx 2^2 \times n_L, \tag{8}$$

and the condition for intra-class HRs as

$$\max_{\{(\mathbf{x}^p,\mathbf{x}^q)\in U_1 \text{or } (\mathbf{x}^p,\mathbf{x}^q)\in U_2\}} \left\|\mathbf{y}^{(p,L)}-\mathbf{y}^{(q,L)}\right\|^2 \approx 0. \tag{9}$$

The learning rates are $\eta^{att}=0.01$ and $\eta^{rep}=0.1$. This means that the repelling force is weighted stronger than the attractive force. The successful isolation is reached when $L=2$. We set one neuron, $n_1^c=1$, in the labeling sector as the output layer and use the class identities, $c_i\in\{1,-1\}$, to train this neuron. Figure 6(b) plots the trained result.
We also compare the result with those obtained by the MLP in Figure 6(a), and SVM in Figure 6(c). The MLP is a multilayer perceptron with two hidden layers, $n_1^{MLP}=n_2^{MLP}=5$. This MLP is trained by the supervised BP. The Gaussian kernel, $K(\mathbf{u},\mathbf{v})=\exp\left(\|\mathbf{u}-\mathbf{v}\|^2\right)$, is used in SVM Chang & Lin (2001).
The boundary in Figure 6(b) is much more close to the border than the result of the supervised MLP in Figure 6(a). Using the polynomial kernel, the boundary learned by SVM is also close to the border.

3.2 Multiple-class problem

Figure 7 plots the training patterns sampled from three classes separated by three ellipses, $c_i\in\{1,2,3\}$.
In this simulation, we train four SOM perceptrons with different number of neurons in each layer, $\{n_m=5, n_m=7, n_m=9, n_m=11\}$. Each layer is trained by 1000 epochs. The isolation conditions (8, 9) are used in this simulation to stop the addition of a new layer. The learning rates are $\eta^{att}=0.01$ and $\eta^{rep}=0.1$. The values of the isolation conditions for each layer

$$MinInterClass(m)=\min_{(\mathbf{x}^p,\mathbf{x}^q)\in\{V_{1,2},V_{1,3},V_{2,3}\}}\left\|y^{(p,m)}-y^{(q,m)}\right\|^2 \tag{10}$$

and

$$MaxIntraClass(m)=\max_{(\mathbf{x}^p,\mathbf{x}^q)\in\{U_1,U_2,U_3\}}\left\|y^{(p,m)}-y^{(q,m)}\right\|^2, \tag{11}$$

are recorded and plotted in the Figure 8.
When the isolation is reached, we set two layers in the labeling sector with $n_1^c=2$ and $n_2^c=3$ and use the class identities to train these two layers. In the layer $n_2^c=3$, each neuron represents a single class.

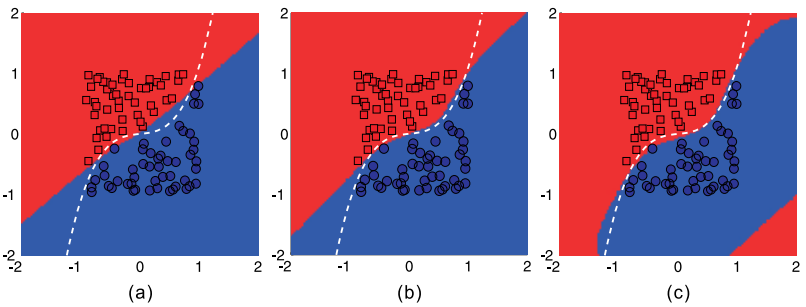


Fig. 6. The dash lines indicate the intrinsic border. (a) The trained result of MLP. (b) The result of SOM perceptron. (c) The result of SVM.

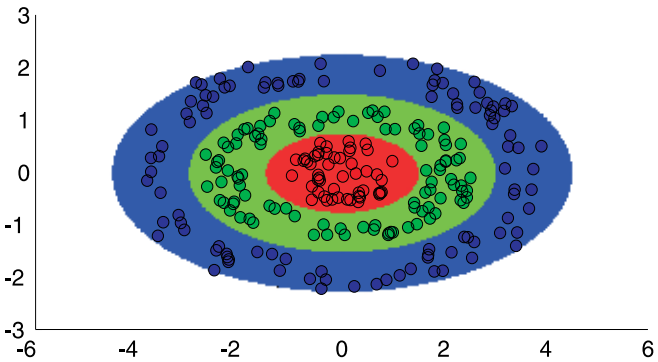


Fig. 7. The patterns in three classes.

We employ the SOM to visualize the output signals, $\mathbf{y}^{(j,m)}$, of each layer, to see the isolation of classes. The neurons of the SOM are placed on the regular points, see Figure 9. The SOM consists of 10×10 neurons.

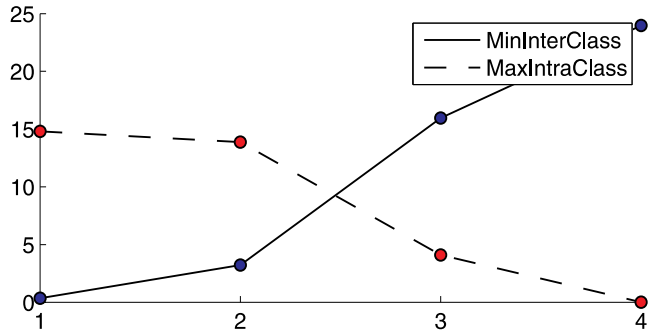


Fig. 8. Recorded isolation conditions, MinInterClass (10) and MaxIntraClass (11), for each layer, $m = 1,2,3,4$.

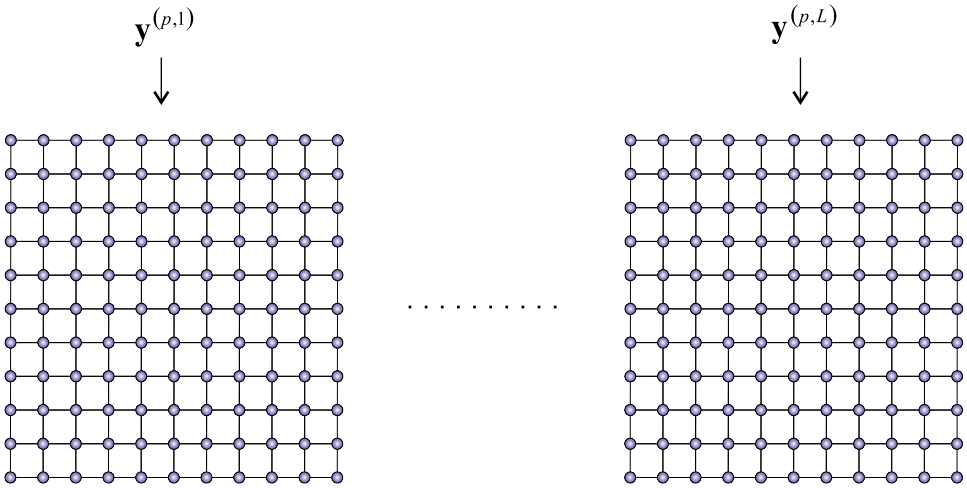


Fig. 9. The SOM used for visualization.

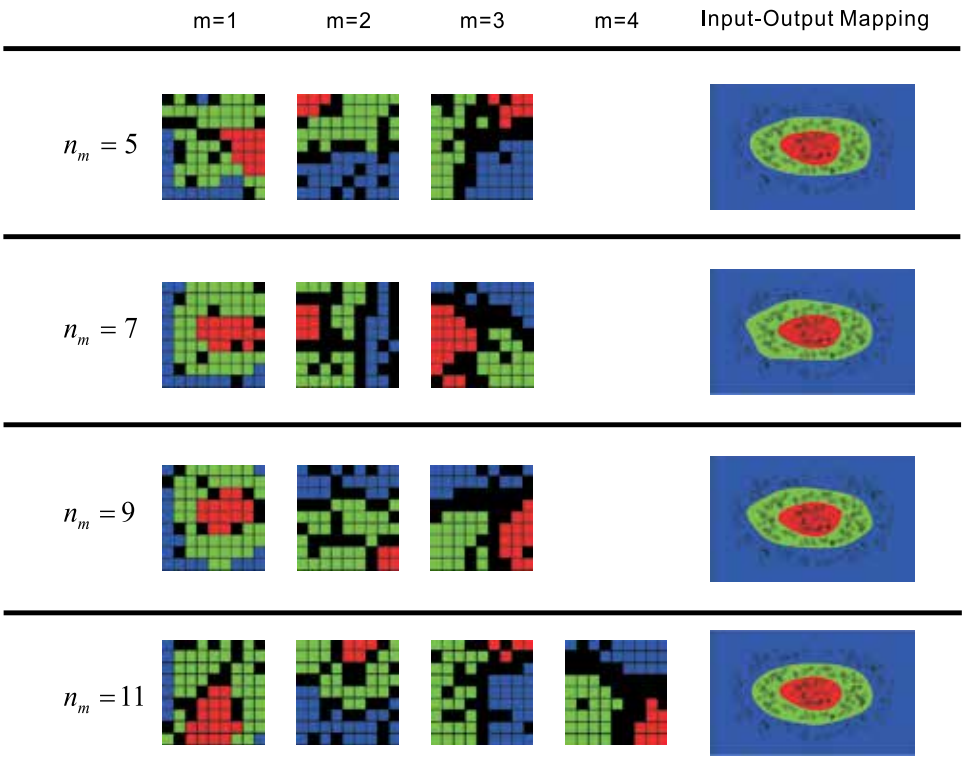


Fig. 10. The results of using the SOM to visualize the isolation of the output vectors of each layer. The images on the right column display the mapping relation between input space and output space of the labeling sector. Each neuron of the layer $n_2^c = 3$ indicates a single class (one color) in the input pattern space.

Figure 10 plots the SOM results for each layer. Each pixel denotes a SOM neuron. The pattern color is marked on its winner neuron. This figure shows that well isolated signals are gradually accomplished in the last few layers. The output signals of the last layer have three concentrated points in the SOM.

3.3 Real datasets

The Sonar Dataset, Wine Dataset, Ionosphere data and promoter gene sequences will be used in these experiments. Four machine learning techniques, k-NN (k-nearest neighbors algorithm), SOM perceptron, MLP and SVM, are compared using the 10-fold cross-validation. The dataset is randomly split into ten partitions, nine of them are used in the training process and the rest one is used in the testing process. The results are the average of the 10-fold cross-validation. The settings in the labeling sector for all datasets are listed in Table 1. The parameters of SVM are the cost C for the error tolerance and the gamma γ in the Gaussian kernel. Parameter k indicates the number of neighboring cells in the k-NN algorithm. The values of C, γ, and k, are optimized using an inner 10-fold cross-validation procedure. The settings that produce the lowest errors are used to learn the models with the whole set of training data. The MLP has two hidden layers. All parameters are listed in Table 1 and Table 2. The values of the input patterns are normalized within the range $[-1,1]$.

	k-NN	SOMP
		(n_m, n_1^c, n_2^c)
Sonar	(3,1,1,3,3)	(35,5,1)
	(3,1,1,1,1)	
Wine	(3,15,13,11,19)	(10,5,3)
	(15,19,13,15,11)	
Ionosphere	(1,3,1,1,1)	(10,5,1)
	(1,11,1,1,1)	
Promoters	(3,5,1,3,1)	(100,40,1)
	(3,3,3,3,3)	

Table 1. Parameters in k-NN and SOM perceptron

Table 3 and Table 4 show that the SOM perceptron is competitive and practicable in real world applications. The column 'SOMP' contains the results of SOM perceptron. As we expected, the training accuracy of the SOM perceptron is perfect 100%. In all our experiments the number of layers L is always small.

4. Summary

The topological form and correspondence on the L layer space, $\mathbf{Y}^L$, may resemble the fine organization of the corlica sensory representation, Homunculus Penfield & Rasmussen (1950) Kohonen (1982). There exists one and only one representation point on outer cortex for a sensory input.

A high dimensional combined representation point may indicate and serve as the state transition for two consecutive states and as the state transformation between two kinds of sensory inputs. For example, the transition between successive two states can be represented by a combined pattern in some sense similar to the bi-directional associative model Kosko (1988).

4.1 RBM and SOM perceptron

The HRs developed by the restricted Boltzmann machine (RBM) Salakhutdinov et al. (2007) and the Boltzmann machine Ackley et al. (1985); Liou & Lin (1989) are very different from those developed in the SOM perceptron. RBM is also constructed in a bottom-up manner. Each individual layer is developed freely and evolved independently. The labels are not used in RBM. The discrimination differences between classes are not used directly for the

	SVM		MLP	
	C	gamma	n_1^{MLP}	n_2^{MLP}
Sonar	$(2^5,2^1,2^5,2^3,2^7)$	$(2^{-3},2^{-1},2^{-3},2^{-3},2^{-5})$	30	10
	$(2^7,2^7,2^3,2^3,2^5)$	$(2^{-5},2^{-5},2^{-1},2^{-3},2^{-3})$		
Wine	$(2^{-1},2^1,2^{-1},2^3,2^1)$	$(2^{-1},2^{-1},2^{-1},2^{-9},2^{-3})$	20	5
	$(2^5,2^{-1},2^1,2^{-1},2^1)$	$(2^{-5},2^{-1},2^{-1},2^{-1},2^{-3})$		
Ionosphere	$(2^3,2^3,2^3,2^3,2^1)$	$(2^{-1},2^{-1},2^{-1},2^{-1},2^{-5})$	20	5
	$(2^5,2^{-1},2^1,2^{-1},2^1)$	$(2^{-3},2^{-5},2^{-1},2^{-3},2^{-1})$		
Promoters	$(2^1,2^1,2^1,2^3,2^1)$	$(2^{-9},2^{-11},2^{-7},2^{-13},2^{-7})$	20	5
	$(2^5,2^1,2^3,2^1,2^5)$	$(2^{-15},2^{-9},2^{-11},2^{-9},2^{-11})$		

Table 2. Parameters in SVM and MLP algorithms

	Training Accuracy			
	k-NN	SOMP	MLP	SVM
Sonar	95.46%	100.00%	98.24%	100.00%
Wine	97.88%	100.00%	100.00%	99.56%
Ionosphere	97.69%	100.00%	99.46%	99.05%
Promoters	93.72%	100.00%	100.00%	100.00%

Table 3. The training accuracy on real dataset.

development of the RBM HRs. It is expected that the HRs developed in the RBM hidden layers can support the visual patterns to tolerate noisy patterns and variations of patterns. When the number of hidden neurons in a hidden layer is much less than that of visual patterns, the representation capacity is low and the HRs tend to encode those patterns with the complexity of each neuron layer. This encoding scheme has been extensively discussed in Ackley et al. (1985); Liou & Lin (1989). This kind encoding has been used to explain the mechanism of the RBM.

When the number of neurons is much larger than that of visual patterns, the representation capacity is high and there are so many alternative HRs for the visual representations. There exist so many alternative HRs for the visual support. There is no specific preferred HRs by the RBM. The SOM perceptron imposes the repelling energy to split different classes and the attraction energy to concentrate a class directly. It seeks those widely separated HRs to support and isolate the visual patterns.

4.2 Hebbian learning

It is generally accepted that the supervised BP algorithm is not biological plausible. The SIR learning Liou et al. (2000), weights degrading and enhancing mechanism for classes, keeps the Hebbian form for a single layer. It is biological possible. As for the Hebbian form, in the gradient formulas, $\frac{\partial E^{att}(\mathbf{x}^p,\mathbf{x}^q)}{\partial W^m}$ and $\frac{\partial E^{rep}(\mathbf{x}^p,\mathbf{x}^q)}{\partial W^m}$, the terms

$$\left(1-\left(y_i^{(p,m)}\right)^2\right) \text{ and } \left(1-\left(y_i^{(q,m)}\right)^2\right)$$

have nonnegative values. These two terms are the derivatives of the activation function $f(x)=\tanh(x)$. If we substitute 1 for these terms, the learning rule becomes

$$w_{ij}^m(n+1) \longleftarrow w_{ij}^m(n)+\eta\left(y_i^{(p,m)}(n)-y_i^{(q,m)}(n)\right)\left(y_j^{(p,m-1)}(n)-y_j^{(q,m-1)}(n)\right), \tag{12}$$

that is the Hebbian learning with 'bi-patterns'. The modification strength is proportional to the difference between two HR patterns,

$$\left(y_j^{(p,m-1)}(n)-y_j^{(q,m-1)}(n)\right),$$

	Testing Accuracy			
	k-NN	SOMP	MLP	SVM
Sonar	82.74%	86.60%	84.14%	88.00%
Wine	97.78%	98.33%	97.78%	98.30%
Ionosphere	85.17%	90.60%	88.32%	94.87%
Promoters	72.64%	86.55%	85.73%	89.36%

Table 4. The testing accuracy on real dataset.

and the difference between their postsynaptic responses (output vectors),

$$\left(y_i^{(p,m)}(n) - y_i^{(q,m)}(n)\right).$$

The increase of the strength of a synapse, $w_{ij}^m(n+1)$, is proportional to such differences on both sides of that synapse synchronously. To compare, a popular Hebbian learning form is

$$w_{ij}(n+1) \longleftarrow w_{ij}(n) + \eta y_i(n) x_j(n), \tag{13}$$

where $y_i(n)$ is the postsynaptic response and $x_j(n)$ is the presynaptic input pattern.
Another interesting form called the covariance hypothesis was introduced in Sejnowski (1977). According to this hypothesis, the learning applied to the synaptic weight w_{ij} is defined by

$$w_{ij}(n+1) \longleftarrow w_{ij}(n) + \eta\left(y_i(n) - \bar{y}(n)\right)\left(x_j(n) - \bar{x}(n)\right) \tag{14}$$

where $\bar{x}(n)$ and $\bar{y}(n)$ denote the time-averaged values of x_i and y_i, respectively. Comparing Equation (13) and Equation (14), the differences between them are the presynaptic and postsynaptic reference thresholds, which determine the sign of synaptic modification. In Equation (12), instead of the time-averaged references, the presynaptic signal and the postsynaptic signal use the other signals as the references.

4.3 Relation with support vector machine

The support vector machine employs the Mercer kernel to map patterns to a high dimensional space. Usually, the class information is not used in the design of the mapping function. The outcome of the mapping relies on the choice and the setting of the kernel function. The SOM perceptron is an adjustable mapping function to transform the patterns to a high dimensional space. It can be used as the Mercer kernel to map patterns to a space with highly separated representations.
In SVM, a multi-class classification task (polychotomy) can be decomposed into a set of simpler two-class classification tasks (dichotomies). Each dichotomy is implemented using one such machine independently. The outputs of these dichotomizes are reconstructed in classification. Advanced techniques have been developed for decomposition of poychotomy into dichotomies and reconstruction of their outputs. The SOM perceptron attempts to simultaneously divide the whole representation space for all classes. It uses the internal space of the layer perceptrons where each dichotomy (hyperplane) learns in a way dependent on each other. This learning will exhaust the hidden layer space and maximize the utility of all neurons to accomplish highly separated representations in that layer. Such representations have large margins and facilitate the operations of error correction.

4.4 Relation with mutual information learning

The proposed method is based on the maximization of the representations distances among different class representations and the minimization of the distances among the same class representations. There is a network Becker & Hinton (1992) with two modules. It has a different goal. It maximizes the mutual information, $I\left(\mathbf{y}^{(p,L)};\mathbf{y}^{(q,L)}\right)$, where $\mathbf{y}^{(p,L)}$ and $\mathbf{y}^{(q,L)}$ are the output vectors corresponding to the input patterns $\mathbf{x}^p$ and $\mathbf{x}^q$. This mutual information is defined as

$$I = \frac{1}{2}\log\frac{Var\left(\mathbf{y}^{(p,L)} + \mathbf{y}^{(q,L)}\right)}{Var\left(\mathbf{y}^{(p,L)} - \mathbf{y}^{(q,L)}\right)}$$

where *Var* is the variance over the responses of the training samples. This network is plotted in Figure 11(a). The goal of this network is to make the output $\mathbf{y}^{(p,L)}$ and $\mathbf{y}^{(q,L)}$ of the two modules to agree closely (i.e., to have a small value in the denominator $Var\left(\mathbf{y}^{(p,L)}-\mathbf{y}^{(q,L)}\right)$) for a closely related pair of input patterns $\mathbf{x}^p$ and $\mathbf{x}^q$. This goal is similar to, in some sense, the attraction energy for the same class patterns. In the same time, the two modules cannot just produce constant output that is unaffected by the input patterns, otherwise, they convey no information. The outputs of these two modules should vary as the inputs are varied. This constant situation is prevented by a large value in the numerator, $Var\left(\mathbf{y}^{(p,L)}+\mathbf{y}^{(q,L)}\right)$. Since there are two more classes in SOM perceptron, this kind prevention is not necessary. We will ignore the discussion on this numerator.
When we replace this two-module network with a single-module network as shown in Figure 11(b) and confine the output responses in a hypercube space. We then train this network to minimize the object function

$$I' = \frac{1}{2}\log Var\left(\mathbf{y}^{(p,L)}-\mathbf{y}^{(q,L)}\right)$$

for patterns in the same class that are closely related pairs. Conversely, we train the network to maximize this function for different class patterns. We could obtain similar results for the same class patterns as those obtained by the SOM perceptron.
We briefly explain the similarity of the two goals. The object function I' will weight frequent patterns. In our experiments all patterns have equal appearance (uniform probability distribution). Suppose that the mean value of the vector $\left(\mathbf{y}^{(p,m)}-\mathbf{y}^{(q,m)}\right)$ is zero. Assume each pattern in $\{\mathbf{x}^1,\mathbf{x}^2,\ldots,\mathbf{x}^P\}$ has its own representation, $\mathbf{y}^{(p,m)} \neq \mathbf{y}^{(q,m)}$ for $p \neq q$. Assume $\mathbf{y}^{(p,L)}-\mathbf{y}^{(q,L)}$ has equal probability of appearance, P^{-1}. Then

$$I' = \frac{1}{2}\{-2E^{rep}\} - \log P.$$

The information function, I', is similar to the repelling energy and the attraction energy. This shows that the two energies are agree with the mutual information to a certain extent. Note that the assumption on the equal probability of $\mathbf{y}^{(p,L)}-\mathbf{y}^{(q,L)}$ is not precise.

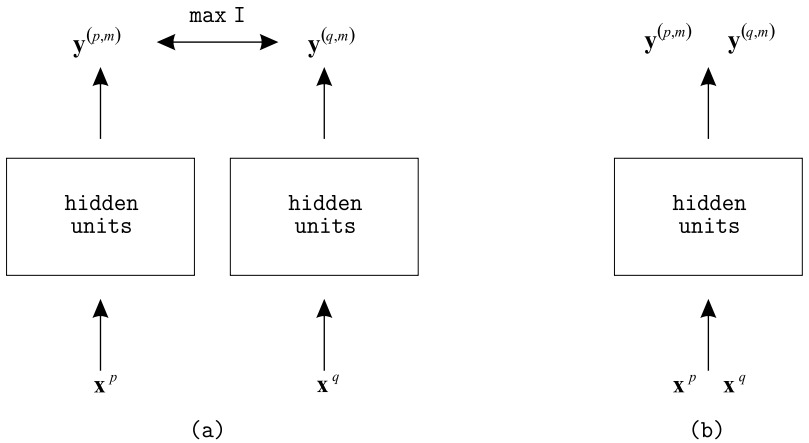


Fig. 11. (a) The two-module network. (b) The single-module network.

When we use

$$E^{rep} = -\frac{1}{2}\sum_{p=1}^{P}\sum_{q=1}^{P}\sum_{i=1}^{n_m}\left(y_i^{(p,m)} - y_i^{(q,m)}\right)^2,$$

the SIR method tends to maximize (or minimize) the variance of each neuron's output difference, $Var\left(y_i^{(p,m)} - y_i^{(q,m)}\right)$, evenly for all pairs of different class patterns (or same class patterns). All neurons will be devoted to these class patterns. All neurons are sensitive to these patterns only. Any unknown pattern will be included in one of these patterns' representations. In other words, these representations exhaust the pattern space.

4.4.1 Acknowledgement

This work is supported by National Science Council.

5. References

Ackley, D., Hinton, G. & Sejnowski, T. (1985). A learning algorithm for boltzmann machines, *Cognitive Science* 9: 147–169.

Becker, S. & Hinton, G. (1992). A self-organizing neural network that discovers surfaces in random-dot stereograms, *Nature* 355: 161–163.

Boser, B., Guyon, I. & Vapnik, V. (1992). A training algorithm for optimal margin classifiers, *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152.

Chang, C.-C. & Lin, C.-J. (2001). Libsvm : a library for support vector machines, Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Daugman, J. (1980). Two-dimensional spectral analysis of cortical receptive field profiles, *Vision research* 20: 847–856.

Dobbins, A., Zucker, S. & Cynader, M. (1987). Endstopped neurons in the visual cortex as a substrate for calculating curvature, *Nature* 329: 438–441.

Elman, J. (1991). Distributed representations, simple recurrent networks, and grammatical structure, *Machine Learning* 7: 195–225.

Huang, J.-C. & Liou, C.-Y. (2010). Solution space of perceptron, *International conference on Artificial Neural Networks (ICANN)*.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps, *Biological Cybernetics* 43: 59–69.

Kosko, B. (1988). Bidirectional associative memories, *IEEE Transactions on Systems, Man, and Cybernetics* 18: 49–60.

Liou, C.-Y. (2000a). NN Chapter 3, http://red.csie.ntu.edu.tw/NN/Classinfo/code/COPLANARITY.M.

Liou, C.-Y. (2000b). NN Chapter 4, http://red.csie.ntu.edu.tw/NN/Demo/hamming_tree.html.

Liou, C.-Y. (2000c). NN Chapter 5, http://red.csie.ntu.edu.tw/NN/Classinfo/code/SIR_METHOD.M.

Liou, C.-Y., Chen, H.-T. & Huang, J.-C. (2000). Separation of internal representations of the hidden layer, *Proceedings of the 2000 International Computer Symposium*, pp. 26–34.

Liou, C.-Y. & Cheng, W.-C. (2008). Resolving hidden representations, *Lecture Notes in Computer Science*, Vol. 4985, Part II, Springer, Heidelberg, pp. 254–263.

Liou, C.-Y., Huang, J.-C. & Kuo, Y.-T. (2005). Geometrical perspective on learning behavior, *Journal of Information Science and Engineering* 21: 721–732.

Liou, C.-Y. & Lin, S.-L. (1989). The other variant boltzmann machine, *Proceedings of the International Joint Conference on Neural Networks*, Washington, D. C., USA, pp. 449–454.

Liou, C.-Y. & Yu, W.-J. (1994). Initializing the weights in multilayer network with quadratic sigmoid function, *Proceedings of the International Conference on Neural Information Processing*, pp. 1387–1392.

Liou, C.-Y. & Yu, W.-J. (1995). Ambiguous binary representation in multilayer neural network, *Proceedings of International Conference on Neural Networks*, Vol. 1, pp. 379–384.

Mirchandini, G. & Cao, W. (1989). On hidden nodes in neural nets, *IEEE Transactions on Circuits and Systems* 36: 661–664.

Penfield, W. & Rasmussen, T. (1950). *The cerebral cortex of man*, The Macmillan Company, New York.

Salakhutdinov, R., Mnih, A. & Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering, *Proceedings of the 24th international conference on Machine learning*, pp. 791–798.

Sejnowski, T. (1977). Storing covariance with nonlinearly interacting neurons, *Journal of mathematical biology* 4: 303–321.

Edited by Josphat Igadwa Mwasiagi

Kohonen Self Organizing Maps (SOM) has found application in practical all fields, especially those which tend to handle high dimensional data. SOM can be used for the clustering of genes in the medical field, the study of multi-media and web based contents and in the transportation industry, just to name a few. Apart from the aforementioned areas this book also covers the study of complex data found in meteorological and remotely sensed images acquired using satellite sensing. Data management and envelopment analysis has also been covered. The application of SOM in mechanical and manufacturing engineering forms another important area of this book. The final section of this book, addresses the design and application of novel variants of SOM algorithms.

ISBN 978-953-51-4526-4

Photo by cornishman / iStock

IntechOpen