IntechOpen

New Approaches to Characterization and Recognition of Faces

Edited by Peter Corcoran

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

NEW APPROACHES TO CHARACTERIZATION AND RECOGNITION OF FACES

Edited by **Peter M. Corcoran**

New Approaches to Characterization and Recognition of Faces
http://dx.doi.org/10.5772/994
Edited by Peter Corcoran

Contributors

Khalil Khattab, Julien Dubois, Johel Miteran, Philip Brunet, Kiyomi Nakamura, Hironobu Takano, Jean-Paul Kouma, Peter M Corcoran, Claudia Iancu, Shefa Dawwd, Basil Mahmood, Jacek Naruniec, Naser Zaeri, Salvador Ayala-Raggi, Gabriel Costache, Sathish Mangapuram, Alexandu Drimbarean, Petronel Bigioi, Eyad Elyan, Daniel C Doolan, Alaa Eleyan, Huseyin Ozkaramanli, Hasan Demirel, Carlos Eduardo Thomaz, Edson Kitani, Gilson Giraldi, Emilio Hernandez, Guido Gainotti

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

New Approaches to Characterization and Recognition of Faces
Edited by Peter Corcoran

p. cm.

ISBN 978-953-307-515-0

eBook (PDF) ISBN 978-953-51-4471-7

Meet the editor

Dr. Peter M. Corcoran received the BAI (Electronic Engineering) and BA (Math's) degrees from Trinity College Dublin in 1984. He continued his studies at TCD and was awarded a Ph.D. in Electronic Engineering for research work in the field of Dielectric Liquids. In 1986 he was appointed to a lectureship in Electronic Engineering at the National University of Ireland Galway. He is currently vice-dean in the College of Engineering & Informatics at NUI, Galway. His research interests include embedded systems applications, home networking, digital imaging, pattern recognition, face & fingerprint biometrics, smart grid and wired and wireless networking technologies. He was technical and conference chair of the IEEE International Conference on Consumer Electronics (ICCE) in 2010 and 2011 respectively. He is also editor of the IEEE Consumer Electronics Society newsletter and Editor-in-Chief of the newly launched (2012) IEEE Consumer Electronics Magazine. Peter is also a co-founder of FotoNation, a leading OEM supplier of in-camera image processing and analysis software, including embedded solutions for red-eye detection & correction and face tracking and recognition. Fotonation is now part of the Imaging and Optics division of Tessera Technologies. He is author of more than 200 technical publications and co-inventor on more than 100 granted US patents. He is a Fellow of the IEEE.

Contents

Preface

As a baby one of our earliest stimuli is that of human faces. We rapidly learn to identify, characterize and eventually distinguish those who are near and dear to us. This skill stays with us throughout our lives.

As humans, face recognition is an ability we accept as commonplace. It is only when we attempt to duplicate this skill in a computing system that we begin to realize the complexity of the underlying problem. Understandably, there are a multitude of differing approaches to solving this complex problem. And while much progress has been made many challenges remain.

This book is arranged around a number of clustered themes covering different aspects of face recognition. The first section presents an architecture for face recognition based on Hidden Markov Models and is followed by an article on coding methods for image retrieval in large databases. The second section of this book is devoted to 3 articles on 3D methods of face recognition and is followed by a section with 5 articles covering various aspects and techniques of face recognition in video sequences and in real-time. This is followed by a section devoted to characterization and the detection of features in faces. The complexity of facial features and expressions is often simplified or disregarded by face recognition methodologies. Finally an article on the human perception of faces and how different neurological or psychological disorders can affect this.

I hope that you find these articles interesting, and that you learn from them and perhaps even adopt some of these methods for use in your own research activities.

Sincerely,

Peter M. Corcoran
Vice-Dean,
College of Engineering & Informatics,
National University of Ireland Galway (NUIG),
Galway,
Ireland

Part 1

Architectures and Coding Techniques

1

Automatic Face Recognition System for Hidden Markov Model Techniques

Peter M. Corcoran and Claudia Iancu
College of Engineering & Informatics,
National University of Ireland Galway,
Ireland

1. Introduction

Hidden Markov Models (HMMs) are a class of statistical models used to characterize the observable properties of a signal. HMMs consist of two interrelated processes: (i) an underlying, unobservable Markov chain with a finite number of states governed by a state transition probability matrix and an initial state probability distribution, and (ii) a set of observations, defined by the observation density functions associated with each state.

In this chapter we begin by describing the generalized architecture of an automatic face recognition (AFR) system. Then the role of each functional block within this architecture is discussed. A detailed description of the methods we used to solve the role of each block is given with particular emphasis on how our HMM functions. A core element of this chapter is the practical realization of our face recognition algorithm, derived from EHMM techniques. Experimental results are provided illustrating optimal data and model configurations. This background information should prove helpful to other researchers who wish to explore the potential of HMM based approaches to 2D face and object recognition.

2. Face recognition systems

In this section we outline the basic architecture of a face recognition system based on Gonzalez's image analysis system [Gonzalez & Woods 1992] and Costache's face recognition system [Costache 2007]. At a top-level this is represented by the functional blocks shown in Figure 1.

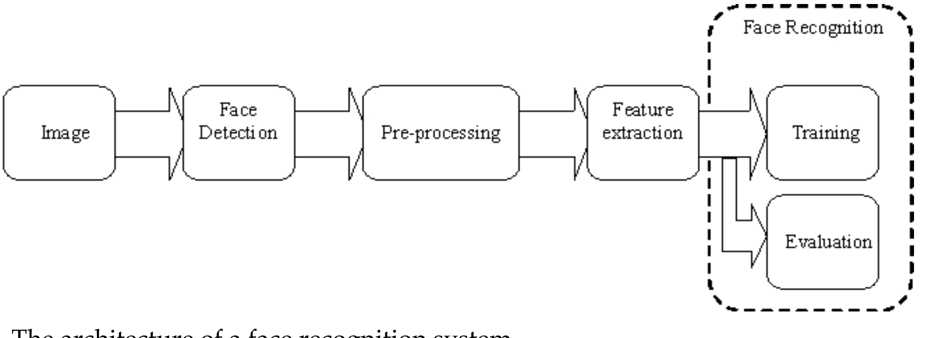


Fig. 1. The architecture of a face recognition system

1. Face detection and cropping block: this is the first stage of any face recognition system and the key difference between a semi-automatic and a fully automatic face recognizer. In order to make the recognition system fully automatic, the detection and extraction of faces from an image should also be automatic. Face detection also represents a very important step before face recognition, because the accuracy of the recognition process is a direct function of the accuracy of the detection process [Rentzeperis *et. al.* 2006, Corcoran *et. al.* 2006].
2. Pre-processing block: the face image can be treated with a series of pre-processing techniques to minimize the effect of factors that can adversely influence the face recognition algorithm. The most critical of these are *facial pose* and *illumination*. A discussion on these factors and their significance w.r.t. HMM techniques is given in Section 3.
3. Feature extraction block: in this step the features used in the recognition phase are computed. These features vary depending on the automatic face recognition system used. For example, the first and most simplistic features used in face recognition were the geometrical relations and distances between important points in a face, and the recognition 'algorithm' matched these distances [Chellappa *et. al.* 1992]; the most widely used features in face recognition are KL or eigenfaces, and the standard recognition 'algorithm' uses either the Euclidian or Mahalanobis distance [Chellappa *et. al.* 1992, 1995] to match features. Our features and the extraction method used are described in Section 4.
4. Face recognition block: this consists of 2 separate stages: a *training process*, where the algorithm is fed samples of the subjects to be learned and a distinct model for each subject is determined; and an *evaluation process* where a model of a newly acquired test subject is compared against all exisiting models in the database and the most closely corresponding model is determined. If these are sufficiently close a recognition event is triggered.

3. Face detection and cropping

As mentioned in the previous section, face detection is one of the most important steps in a face recognition system and differentiates between semi-automatic and fully automatic face recognizer. The goal of an automatic face detector is to search for human faces in a still image and, if found, to accurately return their locations. In order to make the detection fully automatic the system has to work without input from the user. Many attempts to solve the problem of face detection exist in the literature beginning with the basic approach of [Kanade 1977] and culminating with the method of [Viola & Jones 2000, 2001]. Comprehensive surveys of face detection techniques can be found in [Yang *et. al.* 2002] and [Costache 2007]. In this section we underline the main challenges an automatic face detector has to tackle, and we briefly describe the face detector used in our experiments.
Face detection methods were classified by [Yang *et. al.* 2002] into four principle categories: (i) knowledge-based, (ii) feature invariant, (iii) template matching and (iv) appearance-based methods. According to [Gonzalez & Woods 1992], the main disadvantage presented by the majority of these methods is the time required to detect all the faces in an image. State-of-the-art face detection methods provide real-time solutions. The best known of these methods, and the gold standard for face detection was originally proposed by [Viola & Jones 2001]. The original algorithm was, according to its authors, 15 times faster than any previous approach. The algorithm has been well proved in recent years as being one of the fastest and most accurate face detection algorithms reported and is presently the gold standard against which other face detection techniques are benchmarked. For these reasons we adopted it to implement our face detection subsystem.

Our implementation of the Viola & Jones detection algorithm is provided in the Intel digital image processing C++ library [OpenCV 2006]. This can be used both for face detection and subsequent cropping of confirmed facial images. The OpenCV face detector has been pre-trained using a very comprehensive database of face/non-face examples and is widely used in the literature.

4. Pre-processing techniques

Automatic face detection is influenced by a number of key factors [Costache 2007]: facial *orientation* or *pose*: the appearance of the face varies due to relative camera-face pose, between full frontal images and side-profile images; *in-situ occlusions* such as facial hair (e.g. beard, moustache), eye-glasses and make-up; facial *expressions* can significantly influence the appearance of a face image; *overlapping occlusions* where faces are partially occluded by other faces present in the picture or by objects such as hats, or fans; *conditions of image acquisition* where the quality of the picture, camera characteristics and in particular the *illumination conditions* can strongly influence the appearance of a face.

For our system to perform better in the recognition stage, we apply a set of pre-processing techniques: the first step in pre-processing is to bring all images into the same color space and to normalize the size of face regions. This normalization process is critical to improving the final face recognition rate and we will later present some experimental results for our HMM-specific AFR.

4.1 Color to grayscale conversion

In most face recognition applications the images are single or multiple views of 2D intensity data [Zhao *et. al.* 2003], and many databases built for face recognition applications are available as grayscale images. From the four databases used in our experiments, 3 contained grayscale images (BioID, Achermann, UMIST) and one contained RGB images (FERET). Practical images will, naturally, be acquired in color as modern image acquisition systems are practically all color and so we need to convert from color to grayscale, or intensity images of the selected face regions. In practice the intensity data may be available from the imaging system – many camera system employ YCC data internally and the Y component can be utilized directly. In other cases we may need to perform an explicit conversion of RGB data. Here a set of red, green and blue integer values characterize an image pixel. The effective luminance, Y of each pixel is calculated with the following formula [Pratt 1991]:

$$Y = 0.3 \times \text{Red} + 0.59 \times \text{Green} + 0.11 \times \text{Blue} \tag{1}$$

4.2 Image resizing

For a HMM-based face recognition system having a consistently sized face region is particularly important because the HMM requires regional analysis of the face with a scanning window of fixed size. A straightforward approach is to resize all determined face regions to a common size. To facilitate more efficient computation we seek the smallest sized face region possible without impacting the overall system recogntion rate. Some empirical data will be presented later to illustrate how different factors, including the size of normalized face regions, affect recognition rate.

There are many techniques that can be used to enlarge or reduce the size of an image. These methods generally realize a trade-off between speed and the degree to which they reduce the occurrence of visual artifacts in the resulting image. The most commonly used resize method is called bicubic interpolation and has the advantage that the interpolated image is smoother than images obtained using simpler interpolation techniques and has fewer artifacts [Lehmann *et. al.* 1999]. In our work we have used bicubic spline interpolation using bicubic polynomials. More details of how to calculate bicubic spline interpolation functions can be found in [Hummel 1977].

4.3 Illumination normalization

One of the most important factors that influence the recognition rate of a system is illumination variation. In was shown in [Adini *et al.* 1997, Gokemen *et al.* 2007] that variations in illumination can be more relevant than variations between individual characteristics. Such variations can induce an AFR system to decide that two different individuals with the same illumination characteristics are more similar than two instances of the same individual taken in different lighting conditions. Thus normalizing illumination conditions across detected face regions is crucial to obtaining accurate, reliable and repeatable results from an AFR. One approach suitable for face models which combine both facial geometry and facial texture such as active appearance models (AAM) is described by [Ionita 2008]. However as HMM techniques do not explicitly rely on facial geometry or textures it is not possible to integrate the illumination normalization within the structure of the model itself. Instead we must rely on a discrete illumination normalization process. Fortunately most AFR systems employ a similar prefiltering stage and we can draw on a wide range of techniques from the literature.

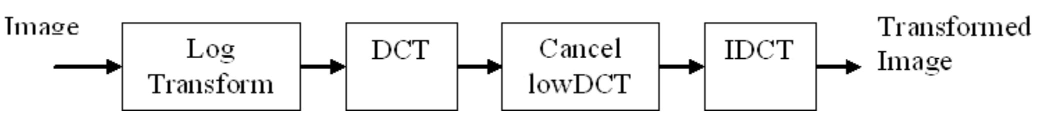


Fig. 2. Block scheme of logDCT algorithm

Algorithms used for performing the normalization vary from a simple histogram equalization (HE) to more complex techniques such as albedo maps [Smith & Hancock 2005] and contrast limited adaptive histogram equalization (CLAHE) [Zuiderveld 1994, Pizer *et al* 2006, Corcoran *et al* 2006]. These algorithms perform well when the variations in illumination are small but there is no commonly adopted method for illumination normalization in images which performs well for every type of illumination. Some tests have been conducted to determine the robustness of face recognition algorithms to changes in lighting [Phillips *et al* 2000, O'Toole *et al* 2007]. Also, numerous illumination normalization techniques have been developed. Some of the more widely used of these - *histogram equalization*, *histogram specification* and *logarithm transformation* - have been compared in [Du *et al* 2005] with more recently proposed methods, *gamma intensity* correction and *self-quotient image*. The results are interesting: both HE and logarithmic transform improved recognition rates over face regions that were not normalized, compared favorably to the other techniques.

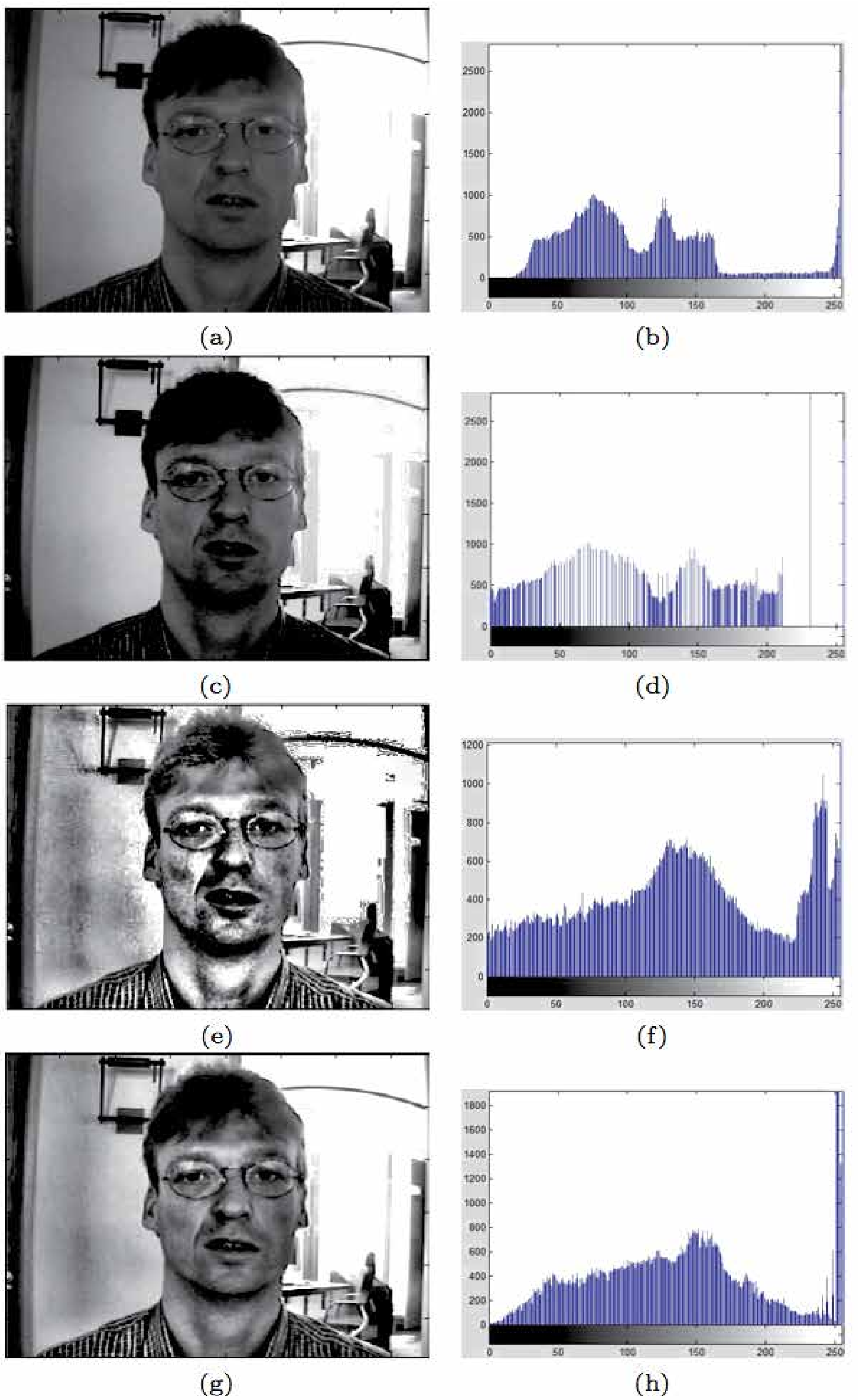


Fig. 3. Examples of illumination normalization techniques – details in the text.

To tackle the problem of illumination variations we implemented the following three illumination normalization algorithms: (i) *histogram equalization* (HE) based on [Gonzalez & Woods 1992], (ii) *contrast limited adaptive histogram equalization* (CLAHE) based on [Zuiderveld 1994], and (iii) the relative new method of DCT in the logarithm domain - *logDCT* based on [Chen *et al* 2006]. In figure 3 above we show some examples of a face image processed by different normalization algorithms: (a) shows the unprocessed image with (b) the original luminance histrogram; (c) is the same image normalized with simple HE and (d) the effect of HE on the image histogram; (e) is the image with *adaptive* HE applied and (f) the effect of AHE on the histogram, in particular note the high frequency blow-up of the histogram; finally (g) shows how CLAHE eliminates the high-frequency artifacts of AHE and (h) reduces the high-frequency blow-up when compared with (f).

5. Feature extraction

Feature extraction for both 1D and 2D HMMs was originally described by [Samaria 1994]. His method was subsequently adopted in the majority of HMM-based face recognition papers. This feature extraction technique is based on scanning the image with a fixed-size window from left-to-right and top-to-bottom. A window of dimensions $h \times w$ pixels begins scanning each extracted face region from the left top corner sub-dividing the image into a set number of $h \times w$ sized blocks.
On each of these blocks a transformation is applied to extract the characterizing features which represent the observation vector for that particular region. Then the scanning window moves towards right with a step-size of n pixels allowing an overlap of o pixels, where o = w − n. Again features are extracted from the new block. The process continues until the scanning window reaches the right margin of the image. When the scanning window reaches the right margin for the first row of scanned blocks, it moves back to the left margin and down with m pixels allowing an overlap of v pixels vertically. The horizontal scanning process is resumed and a second row of blocks results, and from each of these blocks an observation vector is extracted. The scanning process and extraction of blocks is depicted in Figure 4.

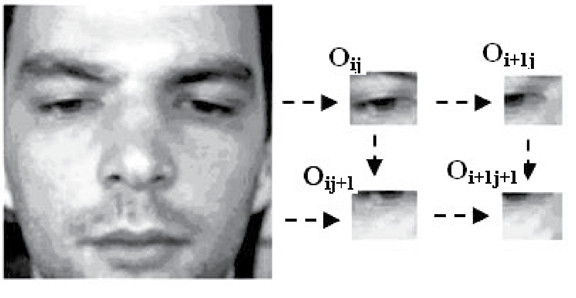


Fig. 4. Blocks extraction from a face image

In our work we have used two types of features to describe the images: 2D DCT coefficients and Daubechies wavelets.

5.1 Overview of features used with HMM in face recognition

The first features used in face recognition performed with HMM were pixel intensities [Samaria & Fallside 1993, Samaria 1994, Samaria & Harter 1994]. The recognition rates obtained by Samaria using pixel intensities with a P2D-HMM were up to 94.5% on the ORL database. However the use of pixel intensities as features has some disadvantages [Nefian & Hayes 1999]: firstly they cannot be regarded as robust features since: (i) the intensity of a pixel is very sensitive to the presence of noise in the image or to illumination changes; (ii) the use of all the pixels in the image is computationally complex and time consuming; and (iii) using all image pixels does not eliminate any redundant information and is thus a very inefficient form of feature. Another example of features used with EHMM for face recognition are KLT features used by [Nefian & Hayes 1998, Nefian & Hayes 2000] with recognition rates of up to 98% on ORL database. The main advantage of using KLT features instead of pixel intensities is their capacity to reduce redundant information in an image. The disadvantage is their dependence of the database of training images from which they are derived [Costache 2009].

The most widely used features for HMM in face recognition are 2D-DCT coefficients. These DCT coefficients combine excellent decorrelation properties with energy compaction. Indeed, the more correlated the image is, the more energy compaction increases. Thus a relatively small number of DCT coefficients contain the majority of information encapsulated in an image. A second advantage is the speed with which they can be computed since the basis vectors are independent of the database and are often pre-computed and stored in an imaging device as part of the JPEG image compression standard. Recognition rates obtained when using 2D DCT with HMM can achieve 100% success on smaller databases such as ORL. In our research we also introduce the use of Daubechies wavelets. Apart from the work of [Le & Li 2004] wavelets have not been previously used with HMMs for face recognition applications.

6. Face recogntion

In the earlier sections of this chapter we have described the main pre-filtering blocks for our AFR system. We nest focus on the actual HMM itself and the various operational processes required to implement the training and recognition phases of our AFR.

6.1 Background to embedded hidden markov models in face recognition

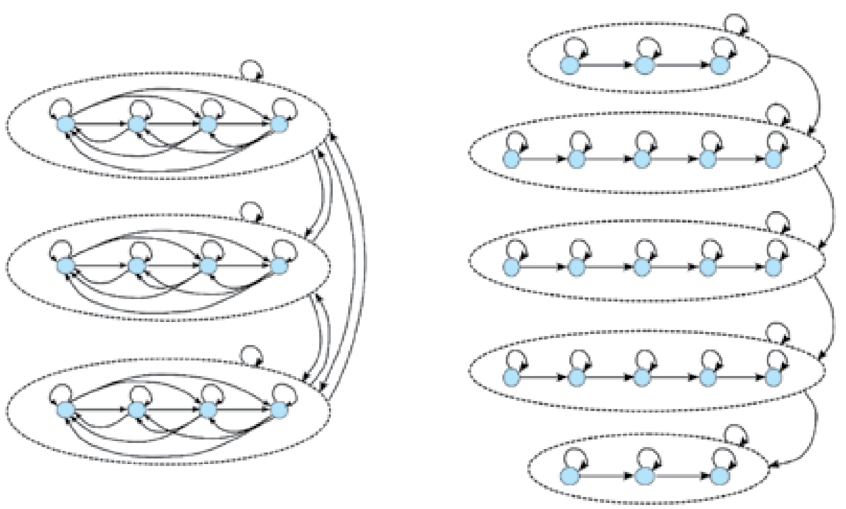
After their introduction in the late 60's by [Baum *et al* 1966, 1970] and the more detailed description in the late 80's by [Raibner & Juang 1986, Rabiner 1989] HMMs have been widely used in speech recognition applications. In this field of application very high recognition rates are obtained due to the specific capacity of HMM to cope with variations in the timing and duration human speech patterns [Juang and Rabiner 2005]. HMMs have also been used successfully in other applications such as OCR and handwriting recognition. Thus it was no surprise that researchers began to consider their use for problems such as face recognition where adaptability of HMMs might offer solutions to some of the underlying problems of accurately recognizing a 2D face region.

Note that the application of HMM techniques to the face recognition problem implies the use of an inherently 1D method of pattern matching to solve an inherently 2D problem. So why did researchers think this might work? Well, as the most significant facial features of a frontal face image occur in a natural order, from top to bottom, and this sequence is

immutable, even if the face is moderately rotated. The first attempts to use HMMs for face recognition and detection were made by [Samaria & Fallside 1993, Samaria & Harter 1994] who used a left-to-right HMM and divided the face in a fixed succession of regions (observantion states) such as eyes, nose, & mouth. This early work by Samaria was essentially 1D in scope and the first attempt to implement a more appropriate 2D models was *Pseudo 2D HMM*, introduced by [Kuo & Agazzi 1994] for character recognition, subsequently adapted by [Samaria 1994] for the face recognition problem. The idea was later taken forward and improved by [Nefian & Hayes 1999, 2000]. These researchers changed the name to *Embedded HMM* (EHMM).
There have been several alternative 2D versions of HMM used for face recognition in the literature. However EHMM has become the standard method employed by researchers working in the field of HMM face recognition. As a result this algorithm has been implemented in the Intel digital image processing C++ library [OpenCV 2006] which was also employed to implement our face detector, described in section 2 above.

6.2 An overview of EHMMs

The embedded HMM is a generalization of the classic HMM, where each state in the one dimensional HMM is itself an HMM. This enables a generalization of the 1D HMM techniques to a second dimension while simplifying the dependencies and transitions between states. Thus, an embedded HMM consists of a set of super states each of which envelopes a set of embedded states. The super states model the two dimensional data in a first dimension, while the embedded HMMs model the data in the other dimension.
The structure of an EHMM with 3 superstates and 4 embedded states is shown in Figure 5(a). This EHMM is unrestricted, meaning that transitions between all the states in the embedded HMMs and between all the superstates are allowed.



(a): An unrestricted EHMM (b): Restricted EHMM for face recognition

Fig. 5. EHMM for face recognition

The elements of an embedded HMM are:

- A set of N_0 superstates $S_0 = S_{0,1}$, $1 \le i \le N_0$

- The initial probabilities of the super states $\Pi_0 = \pi 0,i$ where $\pi 0,i$ is the probability of being in superstate i at time zero.
- The transition probability matrix $A_0 = a_{0,ij}$, where $a_{0,ij}$ is the probability of transitioning from super state i to superstate j.
- The parameters of the embedded HMM for the superstate k, $1 \le k \le N_0$ and where $\Lambda^k = (\Pi_1^k, A_1^k, B^k)$ which includes: (i) the number of embedded states in the k^{th} super state, N_1^k, and the set of embedded states, $S_1^k = S_{1,i}^k$ with $1 \le i \le N_1^k$; (ii) the initial state distribution, $\Pi_1^k = \pi_{1,i}^k$, where $\pi_{1,i}^k$ is the probability of being in state i of the superstate k at time zero; (iii) the state transition probability matrix $A_1^k = a_{1,ij}^k$, where $a_{1,ij}^k$ is the transition probability from state i to state j; (iv) the probability distribution matrix of the observations, B^k; these observations are characterized by a set of continuous probability density functions, considered finite *Gaussian mixtures* of the form:

$$b_i^k(O_{t0,t1}) = \sum_{m=1}^{M_i^k} c_{im}^k N(O_{t0,t1}, \mu_{im}^k, U_{im}^k) \tag{2}$$

  where c_{im}^k is the mixture coefficient for the m^{th} mixture in state i of super state k, and $N(O_{t0,t1}, \mu_{im}^k, U_{im}^k)$ is a Gaussian density with a mean vector μ_{im}^k and covariance matrix U_{im}^k.

In shorthand notation, an embedded HMM is defined as the triplet $\lambda = (\Pi_0, A_0, \Lambda)$ where $\Lambda = \Lambda^1, \Lambda^2, ..., \Lambda^{N_0}$. This model is appropriate for facial images since it exploits an important characteristic of these: frontal faces preserve the structure of "superstates" from top to bottom, and also the left-to-right structure of 'embedded states' within each "superstate" [Nefian & Hayes 1999, 2000]. An example of the state structure of the face model and the non-zero transition probabilities of the embedded HMM are shown in Figure 5(b). The configuration presented is 5 super states, each with 3, 6, 6, 6, 3 states respectively. Each state in the overall top-to-bottom HMM is assigned to a left-to-right 1D HMM. In this case transitions are allowed only from left-to-right or self-transitions and only between *consecutive states* both for the embedded HMMs within each superstate, and for the main superstates of the top-level HMM as well.

6.3 The training process for an EHMMs

The training of HMM, as shown by [Rabiner 1989] is accomplished using the Baum-Welch algorithm. While EHMM exhibits a more complex structure than the simple 1D HMM, the training algorithm follows the same steps. The main difference in training is the use of a doubly embedded Viterbi for segmentation. The training structure is depicted in Figure 6, and the role of each block is described next:

Step 1. PrototypeHMM: the first step is defining the prototype EHMM: *parameters:* N_i^k the numbers of states, N_0 the number of superstates , *K* the number of Gaussians used to model the probability density for the observation vectors in each state of an embedded HMM; *conditions:* which transitions are allowed ($a_{1,ij}^k$ > 0) and which are not ($a_{1,ij}^k$ =0); in our left-to-right HMM the only transitions allowed are self-transitions and transitions to the next state, so the probability of transition to previous states is 0. For a numerical example we choose $N_0 = 5$, $N_1^k = 3,6,6,6,3$ where *k* = 1, 2, ..., 5, and *K* = 3.

Step 2. **Uniform segmentation:** the image is uniformly segmented. Firstly the observation vectors extracted from the entire image are uniformly divided into $N_0 = 5$ vertical super states, or image strips, for the overall top-to-bottom HMM. Next the data corresponding to each of these vertical super states is now horizontally segmented from left to right into N_i^k uniform states. For a 128 × 128 pixel facial region with scanning window 12 × 12 with 8 pixels overlap we obtain 30 observation vectors per scanning row both horizontally and vertically, thus 900 observation vectors in total. In a uniform segmentation, the observation vectors are first divided between $N_0 = 5$ superstates: 30 observation vectors per row/5 superstates, so 6 observation vectors per row in each superstate ⇒ 6 × 30 = 180 observation vectors per superstate. Then horizontally these 180 observation vectors are uniformly divided in states as follows: for the superstates 1 and 5 with only 3 states: there will be 60 observation vectors per state; for superstates 2, 3, 4 with 6 states each: there will be 30 observation vectors per state.

Step 3. **Parameter initialization:** after segmentation, the initial estimates of the model parameters are obtained using the concept of counting event occurrences for the initial probabilities of the states and the transition probabilities. In order to compute the observation probabilities, for each state of the embedded HMMs a K-means clustering algorithm, where *K* is the number of Gaussians per state, is applied. All the observation vectors extracted from each state are used to obtain a corresponding mixture of Gaussians describing the observation probability density function. From these initial values we then begin to iterate. In the example given above the initial probabilities of the states in each superstate are determined from the system constraints as follows: first state in each embedded HMM has initial probability equal to 1.0, all the other states have initial probability of zero. Transition probabilities are then obtained by counting transition occurrences. For example in the first state of the first superstate: there are 60 observation vectors distributed across 6 horizontal rows of scanning implying 6 possibilities of transition from state 1 into state 2: probability of transition from state 1 into state 1 is $P_{1,1} = 54/60$; probability of transition from state 1 into state 2 is $P_{1,2} = 6/60$; transition probabilities for the other states can be calculated in the same way.

Step 4. **Embedded Viterbi segmentation:** in the first step of the iteration, a doubly embedded Viterbi algorithm replaces the uniform segmentation. With the new segmentation and again counting event occurrences, a set of new values for initial and transition probabilities are found. This process is described in detail in section 5.4 below.

Step 5. **Segmental K-means:** according to the new segmentation performed at step 4, another K-means is applied to the current set of observation vectors corresponding to each new state, and new observation probability density functions are computed. On the next iteration, these new values are introduced into the doubly embedded Viterbi and a new segmentation is initiated.

Step 6. **Convergence:** Steps 4 and 5 are repeated until the difference on consecutive iterations is below a set threshold. If convergence is not achieved after a certain number of iterations the training is considered to have failed for the current input face region. Typically we have set the convergence threshold at 0.01 and the maximum number of iterations at 40. Once convergence is achieved, further iterations are stopped and the EHMM is output and stored in a reference database.

6.4 The decoding process for an EHMM (Doubly embedded Viterbi)

In the description of the training process above we have seen that step 4 consists in the re-segmentation of the states in the 1D HMMs and of the superstates in the overall HMM. Re-segmentation means finding the most probable sequence of states given a certain sequence of observation vectors and we can solve this problem by applying the Viterbi algorithm. We can easily apply Viterbi algorithm in the embedded 1D HMMs for which we have determined all the probabilities at step 3 above. However for the overall HMM after step 3 we only have the initial and transition probabilities, without the observations probabilities.

In order to solve this problem a method based on the Viterbi algorithm known as *double embedded Viterbi* was developed [Kuo & Agazzi 1994]. It involves applying the Viterbi algorithm to both the embedded HMMs and to the global, or top-level HMM, hence the name. A detailed description may be found in [Nefian 1999]. As the algorithm is mathematically complex and the formulas are challenging to understand and even more so to implement. For this reason we will next provide a detailed practical (as opposed to theoretical) description of our step by step implementation of the algorithm. The underlying concept is illustrated in Figure 6 and comprises the following steps:

Step 1. After the parameters initialization step No. 3 of the previous section we have: initial probabilities, transition probabilities and observation probabilities for each embedded HMM, and initial and transition probabilities for the top-level HMM. In the first step of the double Viterbi, each scanned row of observation vectors $O_0^i = v_j^i$ with $1 \le i \le (H-v)/(h-v)$ and $1 \le j \le (W-o)/(w-o)$ within each of the embedded 1D HMMs has the conventional Viterbi algorithm applied. After this step the optimal state distribution is obtained for each row of observation vectors based on the relevant 1D HMM $(Q^i \big| Q_0^i, \Lambda_k)$ and also the probability of each row of observation for the top-level HMM, or superstate $P(O_0^i, Q^i \big| \Lambda^k)$, where $1 \le i \le (H-v)/(h-v)$ and $1 \le k \le N_0$.

Step 2. After the first application of the Viterbi algorithm we have: initial and transition probabilities for the superstates as determined at step 1 of the training algorithm described above, and the observations probability distributions for the top-level HMM, that is: the probabilities of each horizontal row of observations given each

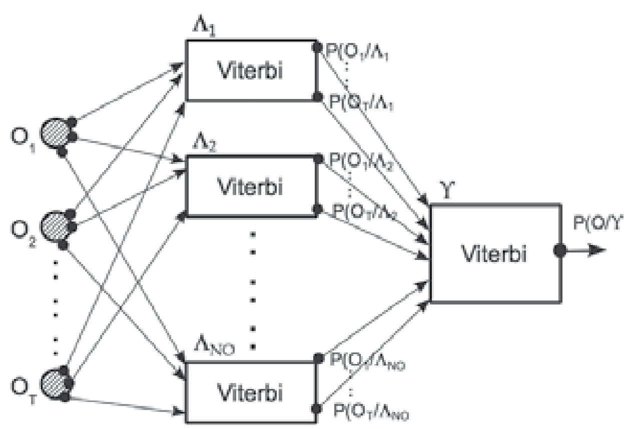


Fig. 6. Doubly embedded Viterbi

superstate. Now Viterbi is applied on the top-level HMM and the optimal sequence of superstates is obtained given the sequence of rows of observation vectors (vertical re-segmentation) and the probability of the entire sequence of observations (which characterizes the entire image) given the EHMM model created. This probability is compared at each iteration with the same probability obtained on the previous iteration (step 6 in Section 5.3).

Step 3. Vertical re-segmentation means reassignment of each row of observation vectors to the corresponding superstate (embedded 1D HMM). After we determine to which embedded 1D HMM each row of observation vectors appertains. Then using the findings at the first step of the *double embedded* Viterbi algorithm horizontal re-segmentation can be achieved.

In Figure 7 below we give an example of states and superstates segmentation for a given face image. Each color represents a different state in the superstates. As one can see, the 5 superstates found in this image are: forehead region, ends right above the eyebrows and is divided into 3 states, eye region, starts just above eyebrows and ends just after the pupil, is divided into 6 states, the nose region, starts just after the pupil and ends just before the nostrils, is divided into 6 states, the nostrils region, starts just under the nose region and ends at the middle between mouth and tip of the nose, is divided into 6 states, and finally the mouth region starts after the nostrils region and ends at the bottom of the image, is divided into 3 states. Also from the image we can see that the rows of observation vectors inside one superstate are distributed unevenly between states.

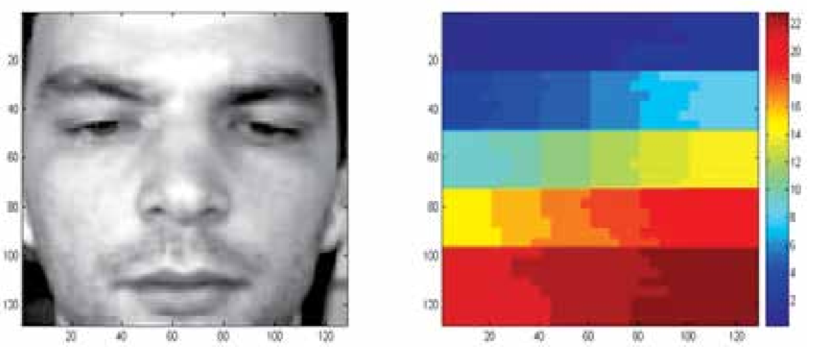


Fig. 7. State distribution after applying doubly embedded Viterbi

6.5 The evaluation process for an EHMM

In the training process described previously we have shown how an EHMM model is built for a subject in the database. After building separate EHMMs for all subjects in the database we can move to the recognition step where the likelihood of a face test image is evaluated against all models. The evaluation process comprises of the following steps:

Step 1. first the face image is scanned with a rectangular window from left-to-right and top-to-bottom and the observation vectors are extracted.

Step 2. then the sequence of observation vectors is introduced in each EHMM model and its corresponding likelihood is computed. Theoretically the probability of a sequence of observation vectors, given a model, is found using the forward-backward *evaluation algorithm*. However, in practice it was argued [Raibner 1989, Kuo & Agazzi 1994] that the Viterbi algorithm can successfully replace the evaluation algorithm. For our EHMM we use a doubly embedded Viterbi algorithm. At the end of this step we have the probabilities of the test image to match each of the EHMM models in the recognition database.

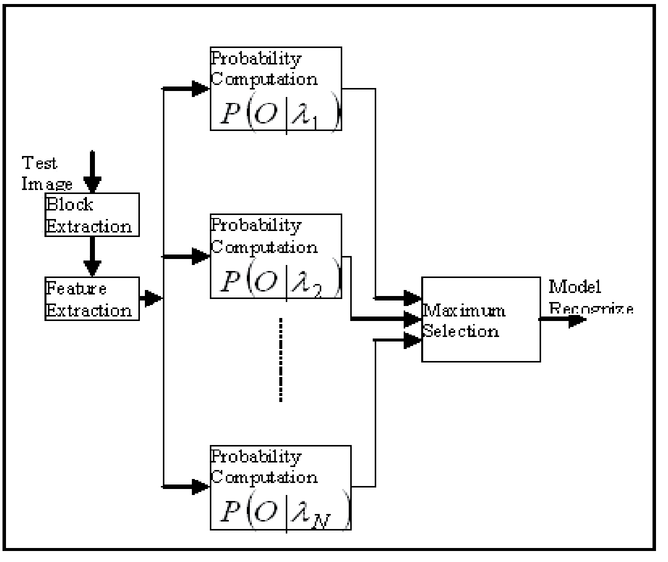


Fig. 8. HMM recognition scheme (N is the number of subjects in the database)

Step 3. the final step consists in comparing all the probabilities computed at the previous step and choosing as winner the model which returns the highest probability. The evaluation process is depicted graphically in Figure 8.

7. Implementation details

In order to implement our AFR system two different software programs were designed: one for the face detection and normalization processes and one to support the HMM based face recognition process. Many functions for face detection and recognition were based on a well known open source image processing library [OpenCV 2006]. Some details on each of these software workflows are given to facilitate other researchers.

7.1 Face detection

For the detection and cropping of all faces in the test databases we employed a well-known face detection algorithm [Viola & Jones 2000, 2001], described in section 2 above. In order to implement detection and cropping of all faces in all images in a single step, a tool was required to operate batch processes. This is implemented using Matlab. Such an approach allows additional high-level filters and other image processing techniques, also implemented in Matlab, to be easily linked with the OpenCV based face detection process. Thus the speed and efficiency of the OpenCV routines are coupled with the flexibility to incorporate supplemental Matlab filters into our test and evaluation process.
The Matlab program takes as input a folder of images, automatically loading each image, calling the face detection function from OpenCV and returning all face rectangles detected

in the given images. These are then cropped and saved on disk as new JPG image files. Note that this process facilitates a manual inspection or supplemental testing of a set of images to determine if they are correctly and uniformly cropped. To achieve the functionality of this program, there are two principal stages:

i. the declaration of the face detection function in OpenCV has to be modified in order to provide Matlab access to the detection function from the detection library using the standard mex (Matlab-C) interface[1];
ii. the OpenCV library is compiled with the Matlab-C compiler. The mex command was used to build the detection function adding all OpenCV dependencies.

At the end of this stage all detected faces were manually separated by subject in different folders. Throughout our tests we used different numbers of pictures per person in the training stage, and a fixed number of pictures in the testing stage. More exactly, we trained the system successively with 1 and up to 5 pictures per person, and we tested with the remaining 5 pictures that were not used in any training. In future tests we will denote the number of faces used for training as Nvs5: 1vs5, 2vs5, etc.

7.2 Face recognition

The second step in implementing the face recognition system was to build a program that would perform the main face recognition processes. The face recognition implementation was done in the C language using Microsoft Visual Studio. The implementation consists of three main components:

1. Top-level component is the first component and has the purpose of (i) reading multiple images from the disc, (ii) saving the output of the training stage (which is represented by the models) and (iii) analyzing the output of the testing stage. A detailed description of the training and testing stages can be found in Chap. 4;
2. Mid-level component: the second component which processes (pre-processing: illumination normalization, resize, filtering etc) the faces, computes observation vectors, builds and stores HMM models and computes likelihoods between faces and models. Please see Chapter 4 for a description of the HMM stages;
3. Low-level component is the third component which contains the basic routines of the EHMM algorithm (feature extraction, segmentation, Viterbi, state probability distribution etc) and uses functions implemented in the OpenCV library.

A detailed description of the processes taking place in each of these components and how they are interfaced is given in [Iancu 2010]. Figure 9 presents a visual explanation.

7.3 Databases and training datasets

For the experiments presented in the next sections of this chapter we used a mix of subjects from 3 standard databases: BioID, Achermann and UMIST. Each database provides some of our desired variations: BioID exhibits high variations in illumination, some expression variations and slight pose variations; Achermann presents some head rotations and slight illumination variations; UMIST covers a range of poses from frontal to semi-profile. A short description of each of these databases is given next.

BioID database: BioID[2] is a dataset consisting of 1521 gray level images with a resolution of 384 × 286 pixels, containing 23 different test persons with frontal views with variations in

[1]http://www.mathworks.com/support/compilers/interface.html
[2]http://www.bioid.com/support/downloads/software/bioid-face- database.html

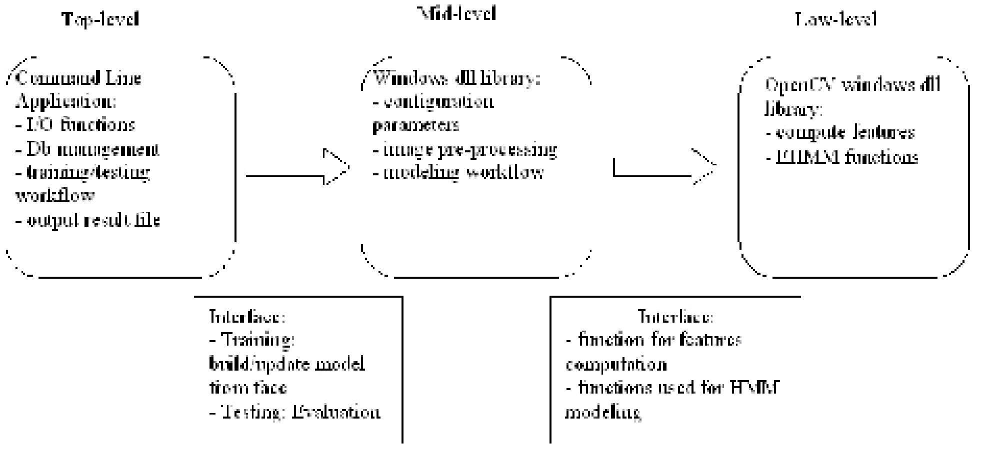


Fig. 9. Main program workflow

facial expression and illumination. The actual size of the face inside the picture is on average 128 × 128. From the entire database 200 pictures of 20 different persons were selected. Faces were selected to maximize pose and illumination variations as far as possible in the selected picture.

Achermann database: The Achermann database[3] contains 260 images of 26 people, each with 10 images, with 143×143 pixel in size. Unlike many other databases which contain only frontal views, the faces in this database span a range of pose angles.

UMIST database: UMIST database[4] consists of 564 images of 20 people, each covering a range of poses from profile to frontal views. Subjects cover a range of race/-sex/appearance. The files are all in PGM format, approximately 220 × 220 pixels in 256 shades of grey. From this database we extracted 100 pictures of 10 subjects, with pictures ranging from frontal up to 30 degrees right and left turn.

Tests were performed using the software tools described in section previously in Section 6.2 on a database formed by combining the 3 databases described above. This comvined database consisted of 560 pictures of 56 subjects, for each subject 10 pictures being selected. Unless stated otherwise, pictures were resized to 128 × 128 grayscale pixels then each face was scanned using a 12 × 12 window from left to right and top to bottom, with an overlap of 8 pixel both vertically and horizontally, in order to extract the observation vectors. The first 3×3 2D DCT coefficients corresponding to the low frequencies are retained. The EHMM depicted in Figure 5(b) is used, with 5 super states and 3-6-6-6-3 embedded states respectively, representing forehead-eyes-nose-mouth-chin areas. A standard number of 3 Gaussians were used in each state to model the feature vectors for most of the following experiments. In the training step, 1 to 5 pictures for each subject are used for building the corresponding EHMM model. After training, in order to check the algorithm, *verification* is performed, where the images used for training are also used for testing. If the algorithm is correct than the recognition rate should be 100%. In the recognition phase, the 5 images per person not used for training are utilized.

---

[3]http://iamwww.unibe.ch/ fkiwww/staff/achermann.html

[4]http://www.sheffield.ac.uk/eee/research/iel/research/face.html

8. Experimental results

As structure for the EHMM we used Samaria's thesis as starting point. According to Samaria's experiments presented in his PhD thesis [Samaria 1994], the most efficient structure allowing the smallest error rate (5.5%) consists in 5 super states with 3-6-6-6-3 embedded states respectively and was tested employing as observation vectors the pixel intensities of a face region and the dataset of face regions was drawn from the ORL database. The same EHMM structure was used by [Nefian 1999] 2D DCT coefficients as observa- tion vectors and the dataset of face regions was drawn from the ORL database. Nefian obtained an improved error rate of 2%.

8.1 Test data 1 – different sizes of face region

A first set of experiments is directed towards determining an optimal size of the detected and cropped face regions used for recognition. The sizes of the faces available in research databases can vary significantly. As examples, it varies between 128 × 128 for BioID, 143 × 143 for Achermann and 220 × 220 for UMIST. As our experiments draw images from these, and other data sources, a consistent normalization between face regions is important. Downscaling will result in information loss, but this is still preferable to upscaling which can introduce artifacts and create false information. There is also a reduction in computational demands for smaller pictures – another reason to favour downscaling. For example it takes 75% less time to process a 64×64 image region than for 128 × 128 sized regons. Thus processing smaller pictures is clearly desirable for algorithms to run efficiently in handheld devices such as digital cameras. In this set of experiments we tested sizes between 64 × 64 and 196 × 196. The bicubic spline interpolation technique was used for image downscaling where required. The results are given in the Table 2 and depicted in Figure 10.

Picture size	1vs5	2vs5	3vs5	4vs5	5vs5
64 × 64	42.86	56.79	73.57	83.57	89.64
96 × 96	52.14	64.64	76.43	86.07	87.86
128 × 128	66.43	71.07	81.07	83.57	86.07
192 × 192	61.79	69.29	77.86	83.93	83.57

Table 2. Recognition rates (%) for different picture sizes

8.2 Test data 2 – different numbers of gaussians

One of the underlying assumptions of our HMM model is that a random signal can be modeled by one or more Gaussian probability functions. If the signal is simple, it clusters around one mean value and it can be modeled by a single Gaussian. For more complex signals there may be more than one clustering center, and a more accurate model will match the number of Gaussians to the number of primary clusters. The objective of this experiment is to test how a varying number of Gaussians will influence the underlying recognition rates. We used face regions of size 128 × 128 pixels. This size of images should provide close to an optimal recognition rate as detailed in section 8.1.

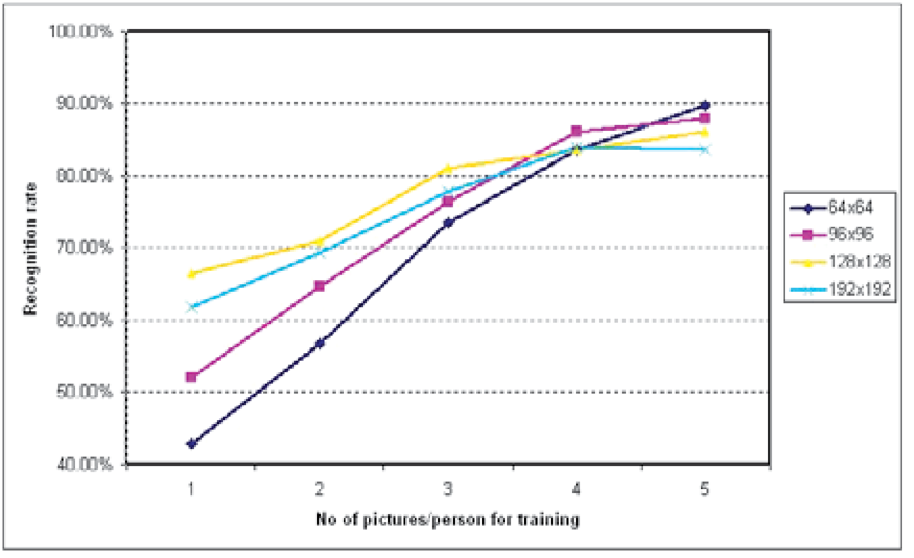


Fig. 10. Recognition rates for different picture sizes

No. of Gaussians	1vs5	2vs5	3vs5	4vs5	5vs5
1	58.93	65.71	72.14	81.07	82.14
2	58.21	67.86	79.29	82.86	86.07
3	66.43	71.07	81.07	83.57	86.07
4	55.00	67.14	78.21	84.29	88.21

Table 3. Recognition rates (%) for different numbers of Gaussians

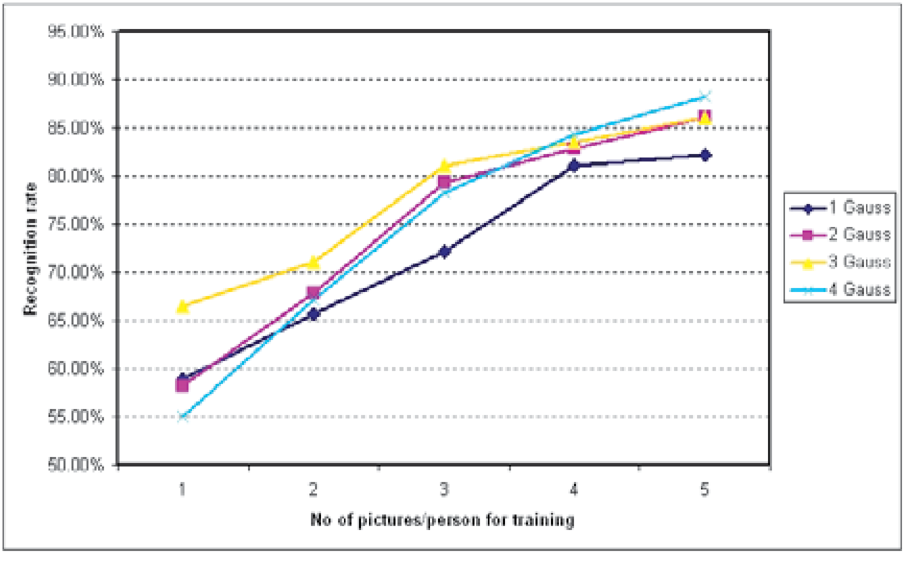


Fig. 11. Recognition rates for different numbers of Gaussian functions

The best results are obtained for mixtures of 3 Gaussians per HMM. For 2 or 4 Gaussians per HMM good results are also obtained, particularly when more training images per person are available. Note, however, that 4 Gaussians per HMM requires significantly more computation time and the benefits are marginal. Because the difference in processing time between modeling with 2 and 3 Gaussians is not significant (the recognition process with 2 Gaussians is in average 15% faster than for 3 Gaussians), it was decided to use a mixture of 3 Gaussians for our later experiments.

Note that for more complex or unusual model characteristics it is expected that the improvement in general recognition rates for 3 Gaussians could be more significant than is shown here. For now we can state that increased computation from using 3 Gaussians is likely to be balanced by the adaptability that is available to the model to handle more extreme variations in facial characteristics. No investigation of varying the underlying number of Gaussians for each superstate was undertaken. However it seems likely that superstates which cover complex features, such as the eye, nose and mouth regions are likely to benefit from increasing the underlying number of Gaussians in the more complex face regions.

8.3 Experimental results for different numbers of DCT coefficients

Again, the idea behind using fewer features to represent a signal is to speed up the recognition process, without adversely affecting the recognition rates. The number of 2D DCT coefficients we used previously was 3 × 3. In Figure 12 we show the results for 2×2, 3×3 and 5×5 DCT coefficients used. The results are more clearly presented in Table 4. We can see that on average 3×3 coefficients perform marginally better than 5 × 5 when up to 3 training pictures are used. However as more pictures are used in training, 5×5 coefficients seem to characterize the image better although in many applications this improvement is may not be sufficient to justify the additional computational effort for handheld imaging devices.

DCT	1vs5	2vs5	3vs5	4vs5	5vs5
2×2	51.79	63.57	75.71	82.86	83.21
3×3	66.43	71.07	81.07	83.57	86.07
5×5	62.14	70.36	79.64	86.07	90

Table 4. Recognition rates (%) for different numbers of DCT features

8.4 Experimental results for simplified topologies

The final tests of this section involve using a different topology for the EHMM. The core model was drawn from the work of Samaria and Nefian employs 5 superstates with an internal organization of 3-6-6-6-3 embedded states respectively, on images resized at 128 × 128. Here a simplified version of the classic EHMM is tested: the face is segmented into 4 superstates and 2-4-4-2 embedded states. Smaller face regions are used to further reduce the memory and computation required. Face regions were resized to 32 × 32 and 64 × 64. Details of the tests and results are presented in Figures 13 & 14. The numerical results can be found in Tables 5 and 6.

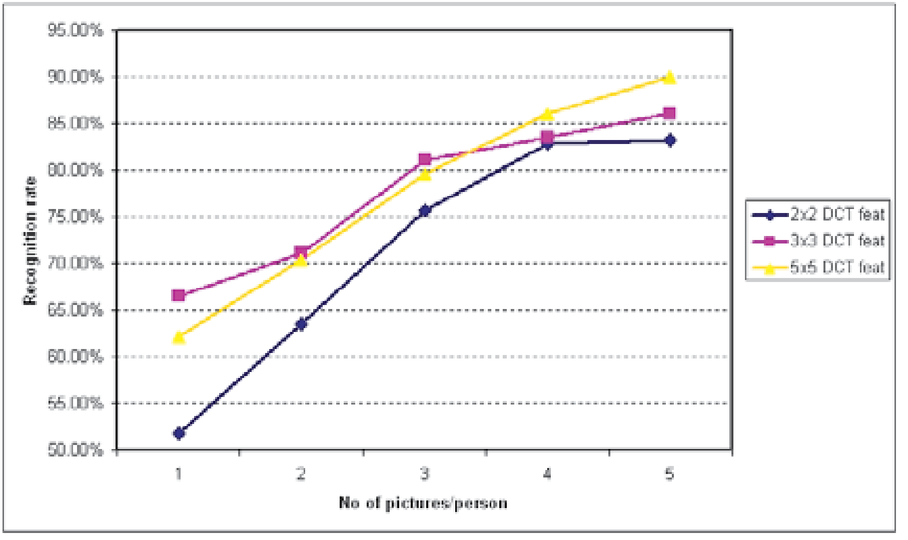


Fig. 12. Recognition rates for different numbers of features

Window size / No Gaussian	1vs5	2vs5	3vs5	4vs5	5vs5
8 × 8/1 Gauss	57.86	67.50	72.86	80.00	82.14
12 × 12/1 Gauss	51.43	51.79	64.29	76.43	83.21
8 × 8/2 Gauss	58.57	64.64	76.43	84.29	86.43
12 × 12/2 Gauss	55.36	49.64	61.07	74.64	82.14

Table 5. Recognition rates (%) for picture size 32 × 32

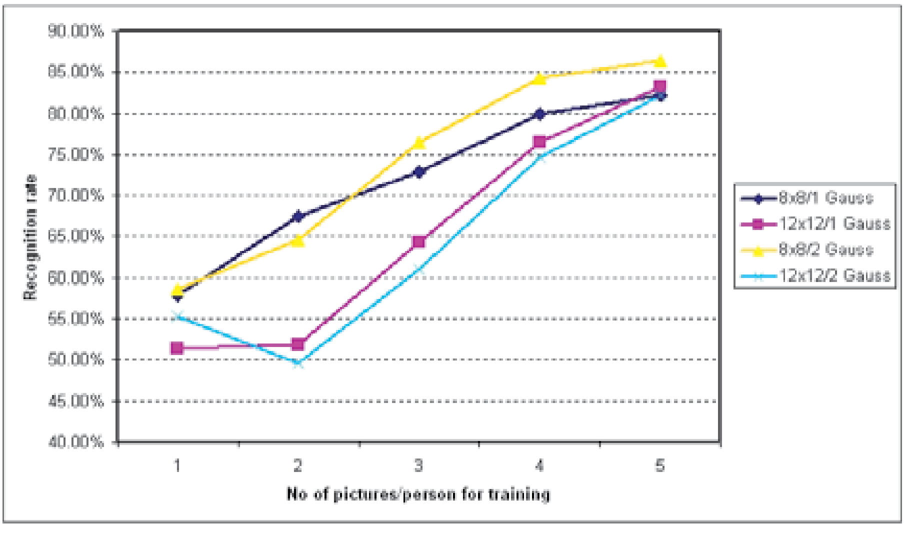


Fig. 13. Recognition rates for 32 × 32 picture size

There were other parameters we had to change in the tests in order to increase the recognition rates. Originally we started the tests on 32 × 32 pixels images scanned with a 12 × 12 pixels window with 8 pixels overlap (75% of the size of the scanning window). However a simple calculation shows that in this case we only have 6 scanning steps both vertically and horizontally so there will be superstates with only 6 observation vectors, that meaning that some states will have only 1 or 2 observation vectors to be modeled with Gaussian mixtures. Obviously modeling with 2 Gaussian mixtures a single observation vector does not make sense. Furthermore having a single observation vector per state cannot offer a detailed description of that state.

Window size/No. of Gaussians	1vs5	2vs5	3vs5	4vs5	5vs5
8 × 8/1 Gauss	55.71	61.79	69.64	75.36	76.43
12 × 12/1 Gauss	60.71	65.36	72.50	78.57	82.50
8 × 8/2 Gauss	59.64	63.93	71.79	79.64	82.86
12 × 12/2 Gauss	57.14	65.71	70.36	81.07	85.00

Table 6. Recognition rates (%) for picture size 64 × 64

Because of this, low recognition rates were obtained for 32 × 32 pixels images scanned with 12×12 pixels windows. However, for images sized 64 × 64 pixels, scanning with an 8 × 8 pixels window could mean taking too much detail and missing larger facial features. This may explain why the best results for 64 × 64 size images are obtained when scanning with 12 × 12 pixels window.

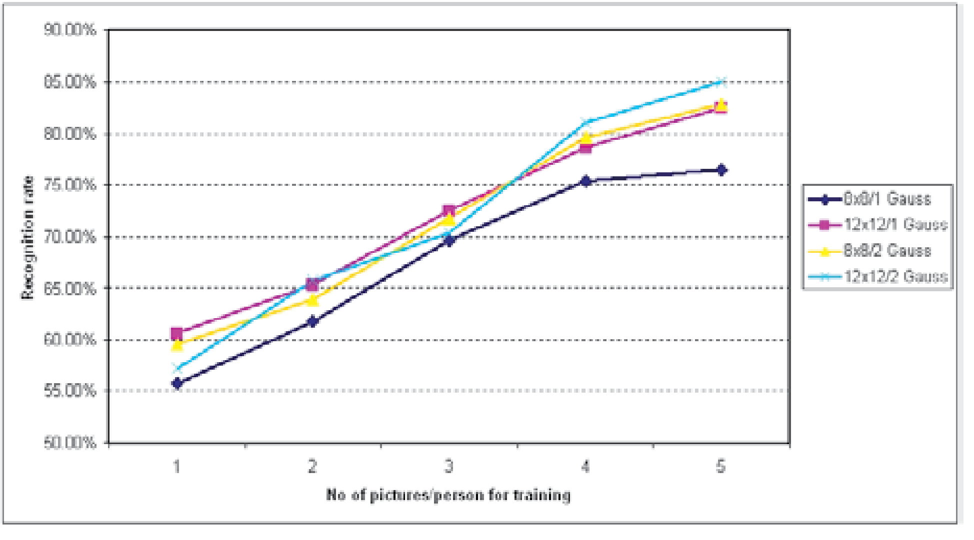


Fig. 14. Recognition rates for 64 × 64 picture size

8.5 Experiments using different illumination normalization techniques

For our experiments to tackle the problem of illumination variations a study was undertaken of the effectiveness of the following three illumination normalization algorithms: (i) *histogram equalization* (HE) based on [Gonzalez & Woods 1992], (ii) *contrast limited adaptive histogram equalization* (CLAHE) based on [Zuiderveld 1994], and (iii) the relative new method of DCT in the logarithm domain - *logDCT* based on [Chen *et al* 2006]. In addition paired combinations of several of these techniques were also evaluated. To the best of our knowledge this is the first attempt in the literature to compare different illumination normalization techniques in the context of EHMM.

Illum Norm Tech	**1vs5**	**2vs5**	**3vs5**	**4vs5**	**5vs5**
No normalization	66.43	71.07	81.07	83.57	86.07
HE	65.00	74.64	85.71	90.71	92.50
CLAHE	68.21	75.71	82.86	88.21	90.57
LogDCT	52.50	58.21	70.71	76.43	77.86
LogDCT+CLAHE	60.00	70.36	75.00	82.14	84.64
CLAHE+LogDCT	68.21	77.50	84.64	90.00	92.86
LogDCT+HE	66.43	72.86	81.43	86.79	90.00
HE+LogDCT	53.57	64.29	76.79	85.00	88.57
HE+CLAHE	68.57	78.21	86.07	92.14	95.36
CLAHE+HE	71.07	79.29	87.14	92.86	95.71

Table 7. Recognition rates(%) for illumination normalization techniques

We tested these three illumination normalization techniques, *viz*, HE, CLAHE, LogDCT and compared the recognition results with the results obtained when no illumination normalization was performed. After visually observing the results of illumination normalization on each image in the database we concluded that LogDCT flattens the facial features excessively whereas CLAHE enhances them. Thus the idea of combining these two normalization methods, was mooted. It was then decided to investigate a number of other combinations of these techniques and the results are presented below. The number of DCT components that are canceled for logDCT depends on the size of the image and the level of the illumination variation present in the image. For our tests we used the first 4 DCT coefficients for 128 × 128 images. The recognition rates for each illumination normalization technique and the combinations we tested are given in Table 7.

In Figure 15 a set of face images affected by various degrees of illumination are shown together with the results of applying each illumination normalization technique or combinations thereof to the original images. Making a parallel between the recognition rates given in Table 7 and the visual results shown in Figure 15, a few observations can be provided:

- the best recognition rates are obtained when combining HE and CLAHE regardless of the order.

- Better recognition rates are achieved using the simple HE technique when compared to the more sophisticated CLAHE and LogDCT techniques.
- LogDCT returns very poor results even when compared to the rates obtained when no illumination normalization technique is used.
- From a human's perceptual point of view, the most efficient visually illumina- tion normalization technique appears to be LogDCT+CLAHE but this gives recognition rates poorer than the original pictures

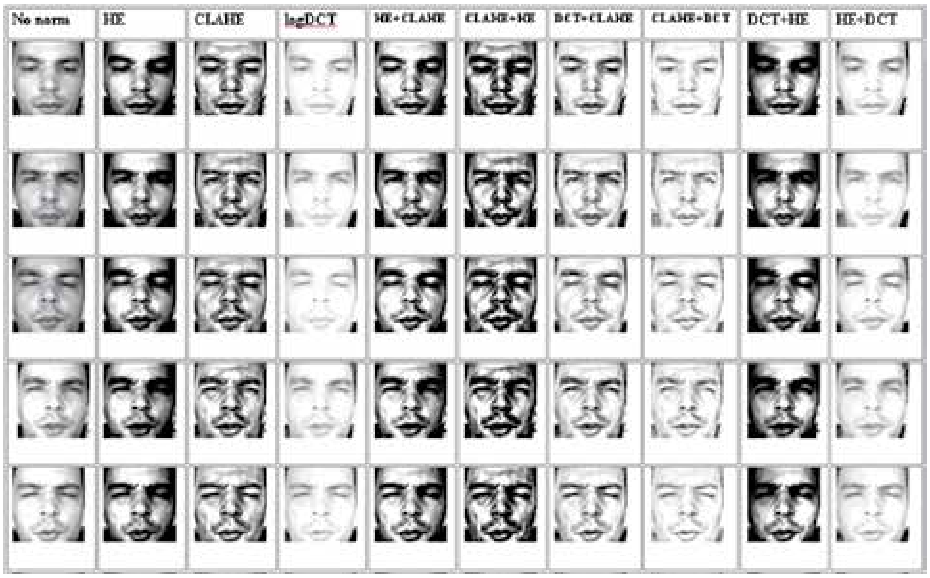


Fig. 15. Illustrative examples of various illumination normalization techniques applied to a representative subset of face data.

- The best recognition rates are obtained for illumination normalization which enhances the facial features: HE, CLAHE and combinations of these two.

There are two exceptions that contradict this last rule: (i) LogDCT+HE: despite the fact that from Figure 15 this combination appears to enhance facial features, it also appears to be unable to eliminate the influence of illumination; (ii) The second exception is CLAHE+LogDCT which gives quite good results despite looking somewhat flatter than the other techniques.

9. Discussion of our experiments

Throughout this chapter we described a series of tests performed with the purpose of finding the optimal combination of factors that influence the recognition process: size of face image, topology of the model, that is number of superstates, number of states for each super state and number of Gaussians to model the observation vectors, illumination normalization technique to diminish the differences in illuminations, and coefficients to describe the information contained in the image.

In order to make things more difficult for our recognizer, we tried to emulate as best possible the less that ideal appearance and diversity of a consumer collection by combining three different standard databases in one big collection of images. The databases used were described briefly in Section 7.3. Also, throughout the tests we used different numbers of faces of the subjects for tests and training: from 1 to 5 faces for training and the remaining 5 faces for testing. From the results presented in Section 8 the following conclusions can be reached:

i. An optimal size for face image when using HMM is 128 × 128 pixels, although the recognition rates obtained for the smaller size of 96 × 96 pixels are quite close and may offer a better choice for applications where memory and computational efficiency are important, e.g. in handheld imaging devices;
ii. From experiments performed in section 8.2 the optimal number of Gaussian functions is 3, representing a trade-off between best precision and computational burden; but we remark that with 2 Gaussians a faster computation is achieved with almost the same recognition performance; the effects of varying the number of Gaussians across super-stages was not considered in this research;
iii. The optimal performance with 2D DCT is achieved by employing the first 9 coefficients, but we noted in section 8.3 that using the first 4 coefficients gives an acceptable result as well and may be preferable where speed of computation and memory efficiency are important; no significant improvement was noted when we used Daubechies wavelets in place of DCT;
iv. In section 8.4, very good results were obtained for a reduced 2-4-4-2 EHMM topology applied on very small images: these results improve when increasing the number of training images per person, and recognition rates as high as 86.43% were achieved in our experiments. As no illumination normalization was used and these tests were performed on a combined database rather than a single standard database, the results which were obtained may be regarded as highly promising for real-world applications.
v. Finally, in Section 8.5 three different illumination normalization techniques are used in the pre-processing phase of our recognizer. We also investigated some non-standard combinations of these techniques to determine their suitability for pre-processing data for a HMM face recognition algorithm. The best results were obtained for CLAHE and HE with over 95% recognition rates. Very good performances were obtained when using a combination of CLAHE and logDCT, with 92.87% recognition rate, but also when using the more basic HE, with up to 92.5%. This analysis of various image normalization filters should pro- vide a useful baseline for future researchers in face recognition. One aspect we would have liked to investigate was the potential of such combining of illumination normalization filters to improve the performance of other well-known face recognition techniques such as PCA, ICA and AAM methods.

10. References

Y. Adini, Y. Moses, and S. Ullman. Face recognition: the problem of compensating for changes in illumination direction. IEEE Trans. on Pattern Analysis and Machine Intelligence, 19(7):721–732, 1997.

L.E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. Ann. Math. Stat, 37(6):1554–1563, 1966.

L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. Ann. Math. Stat, 41(1):164–171, 1970.

R. Chellappa, B.S. Manjunath, and C.V.D. Malsburg. A feature based approach to face recognition. IEEE Conference on Computer Vision and Pattern Recognition, pages 373–378, 1992.

R. Chellappa, C. Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. Proc. of IEEE, 83(5):705–741, May 1995.

W.L. Chen, M.J. Er, and S.Q. Wu. Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain. Systems, Man and Cybernetics, Part B, 36:458–466, 2006.

P. Corcoran, C. Iancu, and G. Costache. Improved HMM based face recognition system. OPTIM, Brasov, Romania, 4:143–146, 2006.

P. Corcoran, M.C. Ionita, and G. Costache. Pose invariant face recognition using AAM models. Proceedings of the 10th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM'06), 4:181– 184, 2006.

G.N. Costache. Advances in automated image categorization: Sorting images using person recognition techniques. PhD thesis, Dept of Electronic Engineering, NUI Galway, 2007.

G.N. Costache, P.M. Corcoran & P. Puslecki, *Combining PCA-based datasets without retraining of the basis vector set*, Pattern Recognition Letters, Volume 30, Issue 16, Pages 1441-1447, December 2009,

B. Du, S. Shan, L. Qing, and W. Gao. Empirical comparisons of sev- eral preprocessing methods for illumination insensitive face recognition. Acoustics, Speech, and Signal Processing, 2:981–984, 2005.

M. Gokmen E. Vucini and M.E. Groller. Face recognition under varying illumination. 15th WSCG, pages 57–64, 2007.

R.C. Gonzalez and R.E. Woods. Digital image processing. Addison-Wesley, Reading, MA, 1992.

R. Hummel. Image enhancement by histogram transformation. Computer Graphics and Image Processing, 6(2):184–195, 1977.

M.C. Ionita. Advances in the design of statistical face modelling techniques for face recognition. PhD thesis, Dept of Electronic Engineering, NUI Galway, 2008.

B.H. Juang and L.R. Rabiner. Automatic speech recognition - a brief history of the technology development. Elsevier Encyclopedia of Language and Linguistics, Second Edition, 2005.

T. Kanade. Computer recognition of human faces. Interdisciplinary Systems Research, 47, 1977.

S. Kuo and 0. Agazzi. Keyword spotting in poorly printed documents using pseudo 2-d hidden markov models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16:842–848, August 1994.

H.S. Le and H. Li. Face identification system using single hidden markov model and single sample image per person. IEEE International Joint Conference on Neural Networks, 1, 2004.

T.M. Lehmann, C. Gnner, and K. Spitzer. Survey: Interpolation methods in medical image processing. IEEE Transactions on Medical Imaging, 18:1049–1075, 1999.

A.V. Nefian. A hidden markov model based approach for face detection and recognition. PhD Thesis, 1999.

A.V. Nefian and M.H. Hayes III. Face detection and recognition using hidden markov models. Image Processing, ICIP 98, Proceedings. 1998 International Conference on,, 1:141–145, October 1998.

A.V. Nefian, and M.H. Hayes III, *Face recognition using an embedded HMM*, IEEE Conf. on A/V-based Biometric Person Authentication, pp. 19–24, 1999.

A.V. Nefian and M.H. Hayes III. Maximum likelihood training of the em- bedded hmm for face detection and recognition. International Conference on Image Processing, 1:33–36, 2000.

OpenCV *Online Reference Manual*, Intel Corporation 2006.

A.J. O'Toole, P.J. Phillips, F. Jiang, J. Ayyad, and N. Penardand H. Abdi. Face recognition algorithms surpass humans matching faces over changes in illumination. PAMI(29), 29(9):1642–1646, 2007.

P.J. Phillips, M. Hyeonjoon, S.A. Rizvi, and P.J. Rauss. The feret evaluation methodology for face-recognition algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(10):1090–1104, 2000.

W. K. Pratt, *Digital Image Processing*, New York: John Wiley & Sons 1991.

L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. Proc. of IEEE, 77(2):257–286, February 1989.

L.R. Rabiner and B.H. Juang. An introduction to hidden markov models. IEEE ASSP Mag, 3(1):4–16, 1986.

E. Rentzeperis, A. Stergiou, A. Pnevmatikakis, and L. Polymenakos. Impact of face registration errors on recognition. Artificial Intelligence Applications and Innovations, 204:187–194, 2006.

F. Samaria. Face recognition with hidden markov models. Ph.D. thesis, Department of Engineering, Cambridge University, UK, 1994.

F. Samaria and F. Fallside. Face identification and feature extraction using hidden markov models. Image Processing: Theory and Applications, Elsevier, pages 295–298, 1993.

F. Samaria and A.C. Harter Parameterization of a stochastic model for human face identification. Applications of Computer Vision, 1994., Pro- ceedings of the Second IEEE Workshop on, 77:138–142, December 1994.

W.A.P. Smith and E.R. Hancock. Single image estimation of facial albedo maps. Lecture Notes in Computer Science, 3704:517–526, 2005.

E. Steinberg, P. Corcoran, Y. Prilutsky, P. Bigioi, M. Ciuc, S. Ciurel, & C. Vertran, *Classification system for digital images using workflow, face detection, normalization, and face recognition*, US Patent 7,555,148, June 2009.

P. Viola and M. Jones, *Robust real-time object detection*, presented at the 2nd International workshop on Statistical and Computational Theories of Vision, Vancouver, Canada, July 13th, 2001.

M.H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(1):34–58, 2002.

W. Zhao, R. Chellappa, A. Rosenfeld, and P.J. Phillips. Face recognition: A literature survey. ACM Computing Surveys, pages 399–458, 2003.

K. Zuiderveld. Contrast limited adaptive histogram equalization. Graphics gems IV, Academic Press Professional, Inc., pages 474–485, 1994.

2

Large-Scale Face Image Retrieval: *A Wyner-Ziv Coding Approach*

Jean-Paul Kouma and Haibo Li
Digital Media Lab
Department of Applied Physics and Electronics,
Umeå University
Sweden

1. Introduction

Great progress in face recognition technology has been made recently. Since the first face recognition vendor test (FRVT)Phillips et al. (2007) in 1993, face recognition performance has been improved by two orders of magnitude in thirteen years. Notably, in the FRVT 2006 it is the first time that algorithms are capable of human performance levels, and at false acceptance rates in the range of 0.05, machines can outperform humans Phillips et al. (2007). The advances are promising for face verification applications where a typical one-to-one match is performed. It is still a grand challenge to power large-scale face image retrieval. Large-scale face image retrieval is the enabling technology behind the next generation search engines (search beyond words), by which web users can do social search with personal photos. High performance face identification algorithms are needed to support large scale face image retrieval. Compared with face verification, face identification is believed **N** times harder than face verification due to its nature of **1:N** problems. The number of individuals **N** in the database has a great impact on both the *effectiveness* and *efficiency* of face identification algorithms.

With the state of the art face identification algorithms, the identification rate is only around 70% (rank = 1) for the FERET database, a gallery of ten thousands individuals. When to serve for large-scale face image retrieval applications, the identification rate will further decrease as the gallery size increase (fortunately not linearly but logarithmically).The computing complexity of face identification is linearly related to the number of individuals N. For large-scale face image retrieval the efficiency of face identification is a key issue. In this paper we focus on the efficiency aspects of face identification.

Technically, it is very challenging to find a person from a very large or extremely large database which might hold face images of millions or hundred millions people. A highly efficient image retrieval technology is needed. Indexing technology based on tree structures has been widely used in commercial search engines. These structures are quite efficient for small dimensions (of the order of 1-10). However, as the data dimensionality increases, the query performance of these structures degrades rapidly. For instance, White and Jain report that as the dimensionality increases from 5 to 10, the performance of a nearest-neighbor query in multi-dimensional structures such as the SS-tree and the R-tree, degrades by a factor of 12. This phenomenon, known as the *dimensionality curse* is a common characteristic of all multi-dimensional index structures. In spite of the progress in the design and analysis of multi-dimensional structures such as the TV-trees, the X-trees, and the SR-trees, the dimensionality curse persists.

A very efficient approach to large-scale image retrieval is to use an approximate similarity searching strategy Tuncel et al. (2004). Without building an indexing mechanism, the search engine simply accesses *partial* information about *all* the feature vectors. Popular examples of this approach are the VA-file algorithm Weber & Blott (1997), and the dimensionality reduction techniques. Feature vectors are approximated using the accessed partial information. In the state of the art face recognition techniques, all face images in a gallery are transferred into lower resolution used for feature vectors (called face signatures). Two examples are face signatures (images) of 21-by-12 pixels used in the statistical subspace methodShakhnarovich & Moghaddam (2004) and face signatures of 28-by-23 pixels used in our 1D HMM methodLe (2008). The problem is that due to huge number of individuals in a larger gallery all signatures are too big to fit into a single server's memory. The signatures have to be stored in hard disks. The number of disk I/O operations will be the bottleneck for query processing. The processing time is approximately proportional to the size of a signature image. Therefore, compression of face signatures will play a central role in large-scale face image retrieval. High compression will lead to a fast retrieval but the distortion due to compression will affect the retrieval quality. Therefore, two types of scientific challenges are:

- *How to characterize the trade-off between retrieval quality and speed.*
- *How to efficiently compress face signatures under the fixed distortion.*

This chapter brings together our earlier works in a more detailed and coherent whole. A shorter version can be found in Kouma & Li (2009). Our contributions are:

1. *To treat the image retrieval problem as a source coding problem and the rate-distortion theory **R(D)** is used to characterize retrieval quality (**D**: distortion of coding) and retrieval speed (**R**: rate of coding)*
2. *To view compression of signature images it as a typical "Wyner-Ziv Coding" problem, which circumvents the problem that the query images are not available until we decompress the signature image*
3. *To develop a distributed coding scheme based on LDPC codes to compress face signature images*

2. Image retrieval as a Wyner-Ziv coding problem

In the 1970s Slepian and Wolf had already proved that efficient compression can also be achieved by exploring source statistics partially or wholly at the decoder only. This was known as distributed lossless coding Slepian & Wolf (1973). This is illustrated in Figure 1 by an example of compressing an information source **X** in the presence of side information **Y** when **X** and **Y** are two correlated sources. For the simplest case when **Y** is not known at all, **X** can be compressed at the rate larger than or equal to **H(X)**. When **Y** is known at both the encoder and decoder, the problem of compressing **X** is also well understood: one can compress **X** at the theoretical rate of its conditional entropy **H(X|Y)**. But what if **Y** is known

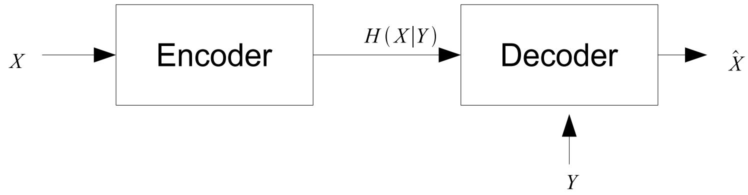


Fig. 1

only at the decoder for **X** and not at the encoder? The surprising answer from the Slepian-Wolf coding theorem Slepian & Wolf (1973) is that one can still compress **X** using only **H(X|Y)** bits, the same bits as the case where the encoder does know **Y**! This was extended to the lossy encoding by Wyner and Ziv Wyner & Ziv (1976) and yielded a similar result: *Under certain conditions, as when* **X** *and* **Y** *are jointly Gaussian with the MSE measure, when the decoder knows* **Y**, *then whether or not the encoder knows* **Y**, *the rate-distortion performance for coding* **X** *is identical.*
In face retrieval applications, for a given signature **X**, **R** $\geq$ **H(X)** bits are needed to represent it. If the query image **Y** is from the same individual, then **Y** will be highly related to **X**. If we know **Y** in advance then we don't need to store the whole information of **X**, instead just the conditional information, **R** $\geq$ **H(X|Y)**. Obviously, **H(X|Y)** $\leq$ **H(X).** As mentioned, retrieval speed is determined by (linearly proportional to) the rate **R**, so it makes large sense to reduce the rate from **H(X)** to **H(X|Y)**. The challenge is that in practice, **Y** is not known in advance and we can't directly make use of the knowledge of **Y** to help the compression of **X**. The solution is to treat it as the Wyner-Ziv coding problem: *take the query image* **Y** *as the side information*. There will be no rate loss as long as **X** and **Y** are jointly Gaussian. In fact, in the state of art face recognition techniques Gaussian modeling of human faces are commonly used. For example, in statistical subspace methods Shakhnarovich & Moghaddam (2004), it is assumed that Δ**=X-Y** is a Gaussian distribution if **X** and **Y** are from the same individual. The Gaussian distribution is used to characterize intra-personal variations Ω, caused by different facial expressions, lighting, and poses of the same individual. According to Wyner-Ziv coding theory there is no rate loss when **Y** is available only at the decoder since it is the quadratic Gaussian case. The rate will be:

$$\mathrm{R_{WZ}(D)} = \mathrm{R_{X|Y}(D)} = \frac{1}{2}\sum_{\mathrm{i}} \log\left(\frac{\text{œ}_{\mathrm{Z_i}}}{\mathrm{D_i}}\right) \tag{1}$$

where

$$\mathrm{D} = \sum_{\mathrm{i}} \mathrm{D_i} \tag{2}$$

Note that **Y***c*an be arbitrarily distributed. The rate-distortion $\mathrm{R_{WZ}(D)}$, put it in the image retrieval language, says the minimum time complexity (**R**) achieving retrieval distortion **D**. It governs the tradeoff between the retrieval quality (**D**) and retrieval speed **R** in practice.
Just as the information theory, the Wyner-Ziv theorem only tells us a theoretical bound on information rate but not how to reach the bound in practice. The image coding community has high interest in exploring how to design practical Slepian-Wolf and Wyner-Ziv codecs. With Low-density parity-check codes, when the code performance approaches the capacity

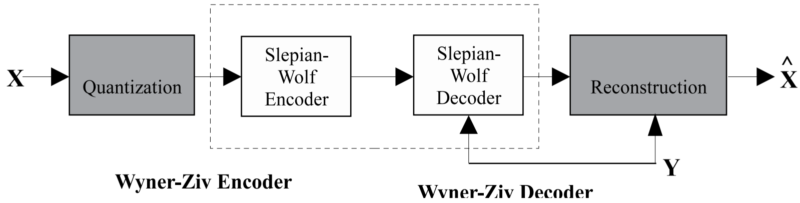


Fig. 2

of the correlation channel, the compression performance approaches the Slepian-Wolf bound (see referenced listed in Varodayan et al. (2005)). In contrast, efforts toward practical Wyner-Ziv coding have been undertaken only recently. Zamir and Shamai proved that linear codes and nested lattices might approach the Wyner-Ziv rate-distortion function if the source data and side information are jointly Gaussian Zamir et al. (n.d.). Xiong et al implemented a Wyner-Ziv encoder as a nested lattice quantizer followed by a Slepian-Wolf coder Xiong et al. (2003). In general, a practical Wyner-Ziv coder can be thought to consist of a quantizer followed by a Slepian-Wolf encoder, as illustrated in figure 2. This makes it possible for us to focus on two basic components: quantization and reconstruction. As an example of practical codec a Wyner-Ziv video coding system is reported to perform 10-12 dB better than H.263+ intra-fame codingVarodayan et al. (2005). In the face image retrieval, compression of face signature images are much more challenging than distributed video coding due to rather large variations between face images of the same person, which may be taken at different time, by using different cameras. In this paper we focus on using Slepian-Wolf coding to compress face signature images.

3. Low Density Parity-Check (LDPC) codes as Slepian-Wolf coder

Low Density Parity-Check Codes are intensively studied in other literatures, But for the sake of completeness we briefly review it here.

LDPC codes are a class of linear block codes. They were invented by Gallager in the early 60's. But due the computational complexity (at that time), LDPC codes were largely forgotten until the early 90's.

LDPC codes are specified by a sparse *parity-check* matrix $\mathbb{H}$, as well as a bipartite graph, introduced by Tanner Tanner (1981). Equation 3 and figure 3 show a parity-check matrix and its graphical representation, respectively. an LDPC code consists of N *variable nodes* (number of bits in a codeword) and M *check nodes* (number of parity bits). A check node c_m is *connected* to a variable node v_n if the element h_{ij} in $\mathbb{H}$ is 1.

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \tag{3}$$

Typically a parity-check matrix is very big - of size over 2000 entries - and very sparse.

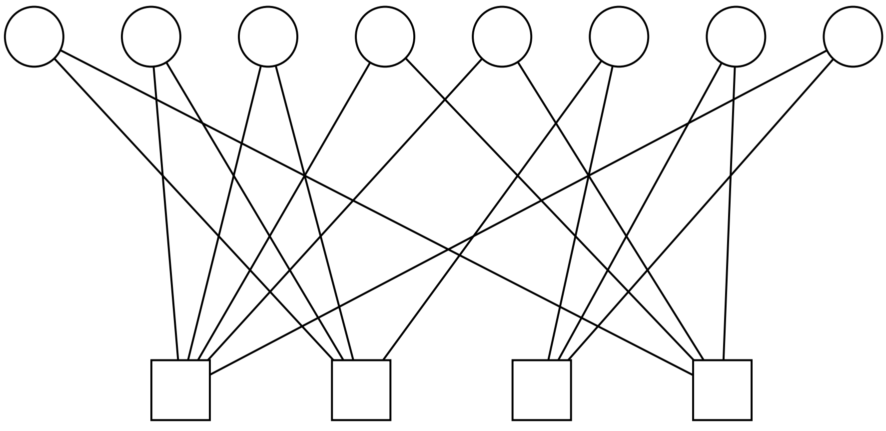


Fig. 3. Equivalent Tanner graph of parity-check matrix in equation (3)

An LDPC code is called *(ir)regular* if the total number of 1's in every column of the matrix is (not) the same as well as the total number of 1's in every row. Equivalently if all the check nodes have (not) the same number of connections to the variable nodes as well as the variable nodes to the check nodes.
LDPC codes are randomly constructed subject to these (ir)regularity constraints Gallager (1962).

3.0.1 Encoding

Given a binary source $X \in \{0,1\}^{1\times n}$ and an LDPC code $\mathbb{H} \in \{0,1\}^{k\times n}$ - $k < n$ - we multiply X with $\mathbb{H}$ and find the corresponding syndrome[1] $Z = \mathbb{H}^T X$, $Z \in \{0,1\}^{1\times n}$. Equivalently in the tanner graph we add all the variable nodes connected to the same check node. All operations are performed in modulo 2. The corresponding syndrome Z will be the compressed version of X.

3.0.2 Decoding

The decoder must estimate X, say $\hat{X}$, from Z, given $\mathbb{H}$ and $Y \in \{0,1\}^{1,n}$, known to be correlated to X. That is $Pr(X_i \neq Y_i) < 0.5, i = 1,2,...,n$.
As in Liveris et al. (2002), the conventional message passing[2] the LDPC decoder Casado et al. (2007); Leiner (2005); Shokrollahi (2003) is modified for the syndrome information to be taken into account. This yields to the following syndrome decoding algorithm:

- $\{x_i, y_i\} \in \{0,1\}, i = 1,2,...,n$ are the values in X and Y, respectively
- $s_i \in \{0,1\}, i = 1,2,...,k$ are the values in Z
- q_{ij} is the message passed variable node v_i to a check node c_j
- r_{ji} is the message passed from a check node c_j to a variable node v_i

[1] Actually the concept of compressing a binary source to its syndrome was first introduced by S. Pradhan et al. Pradhan & Ramch (1999). But that concept was rather an inspiration to constructive frameworks
[2] The message passing algorithm itself, even called *Belief propagation* in some literatures, is intensively studied in Bishop (2006); Kschischang et al. (2001)

- Q_i is the set of connected check nodes to the $i:th$ variable node.
- $Q_{i\backslash j}$ is the set of connected check nodes, excluding the $i:th$ check node, to the $j:th$ variable node.
- R_j is the set of connected variable nodes to the $j:th$ check node.
- $R_{j\backslash i}$ is the set of connected variable nodes, excluding the $i:th$ variable node, to the $j:th$ check node.

1. initialize; Prior Log Likelihood Ratios (LLR)s of x_i:

$$q_i^0 = \log \frac{Pr[x_i = 0|y_i]}{Pr[x_i = 1|y_i]} = (1 - 2y_i) \log \frac{1-p}{p} \tag{4}$$

2. Message (or LLR) sent from $i:th$ variable node to $j:th$ check node:

$$r_{ji} = 2 \operatorname{arctanh} \left((1 - 2s_j) \prod_{i' \in R_{j\backslash i}} \tanh \left(\frac{q_{i'j}}{2} \right) \right) \tag{5}$$

3. Message sent from $j:th$ check node to $i:th$ variable node:

$$q_{ij} = q_i^0 + \sum_{j' \in Q_{i\backslash j}} r_{j'i} \tag{6}$$

4. Hard decision:

$$\hat{x}_i = \begin{cases} 0, & \text{if } q_i^0 + \sum_{j=1}^{k} q_{ij} \geq 0 \\ 1, & \text{otherwise} \end{cases} \tag{7}$$

5. If $\mathbb{H}^T \hat{X} \doteq Z$, stop. Else goto 3

3.1 Potentials and limitations of LDPC codes

We carried out approximatively the same simulations as in Liveris et al. (2002). We compressed a (randomly generated) binary source X with codeword length $n = 16384$ bits to different compression ratios. The side information was generated with different crossover probabilities. The rates were increased until lossless compression was achieved. The results are presented in table 1

Although experimental results showed that LDPC-based compression of binary sources provides rates very close to Slepian-Wolf bound, it is important to mention a few caveats:

- No convergence at all is observed. Probable causes:
  - The **real** crossover probability is higher than required
  - The source signals are "far"[3] [4] from **random**

[3] Experimentally, source signals do not need to be *strictly* random for the decoder to work

[4] There are ongoing research especially dealing with non-random sources Garcia-Frias & Zhong (2003)

- The maximum number of iterations is reached[5], but convergence is not.
- Convergence is reached but the decoded codeword is the wrong one.

To generate parity-check matrix we used the implementation from Avudainayagam (2002). Additionally Liveris et al. showed that the performance achieved by LDPC codes is seen to be better than recently published results using Turbo codes. LDPC codes seem therefore to be more attractive solution to our "Wyner-Ziv Coding of Face Images" problem.

Crossover probabilities	0.01	0.05	0.1	0.2
Theoretical conditional entropies	0.169	0.286	0.469	0.722
Experimental conditional entropies	0.300	0.420	0.600	0.880
Experimental conditional entropies in Liveris et al. (2002)	0.310	0.435	0.630	0.890

Table 1. Lossless compression results using LDPC

4. Compression of face signature images using LDPC codes

Since the query images are not available when face signature images are compressed, a direct solution is to treat face signature images as binary sources X. Since X_i and X_j are two views from the same person and they are highly correlated with $Pr[X_i \neq X_j] = p < 0.5$. To allow the use of distributed coding to compress X the correlation between X_i and X_j can be modeled with a binary symmetric channel (BSC) with crossover probability p as shown in Fig 4 Following Liveris et al. (2002), LDPC codes $\mathbb{H}$ will be used to compress the binary sources with the query image as side information. That is, given a binary source X and a LDPC code $\mathbb{H}$, which is a $k \times n$ parity-check matrix, we multiply X with $\mathbb{H}$ and find the corresponding syndrome $Z = \mathbb{H}X$ with the length $(n-k)$. The LDPC decoder estimates the n-length sequence X from its (n-k) length syndrome X and the side information, query image Y (length n). The system is shown in Fig 5. The compression ratio achieved with this scheme is $\frac{n}{n-k}$. Figure 12 illustrates faces images and their respective syndrome face (signatures).

4.1 Binary coding of face signature images

The simplest way to transfer a grayscale face image into binary sequences is to employ the bit-plane coding to convert each gray level to its binary representation with prefixed resolution and then encode each bit-plane separately. Figure 6 shows the probability distributions for inter-face and intra-face variations over a small-scale face database (containing 40 subjects with 10 photos each). The low correlation is caused by the fact that the bit-plane coding is very sensitive to luminance changes, small changes in gray level can have a significant impact on the complexity of the bit planes. Obviously, it is inadequate to use LDPC to compress bit-planes, directly.

Since our preliminary goal is to investigate how to use LDPC to compress face signature images, here we just select a working binary coding scheme for our experiments. Our choice goes to Expectation Maximization (EM) Dempster et al. (1977) for segmentation Weiss (1997).

[5] For the sake of simplicity, we designed a decoding scheme that runs a predefined number of iterations. Intense studies for convergence rules are carried out in Casado et al. (2007); Daneshgaran et al. (2007); Hou et al. (2001); Matache et al. (2000)

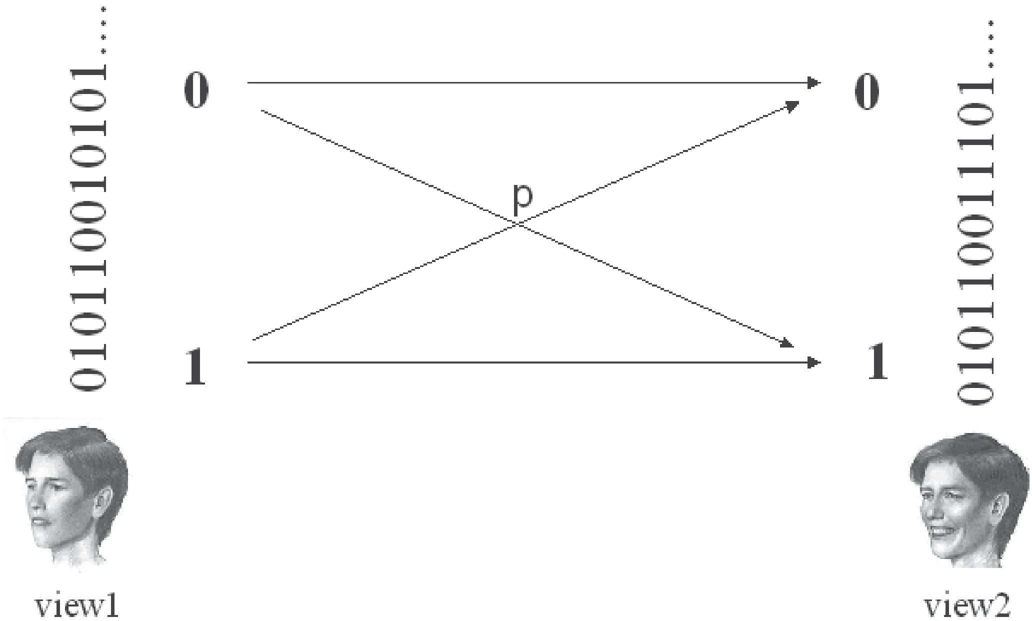


Fig. 4. Intra-face variation example

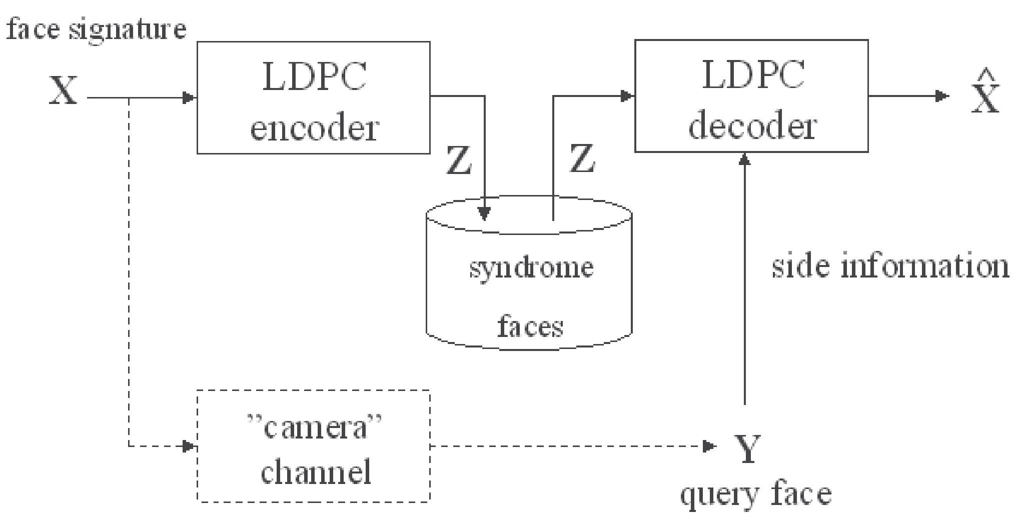


Fig. 5. Overview of system

EM attempts to assign objects a class but in a unsupervised way, tending to maximize the inter-class variation, while keeping the within-class semantic. Figure 7 shows segmentation results using EM. The results in figure 8 show that intra-face and inter-face variations are approximately $p_{intra} \sim N(0.21, 1.4310 \cdot 10^{-2})$ and $p_{inter} \sim N(0.35, 1.3510 \cdot 10^{-2})$, respectively. A certain improvement in correlation over intra-person faces is noticed (as shown in figure 6, yet not sufficient, because to employ LDPC the crossover probability has to be less than

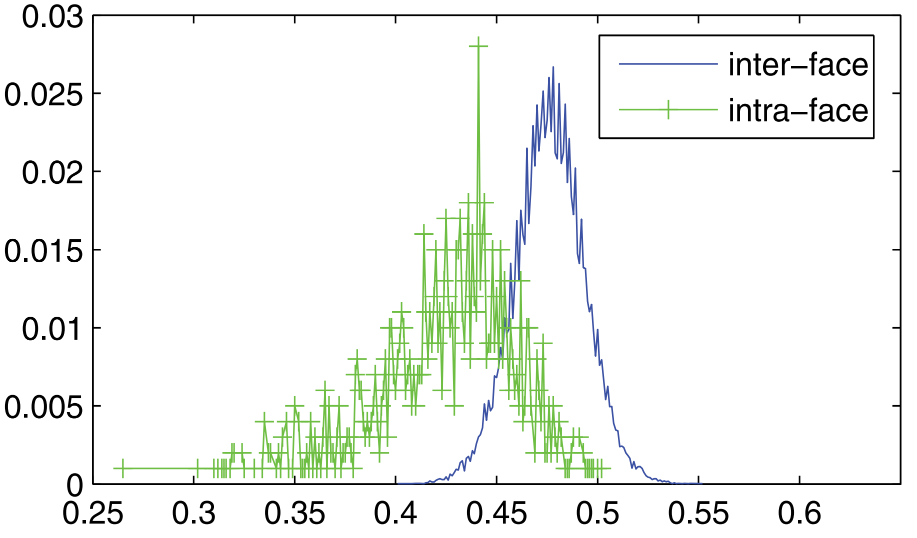


Fig. 6. Distribution of facial variation in grayscale

Fig. 7. Segmentation using EM

0.2 Liveris et al. (2002). We have to go for further processing. After carefully examining all face signatures one will see that intra face variations are mainly caused by the following factors: facial expressions, face poses, illuminance changes, camera factors etc. We have to do alignment between face signatures images.

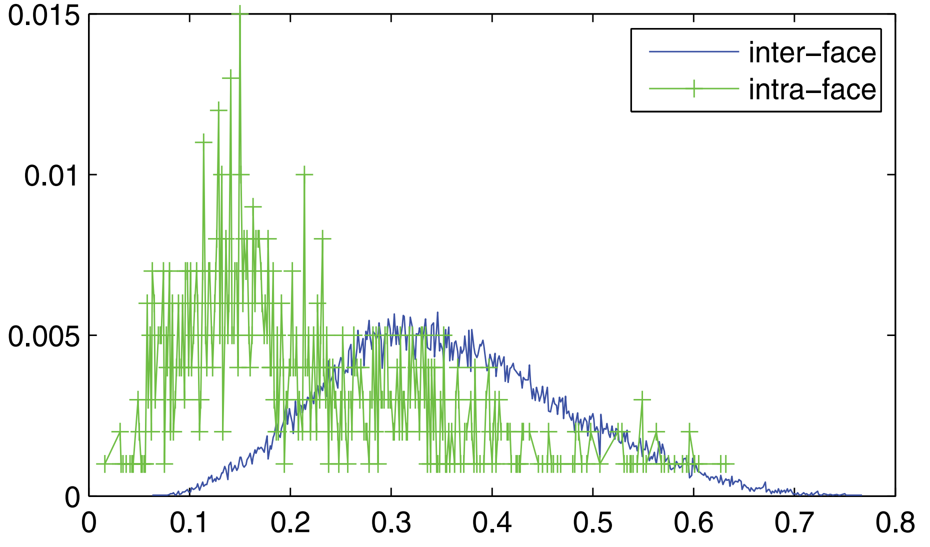


Fig. 8. Distribution of facial variation in binary before alignment

Fig. 9. Alignment result on some input images. First row: reference image, second row: input images, last row: result from alignment

4.2 Motion compensated alignment

To take away the displacement between two face signature images, a motion-compensated alignment technique is employed Li & Forchheimer (1995). The idea is first to select a natural face from the database as the reference face. To align a face signature image with respect to the reference image, the reference image has to be divided into blocks, $\{D_k\}$. Given a block D_k in the reference image, the aim is to find the corresponding block B_i in the face image by minimizing the distance $d(D_k, B_i)$. The distance is given by:

$$d(D_k, B_i) = |D_k - (a_0 + a_1 B_i)| \tag{8}$$

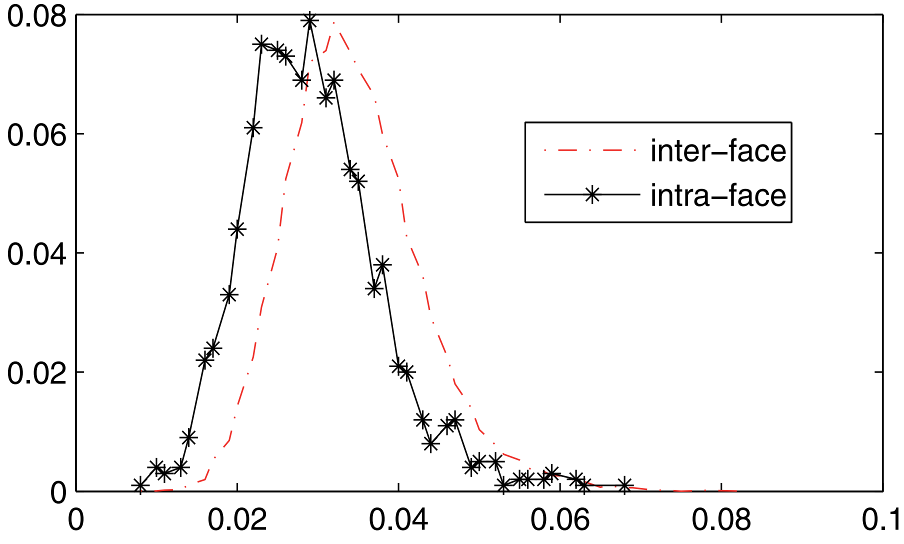


Fig. 10. Face distributions in binary vs. crossover probabilities, after alignment. dash-star line: intra-face. dash-point line: inter-face.

Coefficients a_0 and a_1 are defined by

$$a_0 = u_d - \frac{\sigma_d}{\sigma_b} u_b \tag{9}$$

$$a_1 = \frac{\sigma_d}{\sigma_b} \tag{10}$$

where u_b and σ_b are the mean value and variance of the block B_i and u_d and σ_d are the mean value and variance of the block D_k. The effect of the employed motion compensated alignment on face signature images is shown in figure 9. It is noted here that motion compensated alignment is performed before binary coding via the EM segmentation approach.

We carry out experiments over the face database and computed inter-face and intra-face variations as shown in figure 10. p_{inter} and p_{intra} can be modeled as normal distributions, $p_{inter} \sim N(0.035, 7.04 \cdot 10^{-5})$ and $p_{intra} \sim N(0.029, 7.17 \cdot 10^{-5})$. The small intra face variations make it perfect to use LDPC for compression of face signature images. More important, we know the compression bound: the theoretical limit for lossless compression of face signature images X is

$$nR \geq nH(X|Y) = nH(p) \tag{11}$$

$$H(p) = -plog_2(p) - (1-p)log_2(1-p) \tag{12}$$

Here $p = 0.0298$ corresponding to $H(X|Y) = 0.1934$, that is, a compression ratio of 5 can be achieved.

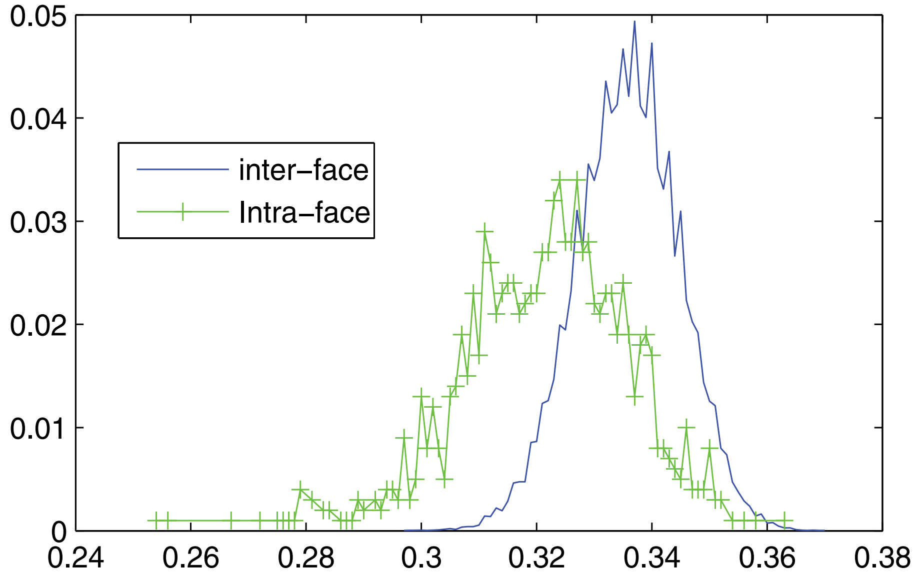


Fig. 11. Facial variation distributions in grayscale after alignment. Solid line: inter-face. dash-plus line: intra-face.

5. Retrieval metric

We introduce a similarity measure where the key is using syndrome decoding Liveris et al. (2002) and normalized Hamming distance.
At the retrieval phase, Given a query Y, we will process Y in the same manner as in the enrollment phase, first motion compensated alignment followed by binary coding. For each syndrome Z_i, $\hat{X}_i$ is estimated with respect to $\mathbb{H}$. This is equivalent to the Slepian-Wolf's insight of *sources coding with side information available only at the decoder* Slepian & Wolf (1973), where Y represents the side information and Z_i, the compressed correlated source. See figures 2 and 5. Normalized Hamming distance is performed between every $(Y, \hat{X}_i)$ pair. The normalized Hamming Distance is given by:

$$D_i = \frac{1}{n} \sum_{j=1}^{n} Y_j \oplus \hat{X}_{ij} \tag{13}$$

The templates are then ranked according to their distance to the query.

6. Preliminary results

In our experiment we use the ORL Database of Faces. In the database there are 10 different images of each of 40 distinct subjects, taken at different times, varying light conditions and facial expressions. For our purpose 5 randomly chosen images out 10 of each 40 subjects are used as training set and 5 for validation and test. It is also important to mention that the images were resized to 28×24 before further processing. The resizing parameters are

mainly motivated by the psychological assumption made in Torralba et al. (2008) and our own research on face recognition Le (2008).
Three LDPC codes are employed corresponding to different compression ratios, $R = 0.31$, $R = 0.50$ and $R = 0.76$. Recall that we found experimentally that $p_{inter} \sim N(0.035, 7.04 \cdot 10^{-5})$ and $p_{intra} \sim N(0.029, 7.17 \cdot 10^{-5})$. Theoretically a compression rate $R = 0.1934$ is thus expected Slepian & Wolf (1973). This makes great sense since a 28×24 grayscale image, when transformed to binary, requires 672 bits to be stored. With The theoretical compression rate 537 bits saved. Using an LDPC code with rate $R = 0.31$, we were able to save 464 bits per template, while achieving comparable results the with the scheme with no compression. Thus for 200 templates, we save 92800 bits! Figure 13 reports performance results, where the retrieval efficiency is plotted against the number of outputs. the line specifications in figure 13 are denoted as follow:

- *Solid-asterisk*: Alignment followed by bit-plane-wise binary representation
- *Solid*: Alignment followed by binarization. No compression
- *Dash-dot*: proposed scheme with rate 0.3
- *Solid-upward triangle*: proposed scheme with rate 0.5
- *Solid-downward triangle*: proposed scheme with rate 0.7

Fig. 12. Face images and respective resulting syndrome

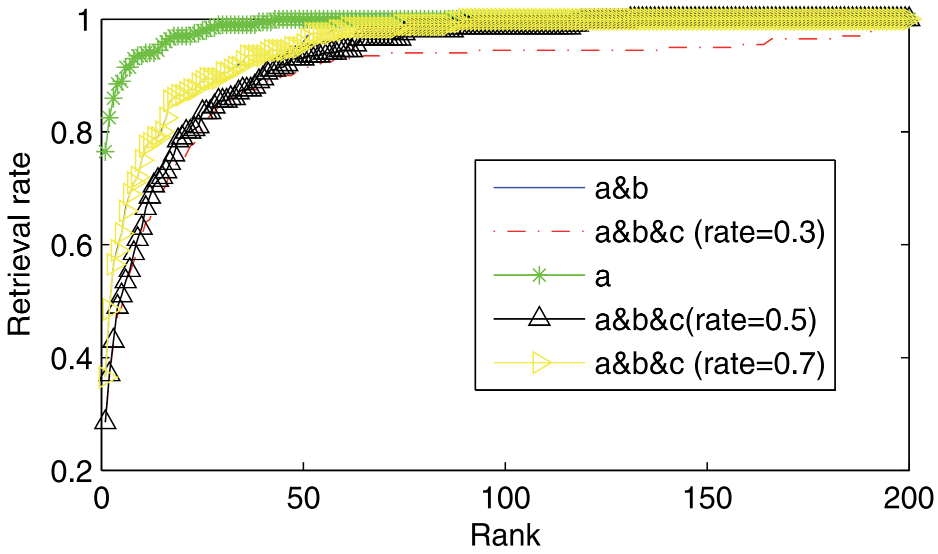


Fig. 13. Recognition Rate vs. Rank

7. Concluding remarks

Wyner-Ziv coding is radically different from conventional image coding. It gives a totally new coding paradigm. Most research efforts are devoted to how image and video compression can be done under the new paradigm. This paper is the debut of our effort to investigate how Wyner-Ziv coding can be used for large-scale image retrieval problem.

Image coding and image retrieval have been conventionally two different disciplines. In this paper image retrieval is considered as an image-coding problem. The powerful rate-distortion theory can be directly used to characterize the tradeoff between retrieval quality and retrieval speed through the crossover probability p.

Wyner-Ziv coding has a great potential to improve the efficiency of large-scale image retrieval. Under the Wyner-Ziv coding framework the query information provided by a huge number of web users can be utilized to reduce the storage and transmission of face images. Considering that Google receives hundreds of millions of queries per day and they use a million servers to run their search service, it is a big impact to our environment if consumption of storage and transmission can be reduced 90% by adopting Wyner-Ziv coding.

The results we reported here are very preliminary. We focus ourselves on how to use LDPC codes to compress binary coding of face signature images X to reach the Slepian-Wolf bound $H(X|Y)$. We haven't addressed at all how to quantize X to achieve Wyner-Ziv coding, $H(Q(X)|Y)$. To focus on LDPC coding of X, we just use an EM-strategy to do binary coding of face signature images and use it for our benchmark. We already see that the binary coding results in a significant loss in the quality of image retrieval. To build an efficient whole system, the study of $Q(X)$ has to be carried out. In addition, motion compensated alignment plays a very important role and has a big impact on both compression efficiency and retrieval quality. How to achieve an optimal alignment is an important topic for future research.

8. References

Avudainayagam, A. (2002). Ldpc toolkit for matlab.
URL: *http://arun-10.tripod.com/ldpc/ldpc.htm*

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*, Springer, chapter 8, pp. 360–418. URL: *http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0387310738*

Casado, A. I. V., Griot, M. & Wesel, R. D. (2007). Informed dynamic scheduling for belief-propagation decoding of ldpc codes, *Communications, 2007. ICC '07. IEEE International Conference on*, pp. 932–937.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4288829*

Daneshgaran, F., Laddomada, M. & Mondin, M. (2007). Ldpc-based iterative algorithm for compressionof correlated sources at rates approaching the slepian-wolf bound, *Technical report*.

Dempster, A. P., Laird, N. M. & Rdin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm, *Journal of the Royal Statistical Society, Series B* 39: 1–38.

Gallager, R. (1962). Low-density parity-check codes, *Information Theory, IEEE Transactions on* 8(1): 21–28.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1057683*

Garcia-Frias, J. & Zhong, W. (2003). Ldpc codes for compression of multi-terminal sources with hidden markov correlation, 7(3): 115–117.

Hou, J., Siegel, P. H. & Milstein, L. B. (2001). Performance analysis and code optimization of low density parity-check codes on rayleigh fading channels, *Selected Areas in Communications, IEEE Journal on* 19(5): 924–934.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=924876*

Kouma, J.-P. & Li, H. (2009). Large-scale face images retrieval: A distribution coding approach, *International Conference on Ultra Modern Telecommunications (ICUMT 2009)*, St. Petersburg, Russia.

Kschischang, F. R., Frey, B. J. & Loeliger, H. A. (2001). Factor graphs and the sum-product algorithm, 47(2): 498–519.

Le, H. S. (2008). *Face Recognition: A Single View Based HMM Approach*, PhD thesis, Umeå University, Faculty of Science and Technology, Applied Physics and Electronics.
URL: *urn:nbn:se:umu:diva-1485*

Leiner, B. M. (2005). Ldpc codes Ű a brief tutorial, *Technical report*.

Li, H. & Forchheimer, R. (1995). A transformed block-based motion compensation technique, *IEEE Transactions on Communications*, Vol. 43, pp. 1673–1676.

Liveris, A. D., Xiong, Z. & Georghiades, C. N. (2002). Compression of binary sources with side information at the decoder using ldpc codes, *Communications Letters, IEEE* 6(10): 440–442.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1042242*

Matache, A., Dolinar, S. & Pollara, F. (2000). Stopping rules for turbo decoders, *JPL TMO Progress Report 42-142, Aug. 15, 2000. 103*, pp. 42–142.

Phillips, J. P., Scruggs, T. W., OŠtoole, A. J., Flynn, P. J., Bowyer, K. W., Schott, C. L. & Sharpe, M. (2007). Frvt 2006 and ice 2006 large-scale results, *Technical report*, National Institute of Standards and Technology.

Pradhan, S. S. & Ramch, K. (1999). Distributed source coding using syndromes (discus): Design and construction, *IEEE Trans. Inform. Theory* 49: 626–643.

Shakhnarovich, G. & Moghaddam, B. (2004). Face recognition in subspaces, *in: S.Z. Li, A.K. Jain (Eds.), Handbook of Face Recognition*, Springer, pp. 141–168.

Shokrollahi, A. (2003). Ldpc codes: An introduction, *Digital Fountain, Inc., Tech. Rep* p. 2.

Slepian, D. & Wolf, J. K. (1973). Noiseless coding of correlated information sources, *Information Theory, IEEE Transactions on* 19(4): 471–480.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1055037*

Tanner, R. (1981). A recursive approach to low complexity codes, *Information Theory, IEEE Transactions on* 27(5): 533–547.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1056404*

Torralba, A., Fergus, R. & Freeman, W. T. (2008). 80 million tiny images: A large data set for nonparametric object and scene recognition, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 30(11): 1958–1970.
URL: *http://dx.doi.org/10.1109/TPAMI.2008.128*

Tuncel, E., Koulgi, P. & Rose, K. (2004). Rate-distortion approach to databases: storage and content-based retrieval, *Information Theory, IEEE Transactions on* 50(6): 953–967.

Varodayan, D., Aaron, A. & Girod, B. (2005). Rate-adaptive distributed source coding using low-density parity check codes, *In: Proc. Asilomar Conf. on Signals, Syst., Comput*.

Weber, R. & Blott, S. (1997). An approximation-based data structure for similarity search.

Weiss, Y. (1997). Motion segmentation using em - a short tutorial.

Wyner, A. & Ziv, J. (1976). The rate-distortion function for source coding with side information at the decoder, *IEEE Transactions on Information Theory* 22(1): 1–10.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1055508*

Xiong, Z., Liveris, A., Cheng, S. & Liu, Z. (2003). Nested quantization and slepian-wolf coding: a wyner-ziv coding paradigm for i.i.d. sources, *Statistical Signal Processing, 2003 IEEE Workshop on* pp. 399–402.

Zamir, R., Shamai, S. & Erez, U. (n.d.). Nested linear/lattice codes for structured.

Part 2

3D Methods for Face Recognition

3

3D Face Recognition

Naser Zaeri
Arab Open University
Kuwait

1. Introduction

Biometric systems for human recognition are an ongoing demand. Among all biometric technologies which are employed so far, face recognition is one of the most widely outspread biometrics. Its daily use by nearly everyone as the primary mean for recognizing other humans and its naturalness have turned face recognition into a well-accepted method. Furthermore, this image procurement is not considered as intrusive as the other mentioned alternatives.

Nonetheless, in spite of the various facial recognition systems which already exist, many of them have been unsuccessful in matching up to expectations. 2D facial recognition systems are constrained by limitations such as physical appearance changes, aging factor, pose and changes in lighting intensity. Recently, to overcome these challenges 3D facial recognition systems have been issued as the newly emerged biometric technique, showing a high level of accuracy and reliability, being more robust to face variation due to the different factors.

A face-based biometric system consists of acquisition devices, preprocessing, feature extraction, data storage and a comparator. An acquisition device may be a 2D-, 3D- or an infra-red- camera that can record the facial information. The preprocessing can detect facial landmarks, align facial data and crop facial area. It can filter irrelevant information such as hair, background and reduce facial variation due to pose change. In 2D images, landmarks such as eye, eyebrow, mouths etc, can be reliably detected, in contrast, nose is the most important landmark in 3D face recognition.

The 3D information (depth and texture maps) corresponding to the surface of the face may be acquired using different alternatives: A multi camera system (stereoscopy), range cameras or 3D laser and scanner devices. Different approaches have been presented from the 3D perspective. The first approach would correspond to all 3D approaches that require the same data format in the training and in the testing stage. The second philosophy would enclose all approaches that take advantage of the 3D data during the training stage but then use 2D data in the recognition stage. Approaches of the first category report better results than of the second group; however, the main drawback of this category is that the acquisition conditions and elements of the test scenario should be well synchronized and controlled in order to acquire accurate 3D data. Thus, they are not suitable for surveillance applications or control access points where only one "normal" 2D texture image (from any view) acquired from a single camera is available. The second category encloses model-based approaches. Nevertheless, model-based face recognition approaches present the main drawback of a high computational burden required to fit the images to the 3D models.

In this chapter, we study 3D face recognition where we provide a description of the most recent 3D based face recognition techniques and try to coarsely classify them into categories, as explained in the following subsequent sections.

2. Iterative closest point

(Maurer et al., 2005) presented a multimodal algorithm that uses Iterative Closest Point (ICP) to extract distance map, which is the distance between mesh of reference and probe. This method includes, face finding, landmark finding, and template computation. They used weighted sum rule to fuse shape and texture scores. If 3D score is high, algorithm uses only shape for evaluation. In experimental tests by using 4007 faces in the FRGC v2 database, a verification rate of 87.0% was achieved at %0.1 false accept rate (FAR). (Kakadiaris et al., 2007) performed face recognition with an annotated model that is non-rigidly registered to face meshes through a combination of ICP, simulated annealing and elastically adapted deformable model fitting. A limitation of this approach is the imposed constraints on the initial orientation of the face.

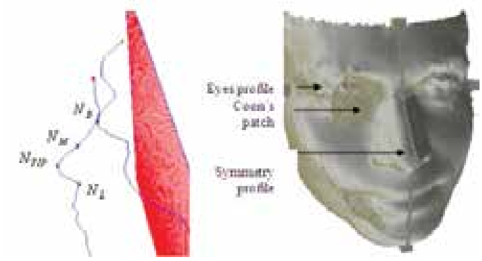
Performance of 3D methods highly depends on registration performance, where ICP is commonly used. ICP registration performance is highly dependent on initial alignment and it performs solid registration. However, expression variations degrade registration success. To overcome this problem, (Faltemier et al., 2008) divided the face into different overlapping regions where each face region was registered independently. Distance between regions was used as a similarity measure and results were fused using modified Borda count. They achieved 97.2% rate on FRGC v2 database. Other approaches to discard the effect of expressions were also studied by dividing the face into separate parts and extracting features from each part in 2D and range images (Cook et al., 2006; McCool et al., 2008).

3. Geometric approach

The early work of applying invariant functions on 3-D face recognition was done over a decade ago. At that time, people began with the geometrical properties introduced in differential geometry, such as principal curvatures, Gaussian curvature, etc. Basically, these approaches use the invariant functions, e.g., Gaussian curvature which is invariant under Euclidean transformations, to extract information from the face surface and, then, perform a classification that is based on the extracted information. (Riccio & Dugelay, 2007) proposed a particular 2D-3D face recognition method based on 16 geometric invariants, which were calculated from a number of "control points". The 2D face images and 3D face data are related through those geometric invariants. The method is invariant to pose and illumination, but the performance of the method closely depend on the accuracy of "control points" localization.

In the approach proposed by (Elyan & Ugail, 2009), the first goal was to automatically determine the symmetry profile along the face. This was undertaken by means of computing the intersection between the symmetry plane and the facial mesh, resulting in a planner curve that accurately represents the symmetry profile. Once the symmetry profile is successfully determined, a few feature points along the symmetry profile are computed. These feature points are essential to compute other facial features, which can then be utilized to allocate the central region of the face and extract a set of profiles from that region

(Fig. 1). In order to allocate the symmetry profile, it was assumed that it passes through the tip of the nose. This was considered as the easiest feature point to recover and to allocate using a bilinear blended Coon's surface patch. Coon's patch is a parametric surface defined by a given four-boundary curves. In (Elyan & Ugail, 2009), the four boundaries of the coon's patch were determined based on a boundary curve that encloses an approximated central region of interest, which is simply the region of the face that contains or likely to contain the nose area. This region was approximated based on the centre of the mass that represents the 3D facial image. They have computed the Fourier coefficients of the designated profiles and stored it in a database, other than storing the actual points of the profile. Thus, having a database of images representing different individuals where each person was represented by two profiles stored by means of their Fourier coefficients.



(a) (b)

Fig. 1. Facial features identification (a) Symmetry profile identification and analysis based on depth value to the reference depth plane (b) Eyes profile shown as the profile that passes through the nose bridge (Elyan & Ugail, 2009)

Moreover, several works in the literature propose to map 3D face models into some low-dimensional space, including the local isometric facial representation (Bronstein et al., 2007), or conformal mapping (Wang et al., 2006). Some works, for simplification, try also to investigate partial human biometry, meaning recognition based only on part of a face, as for example in (Drira et al., 2009), where authors used the nose region for identification purposes. (Szeptycki et al., 2010) explored how conformal mapping to 2D space (Wang et al., 2006) can be applied to partial face recognition. To deal with the computational cost of 3D face recognition they have utilized conformal maps of 3D surface to a 2D domain, thus simplifying the 3D mapping to a 2D one. The principal issue addressed in (Szeptycki et al., 2010) was to create facial feature maps which can be used for recognition by applying previously developed 2D recognition techniques. Creation of 2D maps from 3D face surfaces can handle model rotation and translation. However, their technique can be applied only to images with variation in pose and lighting. The expression changes were avoided. To create face maps which are later used for recognition, they started with models preprocessing (hole, spike removal). Next step was to segment the rigid part of a face that has less potential to change during expression. Finally, they performed UV conformal parameterization as well as shape index calculation for every vertex; the process is shown in Fig. 2.

(Song et al., 2009) detected the characteristics of the three regions eyes, nose and mouth in the human face, and then calculated the geometric characteristics of these regions by finding the straight-line Euclidean distance, curvature distance, area, angle and volume. Another

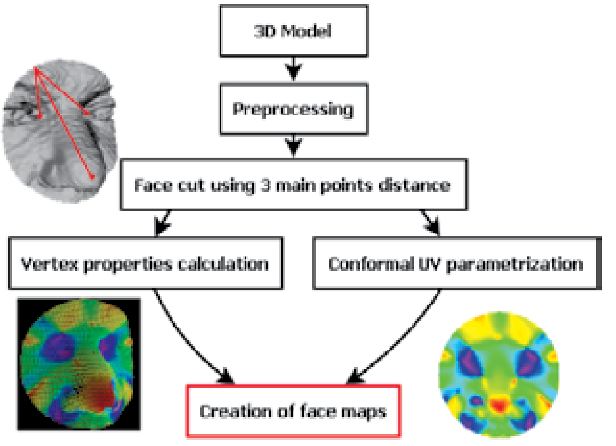


Fig. 2. Face maps creation flow chart (Szeptycki et al., 2010)

face recognition system that is based on 3D geometric features was developed by (Tin & Sein, 2009). It is based on the perspective projection of a triangle constructed from three nodal points extracted from the two eyes and lips corners (Fig. 3). The set of non-linear equations was established using the nodal points of a triangle built by any three points in a 2D scene.

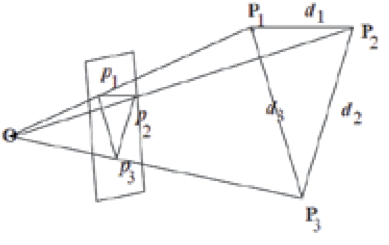


Fig. 3. Illustration of Perspective Projection of a 3D Triangle (Tin & Sein, 2009)

An automatic 3D face recognition system using geometric invariant feature was proposed by (Guo et al., 2009). They utilized two kinds of features, one is the angle between neighboured facets, they made it as the spatial geometric feature; the other is the local shape representation vector, and they made it as the local variation feature. They combined these two kinds of features together, and obtained the geometric invariant feature. Before feature extraction, they have presented a regularization method to build the regular mesh models. The angle between neighboured facets is invariant to scale and pose; meanwhile, local shape feature represents the exclusive individual shape.

(Passalis et al., 2007) focused on intra-class object retrieval problems, specifically, on face recognition. By considering the human face as a class of objects, the task of verifying a person's identity can be expressed as an intra-class retrieval operation. The fundamental idea behind their method is to convert raw polygons in R^3 space into a compact 2D description that retains the geometry information, and then perform the retrieval operation in R^2 space. This offers two advantages: 1) working in R^2 space is easier, and 2) the system can apply the existing 2D techniques. A 3D model is first created to describe the selected class. Apart from the geometry, the model also includes any additional features that characterise the class (e.g., area annotation, landmarks). Additionally, the model has a regularly sampled mapping from R^3 to R^2 (UV parameterization) that can be used to construct the equivalent 2D description, the geometry image. Subsequently, a subdivision-based model is fitted onto all the objects of the class using a deformable model framework. The result is converted to a geometry image and wavelet decomposition is applied. The wavelet coefficients are stored for matching and retrieval purposes (Fig. 4).

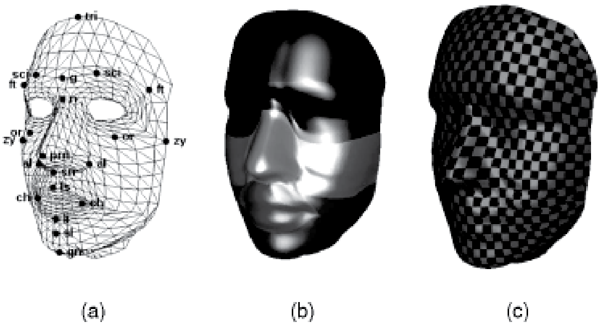


Fig. 4. (a) Anthropometric landmarks used, (b) segmentation into annotated areas, and (c) checkerboard texture to demonstrate parameterization (Passalis et al., 2007)

(Zaeri, 2011) investigated a new 3D face image acquisition and capturing system, where a test-bed for 3D face image feature characteristic and extraction was demonstrated. (Wong et al., 2007) proposed a multi-region face recognition algorithm for 3D face recognition. They identified the multiple sub-regions over a given range facial image and extracted summation invariant features from each sub-region. For each sub-region and the corresponding summation invariant feature, a matching score was calculated. Then, a linear fusion method was developed to combine the matching scores of individual regions to arrive at a final matching score. (Samir et al., 2006) described the face surface using contour lines or iso-contours of the depth function while using the nose tip as a reference point for alignment. The face surface is represented as a 2D image (e.g., depth-map), and then a 2D image classification techniques are applied. This approach requires that the surfaces are aligned by the iterative closest point algorithm or by feature-based techniques. Then, the deformable parts of the face are detected and excluded from the matching stage or downgrade their contribution during matching. This, however, may lead to loss of information (e.g., excluding the lower part of the face) which is important for classification. A different approach is to use an active appearance model or in the general case, a 3D deformable model which may be fitted to the face surface. The difficulty in this case is in building a (usually linear) model that can capture all possible degrees of freedom hidden in facial expressions and fitting the model to the surface in hand.

The approach of (Mpiperis et al., 2007) relies on the assumption that the face is approximately isometric, which means that geodesic distances among points on the surface are preserved, and tries to establish an expression-invariant representation of the face. This technique does not have the disadvantages outlined in some other methods (loss of information and dealing with face variability). (Mpiperis et al., 2007) have considered the face surface as a 2D manifold embedded in the 3D Euclidean space, characterized by a Riemannian metric and described by intrinsic properties, namely geodesics (Figures 5 and 6).

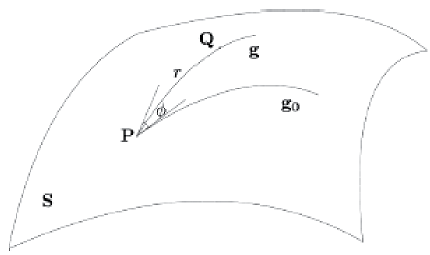


Fig. 5. Definition of geodesic distance r and polar angle φ of an arbitrary point Q. Geodesic path g is the minimum length curve connecting point Q and geodesic pole P. r is the length of g, while φ is the angle between g and a reference geodesic path g0 (Mpiperis et al., 2007)

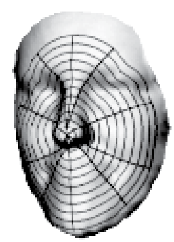


Fig. 6. Geodesic paths and circles defined over a face surface. The tip of the nose was selected as the geodesic pole (Mpiperis et al., 2007)

(Li et al., 2009) proposed a 3D face recognition approach using Harmonic Mapping and ABF++ as the mesh parameterization techniques. This approach represents the face surface in both local and global manners, which encodes the intrinsic attributes of surface in planar regions. Therefore, surface coarse registration and matching can be dealt with in a low dimensional space. The basic idea is to map 3D surface patches to a 2D parameterization domain and encode the shape and texture information of a 3D surface into a 2D image. Therefore, complex geometric processing can be analyzed and calculated in a low-dimensional space. The mean curvature to characterize the points of surface is employed. Then, both local shape description and global shape description with curvature texture are constructed to represent the surface. With the selected surface patches in local regions, Harmonic Mapping is used to construct the local shape description. Harmonic Mappings are the solutions to partial differential equations from the Dirichlet energy defined in Riemann manifolds. An example of the constructed local shape description at a feature point on a facial surface is shown in Fig. 7, while the global shape description is shown in Fig. 8. For the overall meshes of probe or gallery images, nonlinear parameterization ABF++ with free boundary, proposed by (Sheffer et al., 2005), is used to create global shape description.

The method presented by (Guo et al., 2010) is based on conformal geometric maps which does not need 3D models registration, and also maps 3D facial shape to a 2D domain which is a diffeomorphism through a global optimization. The 2D maps integrate geometric and appearance information and have the ability to describe the intrinsic shape of the 3D facial model, called Intrinsic Shape Description Maps (Fig. 9).

(Harguess & Aggarwal, 2009) presented a comparison of the use of the average-half-face to the use of the original full face with 6 different algorithms applied to two- and three-

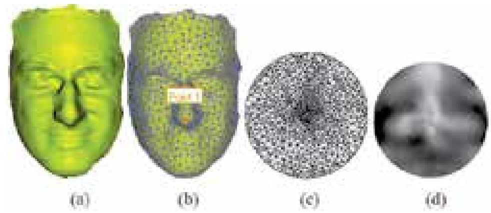


Fig. 7. Local shape description at point 1 for the face model A: (a) Original range image; (b) Triangle meshes in 3D space; (c) Planar meshes after Harmonic Mapping; (d) LSD with mean curvature texture (Li et al., 2009)

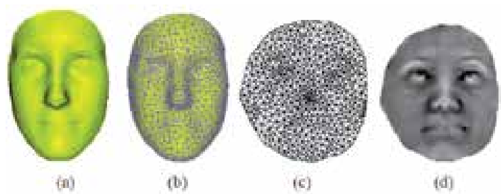


Fig. 8. Global shape description for the face model B: (a) Original range image; (b) Triangle meshes in 3D space; (c) Planar meshes after ABF++; (d) GSD with mean curvature texture (Li et al., 2009)

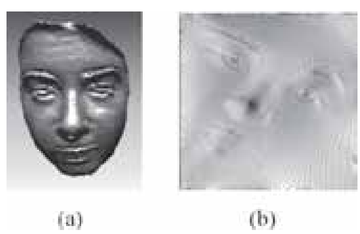


Fig. 9. Constrained conformal mapping result (a) original 3D model (b) the mapping result of (a) (Guo et al., 2010)

dimensional (2D and 3D) databases. The average-half-face is constructed from the full frontal face image in two steps; first the face image is centred and divided in half and then the two halves are averaged together (reversing the columns of one of the halves). The resulting average-half-face is then used as the input for face recognition algorithms. (Harguess & Aggarwal, 2009) compared the results using the following algorithms: eigenfaces, multi-linear principal components analysis (MPCA), MPCA with linear discriminant analysis (MPCALDA), Fisherfaces (LDA), independent component analysis (ICA), and support vector machines (SVM).

4. Active appearance model approach

Many researchers have used the active appearance model (AAM) (Cootes et al., 2001) in modelling 3D face images. The AAM is a generative and parametric model that allows representation of a variety of shapes and appearances of human faces. It uses the basis vectors that are obtained by applying principal component analysis (PCA) to the input images and tries to find the maximum amount of variance. Although AAM is simple and fast, fitting it to an input image is not an easy task because it requires nonlinear optimization that finds a set of suitable parameters simultaneously, and its computation is basically conducted in an iterative manner. Usually, the fitting is performed by a variety of standard nonlinear optimization methods.

(Abboud et al., 2004) proposed the facial expression synthesis and recognition system by face model with AAM. After extracting appearance parameters of AAM for recognition, they recognized facial expression in Euclidian and Mahalanobis space of these parameters. Also, (Abboud & Davoine, 2004) proposed a bilinear factorization expression classifier for the recognition and compared it to linear discriminant analysis (LDA). Their results showed that the bilinear factorization is useful when only a few number of training samples are available. (Ishikawa et al., 2004) used AAM for tracking around the eye region and recognized the direction of gaze.

(Matthews et al., 2004) suggested that the performance of an AAM built with single-person data is better than that of AAM built with multiple person data for the pose and illumination problems. (Xiao et al., 2004) employed 3D shapes in the AAM in order to solve the pose problem and used a nonrigid structure-from-motion algorithm for computing this 3D shape from 2D images. The 3D shape provides the constraints on the 2D shape, which can be more deformable, and these constraints make fitting more reliable. (Hu et al., 2004) proposed another extension of a 2D + 3D AAM fitting algorithm, called the multiview AAM fitting algorithm. It fits a single 2D + 3D AAM to multiple view images obtained simultaneously from multiple affine cameras. (Mittrapiyanuruk et al., 2004) proposed the use of stereo vision to construct a 3D shape and estimate the 3D pose of a rigid object using AAM. (Cootes et al., 2002) proposed using several face models to fit an input image. They estimated the pose of an input face image by a regression technique and then fitted the input face image to the face model closest to the estimated pose. However, their approach requires pose estimation, which is another difficult problem, since the pose estimation might cause an incorrect result when the appearance of the test face image is slightly different from the training images due to different lighting conditions or different facial expressions. (Sung & Kim, 2008) proposed an extension of the 2D + 3D AAM to a viewbased approach for pose-robust face tracking and facial expressions. They used the PCA with missing data (PCAMD) technique to obtain the 2-D and 3-D shape basis vectors since some face models have missing data. Then, they developed an appropriate model selection for the input face image. This model selection method uses the pose angle that is estimated from the 2D + 3D AAM directly.

(Park et al., 2010) proposed a method for aging modelling in the 3D domain. Facial aging is a complex process that affects both the shape and texture (e.g., skin tone or wrinkles) of a face. This aging process also appears in different manifestations in different age groups. While facial aging is mostly represented by facial growth in younger age groups (e.g., ≤ 18 years old), it is mostly represented by relatively large texture changes and minor shape changes (e.g., due to change of weight or stiffness of skin) in older age groups (e.g., >18). Therefore, an age correction scheme needs to be able to compensate for both types of aging processes. (Park et al., 2010) have shown how to build a 3D aging model given a 2D face aging database. Further, they have compared three different modelling methods, namely, shape modelling only, separate shape and texture modelling, and combined shape and texture modelling (e.g., applying second level PCA to remove the correlation between shape and texture after concatenating the two types of feature vectors).

5. Filtering based approach

(Yang et al., 2008) applied the canonical correlation analysis (CCA) to learn the mapping between the 2D face image and 3D face data. The proposed method consists of two phases. In the learning phase, given the 2D-3D face data pairs of the subjects for training, PCA is first applied on both 2D face image and 3D face data to avoid the curse of dimensionality and reduce noise. Then the CCA regression is performed between the features of 2D-3D in the previous PCA subspaces. In the recognition phase, given an input 2D face image as a probe, the correlation between the probe and the gallery is computed as matching score using the learnt regression. Furthermore, to simplify the mapping between 2D face image and 3D face data, a patch based strategy is proposed to boost the accuracy of matching. (Huang et al., 2010) presented an asymmetric 3D-2D face recognition method, that uses textured 3D face image for enrolment while performs automatic identification using only 2D facial images. The goal is to limit the use of 3D data to where it really helps to improve face recognition accuracy. The proposed method contains two separate matching steps: Sparse Representation Classifier (SRC) which is applied to 2D-2D matching, and CCA which is exploited to learn the mapping between range local binary pattern (LBP) faces (3D) and texture LBP faces (2D). Both matching scores are combined for the final decision.
(Günlü & Bilge, 2010) divided 3D faces into smaller voxel regions and applied 3D transformation to extract features from these voxel regions, as shown in Fig. 10. The number of features selected from each voxel region is not constant and depends on their discrimination.

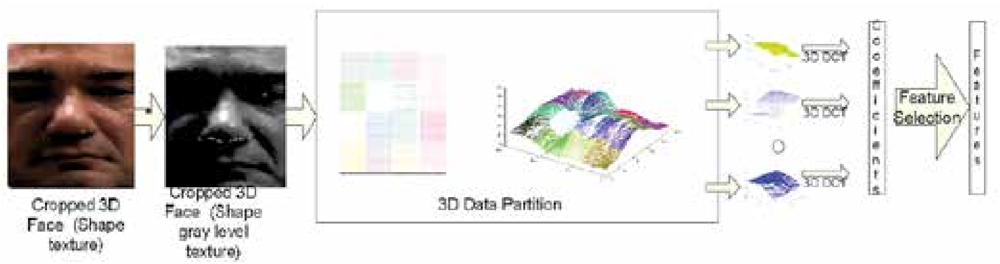


Fig. 10. Proposed method by (Günlü & Bilge, 2010)

(Dahm & Gao, 2010) presented a novel face recognition approach that implements cross-dimensional comparison to solve the issue of pose invariance. The approach implements a Gabor representation during comparison to allow for variations in texture, illumination, expression and pose. Kernel scaling is used to reduce comparison time during the branching search, which determines the facial pose of input images. This approach creates 2D rendered views of the 3D model from different angles, which are then compared against the 2D probe. Each rendered view is created by deforming the 3D model's texture with the 3D shape information, as shown in Fig. 11.
(Wang et al., 2010) proposed another scheme for 3D face recognition that passes through different stages. They used iterative closet point to align all 3D face images with the first person. Then a region defined by a sphere of radius 100 mm centred at the nose tip was cropped to construct the depth image. The Gabor filter was used to capture the useful local structure of the depth images.

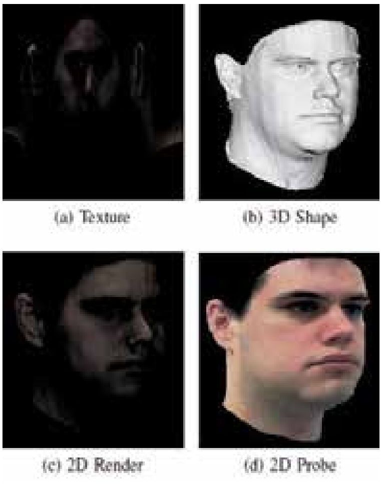


Fig. 11. Comparison of 3D and 2D representation. Contrast and Brightness have been increased on texture and render for viewing (Dahm & Gao, 2010)

Another approach that deals with 3D face recognition was presented by (Cook et al., 2007), where they used multi-scale techniques to partition the information contained in the frequency domain prior to dimensionality reduction. In this manner, it is possible to increase the information available for classification and, hence, increase the discriminative performance of both Eigenfaces and Fisherfaces techniques, which were used for dimensionality reduction. They have used the Gabor filters as a partitioning scheme, and compared their results against the discrete cosine transform and the discrete wavelet transform.

6. Statistical approach

(Rama & Tarrés, 2005) have presented Partial Principal Component Analysis (P^2CA) for 3D face recognition. The main advantage in comparison with the model-based approaches is its low computational complexity since P^2CA does not require any fitting process. However, one of the main problems of their work is the enrolment of new persons in the database (gallery set) since a total of five different images are needed for getting the 180º texture map. Recently, they presented a work that automatically creates 180º texture maps from only two images (frontal and profile views) (Rama & Tarrés, 2007). Nevertheless, this work has also another constraint; it needs a normalization (registration) process for both eyes where they should be perfectly aligned at a fixed distance. Thus, errors in the registration of the profile view lead to noisy areas of the reconstructed 180º images (Fig. 12).

(Gupta et al., 2007) presented a systematic procedure for selecting facial fiducial points associated with diverse structural characteristics of a human face. They have identified such characteristics from the existing literature on anthropometric facial proportions. Also, they have presented effective face recognition algorithms, which employ Euclidean/geodesic distances between these anthropometric fiducial points as features along with linear discriminant analysis (LDA) classifiers. They have demonstrated how the choice of facial fiducial points critically affects the performance of 3D face recognition algorithms that employ distances between them as features. Anthropometry is the branch of science that

Fig. 12. (a) Set of images used for the creation of the training data; (b) Example of a 180º texture training image (Rama & Tarrés, 2007)

deals with the quantitative description of physical characteristics of the human body. Anthropometric cranio-facial proportions are ratios of pairs of straight-line and/or along-the-surface distances between specific cranial and facial fiducial points (Fig. 13).

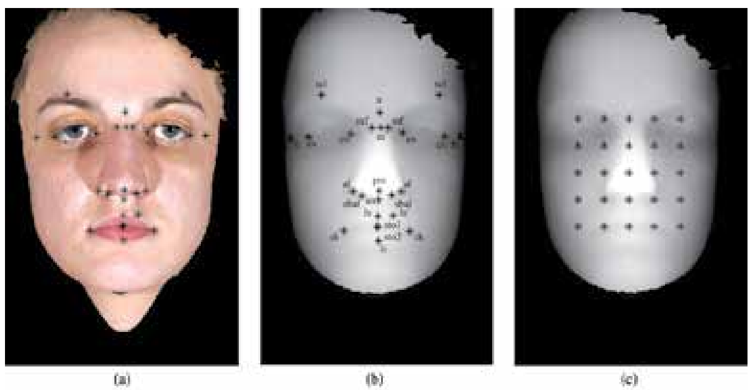


Fig. 13. The figure depicts (a) 25 anthropometric fiducial points on a texture image; (b) 25 anthropometric fiducial points on a range image; (c) 25 arbitrary equally spaced points overlaid on the main facial features (Gupta et al., 2007)

(Ming et al., 2010) proposed algorithm for 3D-based face recognition by representing the facial surface, by what is called a Bending Invariant (BI), invariant to isometric deformations resulting from expressions and postures. In order to encode relationships in neighbouring mesh nodes, Gaussian-Hermite moments are used for the obtained geometric invariant, which provide rich representation, due to their mathematical orthogonality and effectiveness in characterizing local details of the signal. Then, the signature images are decomposed into their principle components based on Spectral Regression Kernel Discriminate Analysis (SRKDA) resulting in a huge time saving.

7. Local binary patterns

In (Zhou et al., 2010), Local Binary Patterns (LBP) method was used to represent 3D face images. The Local Binary Pattern (LBP) method describes the local texture pattern with a binary code. It is built by thresholding a neighbourhood P with radius R (typically denoting the 8 surrounding pixels) by the gray value g of its centre c . Also, (Ming et al., 2010) proposed a framework for 3D face recognition that is based on the 3D Local Binary Patterns

(3D LBP). In the feature extraction stage, 3D LBP is adopted to describe the intrinsic geometric information, negating the effect of expression variations effectively. 3D LBP encodes relationships in neighbouring mesh nodes and own more potential power to describe the structure of faces than individual points. In learning stage, Spectral Regression is adopted to learn principle components from each 3D facial image. With dimensional reduction based on Spectral Regression, more useful and significant features can be produced for a face, resulting in a huge saving in computational cost. Finally, face recognition is achieved using Nearest Neighbour Classifiers.

8. Other 3D face recognition approaches

In order to enhance robustness to expression variations, a procedure for 3D face recognition based on the depth image and Speeded-Up Robust Features (SURF) Operator was proposed by (Yunqi et al., 2010). First, they have applied the Fisher Linear Discriminant (FLD) method on the depth image to perform coarse recognition to catch the highly ranked 3D faces. On the basis of this step, they extracted the SURF features of the 2D gray images that are corresponding only to those highly ranked 3D faces, to carry out the refined recognition. SURF algorithm was first proposed by (Bay et al., 2008). At present, SURF has been applied to image registration, camera calibration and object recognition. Furthermore, (Kim & Dahyot, 2008) presented another approach for 3D face recognition using SVM and SURF Descriptor.

On the other hand, (Wang et al., 2009) used a spherical harmonic representation with the morphable model for 2D face recognition. The method uses a 2D image to build a 3D model for the gallery, based on a 3D statistical morphable model. Also, (Biswas et al., 2009) proposed a method for albedo estimation for face recognition using two-dimensional images. However, they assumed that the image did not contain shadows. (Zhou et al., 2008) used nearest-subspace patch matching to warp near frontal face images to frontal and project this face image into a pre-trained low-dimensional illumination subspace. Their method requires training of patches in many different illumination conditions.

9. 3D face fitting

A 3D Morphable Model (3DMM) consists of a parameterized generative 3D shape, and a parameterized albedo model together with an associated probability density on the model coefficients. Together with projection and illumination parameters, a rendering of the face can be generated. Given a face image, one can also solve the inverse problem of finding the coefficients which most likely generated the image. Identification and manipulation tasks in coefficient space are trivial, because the generating factors (light, pose, camera, and identity) have been separated. Solving this inverse problem is termed "model fitting", and was introduced for faces by (Blanz & Vetter, 1999). A similar method has also been applied to stereo data (Amberg et al., 2007) and 3D scans (Amberg et al., 2008).

A 3D deformation modelling scheme was proposed by (Lu & Jain, 2008) to handle the expression variations. They proposed a facial surface modelling and matching scheme to match 2.5D facial scans in the presence of both nonrigid deformations and pose changes (multiview) to a stored 3D face model with neutral expression.

They collected data for learning 3D facial deformations from only a small group of subjects, called the control group. Each subject in the control group provides a scan with neutral expression and several scans with nonneutral expressions. The deformations (between neutral scan and nonneutral scans) learned from the control group are transferred to and synthesized for all the 3D neutral face models in the gallery, yielding deformed templates with synthesized expressions (Fig. 14). For each subject in the gallery, deformable models are built based on the deformed templates. In order to learn deformation from the control group, a set of fiducial landmarks is needed. Besides the fiducial facial landmarks such as eye and mouth corners, landmarks in the facial area with little texture, for example, cheeks are extracted in order to model the 3D surface movement due to expression changes. A hierarchical geodesic-based resampling scheme constrained by fiducial landmarks is designed to derive a new landmark-based surface representation for establishing correspondence across expressions and subjects.

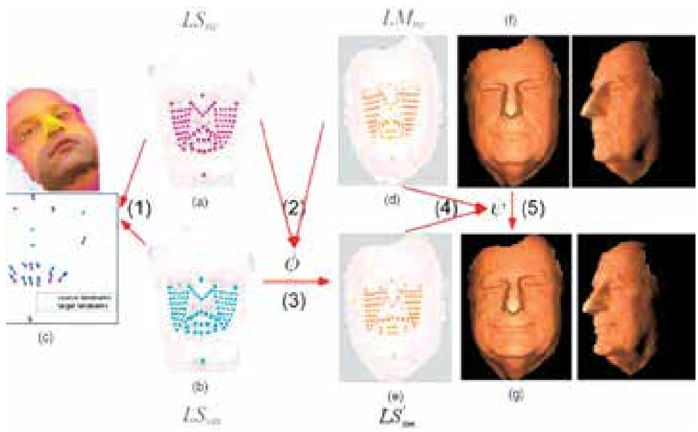


Fig. 14. Deformation transfer and synthesis (Lu & Jain, 2008)

(Wang et al., 2009) proposed an improved algorithm aiming at recognizing faces of different poses when each face class has only one frontal training sample. For each sample, a 3D face is constructed by using 3DMM. The shape and texture parameters of 3DMM are recovered by fitting the model to the 2D face sample which is a non-linear optimization problem. The virtual faces of different views are generated from the 3DMM to assist face recognition. They have located 88 sparse points from the 2D face sample by automatic face fitting and used their correspondence in the 3D face as shape constraint (Fig. 15).

(Daniyal et al., 2009) proposed a compact face signature for 3D face recognition that is extracted without prior knowledge of scale, pose, orientation or texture. The automatic extraction of the face signature is based on fitting a trained Point Distribution Model (PDM) (Nair & Cavallaro, 2007). First, a facial representation based on testing extensive sets of manually selected landmarks is chosen. Next, a PDM is trained to identify the selected set of landmarks (Fig. 16). The recognition algorithm represents the geometry of the face by a set of Inter-Landmark Distances (ILDs) between the selected landmarks. These distances are then compressed using PCA and projected onto the classification space using LDA. The classification of a probe face is finally achieved by projecting the probe onto the LDA-subspace and using the nearest mean classifier.

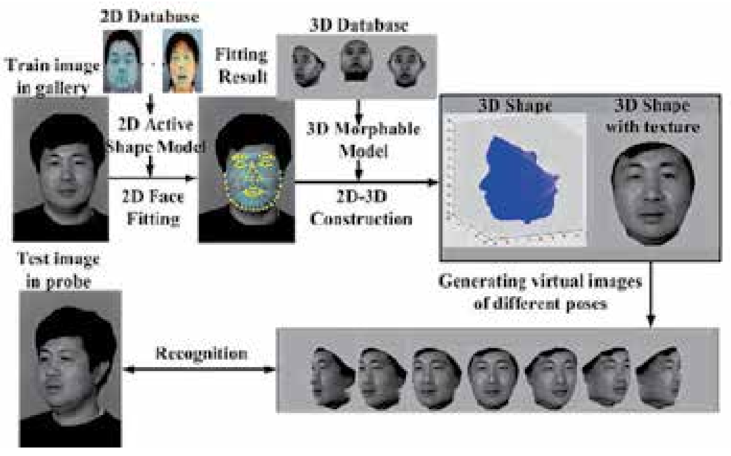


Fig. 15. Algorithm overview (Wang et al., 2009)

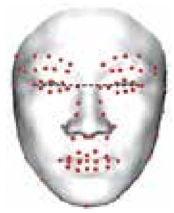


Fig. 16. Sample face scan showing the annotated landmarks and the scaling distance *dS* (dotted line) used in (Daniyal et al., 2009)

(Paysan et al., 2009) proposed a generative 3D shape and texture model, the Basel Face Model (BFM). The model construction passes through four steps: 3D face scanning, Registration, Texture Extraction and Inpainting, and Model. The model is based on parameterizing the faces using triangular meshes. A face is then represented by two dimensional vectors: shape and texture, constructing two independent Linear Models. Finally, a Gaussian distribution is fit to the data using PCA (Fig. 17).

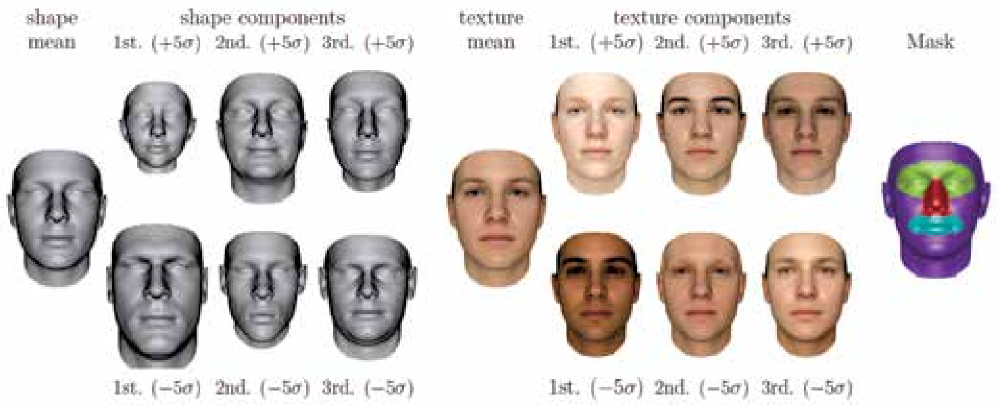


Fig. 17. The mean together with the first three principle components of the shape (left) and texture (right) PCA model (Paysan et al., 2009)

(Toderici et al., 2010) proposed a face recognition method which utilizes 3D face data for enrolment, while it requires only 2D data for authentication. During enrolment, 2D+3D data (2D texture plus 3D shape) is used to build subject-specific annotated 3D models. First, an Annotated Face Model (AFM) is fitted to the raw 2D+3D data using a subdivision based deformable framework. Then, a geometry image representation is extracted using the UV parameterization of the model. In the authentication phase, a single 2D image is used as the input to map the subject-specific 3D AFM. After that, an Analytical Skin Reflectance Model (ASRM) is applied to the gallery AFM in order to transfer the lighting from the probe to the texture in the gallery.

10. Face recognition in video

Face recognition in video has gained wide attention as a covert method for surveillance to enhance security in a variety of application domains (e.g., airports). A video contains temporal information (e.g., movements of facial features) as well as multiple instances of a face, so it is expected to lead to a better face recognition performance compared to still face images. However, faces appearing in a video have substantial variations in pose and lighting. These pose and lighting variations can be effectively modelled using 3D face models (Yin et al., 2006). Given the trajectories of facial feature movement, face recognition is performed based on the similarities of the trajectories. The trajectories can also be captured as nonlinear manifolds and the distance between clusters of faces in the feature space establishes the identity associated with the face. Production of 3D faces from video can be performed using morphable models, stereography, or structure from motion (SFM).

(Park et al., 2005) proposed a face recognition system that identifies faces in a video using 3D face model. Ten video files were recorded for ten subjects under four different lighting conditions at various poses with yaw and pitch motion. Recognition using multiple images and temporal cue was explored and majority voting and score sum were used to fuse the recognition result from multiple frames. To use temporal cues for the recognition, a LDA based classifier was used. After the face pose in a video was estimated, frames of different poses under specific lighting condition and specific order were extracted to form a probe sequence.

(Von Duhn et al., 2007) designed a 3D face analyzer using regular CCTV videos. They used a three view tracking approach to build 3D face models over time. The proposed system detects, tracks and estimates the facial features. For the tracking, an Active Appearance Model approach is adapted to decrease the amount of manual work that must be done. After the tracking stage, a generic model is adapted to the different views of the face using a face adaptation algorithm, which includes two steps: feature point adaptation and non-feature point interpolation. Finally, the multiple views of models are combined to create an individualized face model. To track the facial motion under three different views, i.e., front view , side view, and angle view, predefined fiducial points are used.

Also, (Roy-Chowdhury & Xu, 2006) estimated the pose and lighting of face images contained in video frames and compared them against synthetic 3D face models exhibiting similar pose and lighting. However, the 3D face models were registered manually with the face image in the video. (Lee et al., 2003) proposed an appearance manifold based approach where each database or gallery image was matched against the appearance manifold obtained from the video. The manifolds were obtained from each sequence of pose variations. (Zhou et al., 2003) proposed to obtain statistical models from video using low level features (e.g., by PCA) contained in sample images. The matching was performed between a single frame and the video or between two video streams using the statistical models.

(Park et al., 2007) explored the adaptive use of multiple face matchers in order to enhance the performance of face recognition in video. To extract the dynamic information in video, the facial poses in various frames are explicitly estimated using Active Appearance Model and a Factorization based 3D face reconstruction technique. The motion blur is estimated using Discrete Cosine Transformation (DCT). The performance of the proposed system could be improved by dynamically fusing the matching results from multiple frames and multiple matchers.

Further, (Wang et al., 2004) have successfully developed a hierarchical framework for tracking high density 3D facial expression sequences captured from a structure-lighting imaging system. The work in (Chang et al., 2005), utilized six 3D model sequences for facial analysis and editing. The work was mainly for facial expression analysis. (Papatheodorou & Rueckert, 2004) evaluated a so-called 4D face recognition approach, which was, however, just the 3D static data plus texture, no temporal information was explored. (Li et al., 2003) reported a model fitting approach to generate facial identity surfaces through video sequences. The application of this model to face recognition relies on the quality of the tracked low resolution face model.

(Sun & Yin, 2008) proposed to use a Spatio-Temporal Hidden Markov Model (HMM) which incorporates 3D surface feature characterization to learn the spatial and temporal information of faces. They have created a face database including 606 3D model sequences with six prototypic expressions. To evaluate the usability of such data for face recognition, they applied a generic model to track the range model sequences and establish the correspondence of range model frames over time. After the tracking model labelling and LDA transformation, they trained two HMM models (S-HMM and T-HMM) for each subject to learn the spatial and temporal information of the 3D model sequence. The query sequence was classified based on the results of the two HMMs.

(Medioni et al., 2007) utilized synthetic stereo to model faces in a 3048 x 4560 video stream. By tracking the pose and location of the face, a synthetic stereo rig based upon the different poses between two frames is initialized. Multiple point clouds from different stereo pairs are created and integrated into a single model. (Russ et al., 2006) utilized a 3D PCA based approach for face recognition. The approach determines a correspondence that utilizes a reference face aligned via ICP to determine a unique vector input into PCA. The coefficients from PCA are used to determine the identity as in 2D PCA face recognition. (Kakadiaris et al., 2006) converted the 3D model into a depth map image for wavelet analysis. This approach performs well and does not utilize ICP as the basis for each match score computation, but does for the depth map production.

Moreover, (Boehnen & Flynn, 2008) presented an approach to combine multiple noisy low density 3D face models obtained from uncalibrated video into a higher resolution 3D model using SFM method. SFM is a method for producing 3D models from a calibrated or uncalibrated video stream utilizing equipment that is inexpensive and widely available. The approach first generates ten 3D face models (containing a few hundred vertices each) of each subject using 136 frames of video data in which the subject face moves in a range of approximately 15 degrees from frontal. By aligning, resampling, and merging these models, a new 3D face model containing over 50,000 points is produced. An ICP face matcher employing the entire face achieved a 75% rank one recognition rate.

Using a data set of varying facial expressions and lighting conditions, (Bowyer et al., 2006) reported an improvement in rank one recognition rate from 96.11% with two frames per subject to 100% with four frames per subject. In another study, (Thomas et al., 2007) observed that the recognition rate generally increases as the number of frames per subject increases, regardless of the type of camera being used. They also found that the optimal number of frames per subject is between 12 and 18, given the particular data sets used.

(Canavan et al., 2007) discussed that the 3D geometry of a rotating face can be embedded in the continuous intensity changes of an image stream, and therefore the recognition algorithm does not require an explicit 3D face model. Further, multiple video frames that capture the face at different pose angles can be combined to provide a more reliable and comprehensive 3D representation of the face than any single view image. Also, they have discussed that a video sequence of a face with different poses might help alleviate the adverse effect of lighting changes on recognition accuracy. For instance, a light source can cast shadows on a face, but at the same time, it also reveals the 3D curvatures of the face by creating sharp intensity contrasts (such as silhouette).

(Dornaika & Davoine, 2006) introduced a view- and texture-independent approach that exploits the temporal facial action parameters estimated by an appearance-based 3D face tracker. The facial expression recognition is carried out using learned dynamical models based on auto-regressive processes. These learned models can also be utilized for the synthesis and prediction tasks. In their study, they used the 3D face model *Candide* (Ahlberg, 2001). This 3D deformable wireframe model is given by the 3D coordinates of the vertices $\mathbf{P}_i$, $i = 1, \ldots, n$ where n is the number of vertices. Thus, the shape up to a global scale can be fully described by the $3n$-vector **g,** the concatenation of the 3D coordinates of all vertices $\mathbf{P}_i$. The vector **g** can be written as:

$$g = \overline{g} + S\tau_s + A\tau_a \tag{1}$$

where $\overline{g}$ is the standard shape of the model, and the columns of S and A are the shape and action units, respectively. Thus, the term $S\ \boldsymbol{\tau_s}$ accounts for shape variability (inter-person variability) while the term $A\ \boldsymbol{\tau_a}$ accounts for the facial action (intra-person variability).

11. Conclusion

In this chapter, we have presented a study on the most recent advancements in 3D face recognition field. Despite the huge developments made in this field, there are still some problems and issues which need to be resolved.

Due to the computational complexity, fussy pre-treatment, and expensive equipment, 3D technology is still not used widely in practical applications. To acquire an accurate 3D face data, some very costly equipment must be used, such as 3D laser scan or stereo camera system. Also, they are still not as stable and efficient as 2D cameras, and for some cases like the stereo camera system, calibration is needed before use. Moreover, they take a longer time to acquire (or reconstruct) when compared to the 2D camera. Further, 3D data require much more storage space. Other challenges include feature points allocation (this is still a debatable topic) that is also sensitive to the quality of data. Sampling density of the facial surface and accuracy of the depth, are among the issues that require more investigations. Furthermore, no standard testing protocol is available to compare between different 3D face recognition systems.

On the other hand, in video-based face recognition, experiments have shown that multi-frame fusion is an effective method to improve the recognition rate. The performance gain is probably related to the use of 3D face geometry embedded in video sequences. However, it is not clear how the inter-frame variation has contributed to the observed performance increase. Will the multi-frame fusion work for videos of strong shadows? How many frames are necessary for maximizing the recognition rate without incurring a heavy computational cost? To address these issues, more exploration is needed from the research community.

12. Acknowledgments

The author would like to acknowledge and thank Kuwait Foundation for the Advancement of Sciences (KFAS) for financially supporting this work.

13. References

Abboud, B. & Davoine, F. (2004). Appearance factorization for facial expression recognition and synthesis, *Proceedings of Int. Conf. Pattern Recog.*, pp. 163–166, 2004

Abboud, B.; Davoine, F. & Dang, M. (2004). Facial expression recognition and synthesis based on an appearance model. *Signal Process. Image Commun.*, Vol. 19, No. 8, 2004, pp. 723–740

Ahlberg, J. (2001). CANDIDE-3 - an updated parametrized face, *Tech. Rep.* LiTH-ISY-R-2326, Dept of Electrical Engineering, Link¨oping University, Sweden

Amberg, B.; Knothe, R. & Vetter, T. (2008). Expression invariant 3D face recognition with a morphable model, *Proceedings of FG'08*, 2008

Amberg, B.; Romdhani, S.; Fitzgibbon, A.; Blake, A. & Vetter, T. (2007). Accurate surface extraction using model based stereo, *Proceedings of ICCV*, 2007

Bay, H.; Tuytelaars, T. & Van Gool. I. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 2008, 110346-359

Biswas, S.; Aggarwal, G. & Chellappa, R. (2009). Robust estimation of albedo for illumination-invariant matching and shape recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, 2009, pp. 884–899

Blanz, V. & Vetter, T. (1999). A morphable model for the synthesis of 3D faces, *Proceedings of SIGGRAPH*, 1999

Boehnen, C. & Flynn, P. J. (2008). Increased Resolution 3D Face Modeling and Recognition From Multiple Low Resolution Structure From Motion Models, *Proceedings of IEEE*, 2008

Bowyer, K. W.; Chang, K.; Flynn, P. J. & Chen, X. (2006). Face recognition using 2-D, 3-D and Infrared: Is multimodal better than multisample?, *Proceedings of IEEE*, Vol. 94, No. 11, pp. 2000-2012, 2006

Bronstein, A. M.; Bronstein, M. M. & Kimmel, R. (2007). Expression-invariant representations of faces. *IEEE Trans. on PAMI*, 2007, pp. 1042–1053

Canavan, S. J.; Kozak, M. P.; Zhang, Y.; Sullins, J. R.; Shreve, M. A. & Goldgof, D. B. (2007). Face Recognition by Multi-Frame Fusion of Rotating Heads in Videos, *Proceedings of IEEE*, 2007

Chang, Y.; Vieira, M.; Turk, M. & Velho, L. (2005). Automatic 3d facial expression analysis in videos, *Proceedings of ICCV Workshop on Analysis and Modeling of Faces and Gestures*, 2005

Cook, J.; Chandran, V. & Fookes, C. (2006). 3D face recognition using log-gabor templates, *Proceedings of the 17th British Machine Vision Conference,* 2006

Cook, J.; Chandran, V. & Sridharan, S. (2007). Multiscale Representation For 3-D Face Recognition. *IEEE Transactions on Information Forensics and Security*, Vol. 2, No. 3, September 2007

Cootes, T. F.; Edwards, G. J. & Taylor, C. J. (2001). Active appearance models, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 23, No. 6, Jun. 2001, pp. 681–685

Cootes, T. F.; Wheeler, G. V.; Walker, K. N. & Taylor, C. J. (2002). Viewbased active appearance models. *Image Vis. Comput.*, Vol. 20, No. 9, Aug. 2002, pp. 657–664

Dahm, N. & Gao, Y. (2010). A Novel Pose Invariant Face Recognition Approach Using A 2D-3D Searching Strategy, *Proceedings of Int'l Conf. on Pattern Recognition*, 2010

Daniyal, F.; Nair, P. & Cavallaro, A. (2009). Compact signatures for 3D face recognition under varying expressions, *Proceedings of Advanced Video and Signal Based Surveillance*, 2009

Dornaika, F. & Davoine, F. (2006). Facial Expression Recognition using Auto-regressive Models, *Proceedings of 18th International Conf. on Pattern Recognition*, 2006

Drira, H.; Amor, B. B.; Daoudi, M. & Srivastava, A. (2009). A riemannian analysis of 3d nose shapes for partial human biometrics, *Proceedings of ICCV*, Vol. 1, No. 1, pp. 1–8, 2009

Duhn, S. V.; Yin, L.; Ko, M. J. & Hung, T. (2007). Multiple-View Face Tracking For Modeling and Analysis Based On Non-Cooperative Video Imagery, *Proceedings of IEEE*, 2007

Elyan, E. & Ugail, H. (2009). Automatic 3D Face Recognition Using Fourier Descriptors, *International Conference on CyberWorlds*, 2009

Faltemier, T.; Bowyer, K.W. & Flynn, P.J. (2008). A region ensemble for 3D face recognition. *IEEE Trans. on Information Forensics and Security,* Vol. 3, No. 1, 2008, pp. 62-73

Guo, Z.; Zhang, Y.; Lin, Z. & Feng, D. (2009). A Method Based on Geometric Invariant Feature for 3D Face Recognition, *Proceedings of Fifth International Conference on Image and Graphics*, 2009

Guo, Z.; Zhang, Y.; Xia, Y.; Lin, Z. & Feng, D. (2010). 3D Face Representation And Recognition By Intrinsic Shape Description Maps, *Proceedings of ICASSP*, 2010

Gupta, S.; Aggarwal, J. K.; Markey, M. K. & Bovik, A. C. (2007). 3D Face Recognition Founded on the Structural Diversity of Human Faces, *Proceedings of IEEE,* 2007

Günlü, G. & Bilge, H. S. (2010). 3D Face Decomposition and Region Selection against Expression Variations, *Proceedings of Int'l Conference on Pattern Recognition*, 2010

Harguess, J. & Aggarwal, J. K. (2009). A Case for the Average-Half-Face in 2D and 3D for Face Recognition, *Proceedings of IEEE*, 2009

Hu, C.; Xiao, J.; Matthews, I.; Baker, S.; Cohn, J. & Kanade, T. (2004). Fitting a single active appearance model simultaneously to multiple images, *Proceedings of Brit. Mach. Vis. Conf.*, 2004

Huang, D.; Ardabilian, M.; Wang, Y. & Chen, L. (2010). Automatic Asymmetric 3D-2D Face Recognition, *International Conference on Pattern Recognition*, 2010

Ishikawa, T.; Baker, S.; Matthews, I. & Kanade, T. (2004). Passive driver gaze tracking with active appearance models, *Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep.* CMU-RI-TR-04-08, Feb. 2004

Kakadiaris, I. A.; Passalis, G.; Toderici, G.; Murtuza, N.; Lu, Y.; Karampatziakis, N. & Theoharis. T. (2007). Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. *IEEE Trans. on PAMI*, Vol. 29, No. 4, 2007, pp. 640–649

Kakadiaris, I.; Passalis, G.; Toderici, G.; Murtuza, N. & Theoharis, T. (2006). 3D face recognition, *Proceedings of the British Machine Vision Conference*, pp. 200-208, 2006

Kim, D. & Dahyot, R. (2008). Face components detection using SURF descriptors and SVMs. *Proceedings of Int'l Machine Vision and Image Processing Conf.*, pp.51-56, 2008

Lee, K. C.; Ho, J.; Yang, M. H. & Kriegman, D. (2003). Video- Based Face Recognition using probabilistic appearance manifolds, *Proceedings of Intl. Conf. on Computer Vision and Pattern Recognition*, 2003

Li, W.; Yin, Z.; Wu, J. & Xiong, Y. (2009). 3D Face Recognition Based on Local/Global Shape Description, *Proceedings of Int'l Conf. on Information Technology and Computer Science*, 2009

Li, Y.; Gong, S. & Liddell, H. (2003). Constructing facial identity surfaces for recognition. *Int'l Journal of Comp. Vision*, Vol. 53, No. 1, 2003, pp. 71–92

Lu, X. & Jain, A. K. (2008). Deformation Modeling For Robust 3D Face Matching. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 30, No. 8, August 2008, pp. 1346-1356

Matthews, I.; Baker, S. & Gross, R. (2004). Generic vs. person specific active appearance models, *Proceedings of Brit. Mach. Vis. Conf.*, pp. 1080–1093, Sep. 2004

Maurer, T.; Guigonis, D.; Maslov, I.; Pesenti, B.; Tsaregorodtsev, A.; West, D.; Medioni, G. & Geometrix, I. (2005). Performance of Geometrix Active ID 3D Face Recognition Engine on the FRGC Data, *Proceedings of IEEE CVPR*, pp.154-154, 2005

McCool, C.; Chandran, V.; Sridharan, S. & Fookes, C. (2008). 3D face verification using a free-parts approach. *Pattern Recogn. Lett.*, Vol. 29, No. 9, 2008

Medioni, G.; Fidaleo, D.; Choi, J.; Zhang, L.; Kuo, C.-H. & Kim, K. (2007). Recognition of Non-Cooperative Individuals at a Distance with 3D Face Modeling, *Proceedings of IEEE Workshop on Automatic Identification Advanced Technologies*, 2007

Ming, Y.; Ruan, Q. & Ni, R. (2010). Learning Effective Features For 3D Face Recognition, *Proceedings of 17th International Conference On Image Processing*, September 26-29, 2010

Ming, Y.; Ruan, Q.; Wang, X. & Mu, M. (2010). Robust 3D Face Recognition using Learn Correlative Features, *Proceedings of ICSP*, 2010

Mittrapiyanuruk, P.; DeSouza, G. N. & Kak, A. C. (2004). Calculating the 3D-pose of rigid objects using active appearance models, *Proceedings of IEEE Int. Conf. Robot. Autom.*, Vol. 5, pp. 5147–5152, 2004

Mpiperis, I.; Malassiotis, S. & Strintzis, M. G. (2007). 3-D Face Recognition With the Geodesic Polar Representation. *IEEE Transactions on Information Forensics and Security*, Vol. 2, No. 3, Sept 2007

Nair, P. & Cavallaro, A. (2007). Region segmentation and feature point extraction on 3D faces using a point distribution model, *Proceedings of IEEE Intl. Conf. on Image Processing*, Vol. 3, pp. 85–88, Texas, USA, Sept, 2007

Papatheodorou, T. & Rueckert, D. (2004). Evaluation of automatic 4d face recognition using surface and texture registration, *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2004

Park, U.; Chen, H. & Jain, A. K. (2005). 3D Model-Assisted Face Recognition in Video, *Proceedings of the Second Canadian Conference on Computer and Robot Vision* , 2005

Park, U.; Jain, A. K. & Ross, A. (2007). Face Recognition in Video: Adaptive Fusion of Multiple Matchers, *Proceedings of IEEE*, 2007

Park, U.; Tong, Y. & Jain, A. K. (2010). Age-Invariant Face Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 32, No. 5, May 2010, pp. 947-954

Passalis, G.; Kakadiaris, I. A. & Theoharis, T. (2007). Intraclass Retrieval of Nonrigid 3D Objects: Application To Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 2, February 2007

Paysan, P.; Knothe, R.; Amberg, B.; Romdhani, S. & Vetter, T. (2009). A 3D Face Model for Pose and Illumination Invariant Face Recognition, *Proceedings of Advanced Video and Signal Based Surveillance*, 2009

Rama, A. & Tarrés, F. (2005). P2CA: A new face recognition scheme combining 2D and 3D information, *Proceedings of IEEE International Conference on Image Processing*, Genoa, Italy, Sept 2005

Rama, A. & Tarrés, F. (2007). Face Recognition Using A Fast Model Synthesis From A Profile And A Frontal View, *Proceedings of IEEE, ICIP,* 2007

Riccio, D. & Dugelay, J. L. (2007). Geometric invariants for 2d/3d face recognition. *Pattern Recognition Letters*, Vol. 28, pp. 1907– 1914

Roy-Chowdhury, A. & Xu, Y. (2006). Pose and Illumination Invariant Face Recognition Using Video Sequences. *Face Biometrics for Personal Identification: Multi-Sensory Multi-Modal Systems*, Springer-Verlag, 2006, pp. 9-25

Russ, T.; Boehnen, C. & Peters, T. (2006). 3D Face Recognition Using 3D Alignment for PCA, Computer Vision and Pattern Recognition, New York, 2006, pp. 1391-1398

Samir, C.; Srivastava, A. & Daoudi, M. (2006). Three-dimensional face recognition using shapes of facial curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 28, No. 11, Nov. 2006, pp. 1858–1863

Sheffer, A.; et al. (2005). ABF++: fast and robust angle based flattening. *ACM Transactions on Graphics*, Vol. 24, No. 2, Apr 2005, pp. 311-330

Song, Y.; Wang, W. & Chen, Y. (2009). Research on 3D Face Recognition Algorithm, *First International Workshop on Education Technology and Computer Science*, 2009

Sun, Y. & Yin, L. (2008). 3D Spatio-Temporal Face Recognition Using Dynamic Range Model Sequences, *Proceedings of IEEE*, 2008

Sung, J. & Kim , D. (2008). Pose-Robust Facial Expression Recognition Using View-Based 2d + 3d AAM. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 38, No. 4, July 2008, pp.852- 866

Szeptycki, P.; Ardabilian, M.; Chen, L.; Zeng, W.; Gu, D. & Samaras, D. (2010). Partial face biometry using shape decomposition on 2D conformal maps of faces, *Proceedings of International Conference on Pattern Recognition*, 2010

Thomas, D.; Bowyer, K. W. & Flynn, P. J. (2007). Multi-frame approaches to improve face recognition, *Proceedings of IEEE Workshop on Motion and Video Computing*, pp. 19-19, Austin, Texas, 2007

Tin, M. M. M. & Sein, M. M. (2009). Multi Triangle Based Automatic Face Recognition System By Using 3d Geometric Face Feature, *International Instrumentation And Measurement Technology Conference*, Singapore, May 5-7, 2009

Toderici, G.; Passalis, G.; Zafeiriou, S.; Tzimiropoulos, G.; Petrou, M.; Theoharis, T. & Kakadiaris, I.A. (2010). Bidirectional relighting for 3D-aided 2D Face Recognition, *Proceedings of IEEE*, 2010

Wang, L.; Ding, L.; Ding, X. & Fang, C. (2009). Improved 3d Assisted Pose-Invariant Face Recognition, *Proceedings of IEEE ICASSP*, 2009

Wang, S.; Wang, Y.; Jin, M.; Gu, X. & Samaras, D. (2006). 3d surface matching and recognition using conformal geometry, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2453– 2460, 2006

Wang, X.; Ruan, Q. & Ming, Y. (2010). A New Scheme for 3D Face Recognition, *Proceedings of ICSP*, 2010

Wang, Y.; Huang, X.; Lee, C.; Zhang, S.; Li, Z.; Samaras, D.; Metaxas, D.; Elgammal, A. & Huang, P. (2004). High resolution acquisition, learning and transfer of dynamic 3d facial expressions, *Proceedings of EUROGRAPHICS*, 2004

Wang, Y.; Zhang, L.; Liu, Z.; Hua, G.; Wen, Z.; Zhang, Z. & Samaras, D. (2009). Face relighting from a single image under arbitrary unknown lighting conditions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 11, Nov. 2009, pp. 1968– 1984

Wong, K.-C.; Lin, W.-Y.; Hu, Y. H.; Boston, N. & Zhang, X. (2007). Optimal Linear Combination Of Facial Regions For Improving Identification Performance. *IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics,* Vol. 37, No. 5, October 2007, pp.

Xiao, J.; Baker, S.; Matthews, I. & Kanade, T. (2004). Real-time combined 2D + 3D active appearance models, *Proceedings of Conf. Comp. Vis. Pattern Recog.*, pp. 535–542, 2004

Yang, W.; Yi, D.; Lei, Z.; Sang, J. & Li, S. Z. (2008). 2D-3D Face Matching using CCA, *Proceedings of IEEE*, 2008

Yin, L.; Wei, X.; Sun, Y.; Wang, J. & Rosato, M. J. (2006). A 3D Facial Expression Database For Facial Behavior Research, *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition* , 2006

Yunqi, L.; Haibin, L. & Xutuan, J. (2010). 3D Face Recognition by SURF Operator Based on Depth Image, *Proceedings of IEEE*, 2010

Zaeri, N. (2011). Feature extraction for 3D face recognition system, *Proceedings of IEEE*, 2011

Zhou, S.; Krueger, V. & Chellappa, R. (2003). Probabilistic recognition of human faces from video. *Computer Vision and Image Understanding*, Vol. 91, 2003, pp. 214-245

Zhou, X.; S´anchez, S. A. & Kuijper, A. (2010). 3D Face Recognition with Local Binary Patterns, *Proceedings of Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2010

Zhou, Z.; Ganesh, A.; Wright, J.; Tsai, S. F. & Ma, Y. (2008). Nearest-subspace patch matching for face recognition under varying pose and illumination, *Proceedings of 8th IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–8, Amsterdam, The Netherlands, Sept. 2008

4

Face Image Synthesis and Interpretation Using 3D Illumination-Based AAM Models

Salvador E. Ayala-Raggi, Leopoldo Altamirano-Robles
and Janeth Cruz-Enriquez

Instituto Nacional de Astrofísica Óptica y Electrónica
México

1. Introduction

One of the more exciting and unsolved problems in computer vision nowadays is automatic, fast and full interpretation of face images under variable conditions of lighting and pose. Interpretation is the inference of knowledge from an image. This knowledge covers relevant information, such as 3*D* shape and albedo, both related to the identity, but also information about physical factors which affect appearance of faces, such as pose and lighting. Interpretation of faces not only should be limited to retrieve the aforementioned pieces of information, but also, it should be capable of synthesizing novel facial images in which some of these pieces of information have been modified. This kind of interpretation can be achieved by using the paradigm known as analysis by synthesis, see Figure 1. Ideally, an approach based on analysis by synthesis, should consist of a generative facial parametric model that codes all the sources of appearance variation separately and independently, and an optimization algorithm which systematically varies the model parameters until the synthetic image produced by the model is as similar as possible to the test image, also called *input image*. A full interpretation approach should include the recovery of 3*D* shape, 3*D* pose, albedo and lighting from a single face image which exhibits any possible combination of these sources of variation.

Active appearance models, or simply *AAMs* (Cootes et al. (2001); Edwards et al. (1998); Matthews & Baker (2004)), with respect to other approaches, represent a fast alternative to perform face interpretation using the *analysis by synthesis* paradigm. Texture and shape, are attributes modeled by *AAMs* by using statistic tools such as *principal components analysis* or shortly *PCA*. However, the apparent texture of a face is an implicit combination of lighting and albedo. The separation process of these two attributes is not an easy task within the context of sparse models, like *AAMs*. *AAMs* use a sparse set of vertices which outline the shape. Texture is interpolated over that shape. In fact, a detailed dense set of surface normals, which is not available in *AAMs*, is required to perform the separation of lighting and albedo. On the other hand, texture and shape variation among human faces is relatively small when uniform lighting is considered. *AAMs* take advantage of this fact by supposing a constant relationship between changes of appearance and the variation of the model parameters producing those changes. This approximately constant relationship is a constant gradient which is used for performing fast fitting to input images. However, for most purposes, lighting is not uniform, and a proper separation of albedo and lighting becomes necessary. In a similar way as is texture variation in uniform lighting, albedo variation among human

faces is small. In contrast to albedo, lighting is not necessarily constrained to a small variation interval. In fact, lighting affects appearance more than identity and pose, and presents many degrees of freedom (see Ramamoorthi & Hanrahan (2001) and Basri et al. (2003)). During a fitting process, an initial model is gradually modified in each iteration until it match the input image. Therefore, if the illumination of the input image is too different from the illumination of the initial model, the ratio of appearance variation with respect to the parameters variation can not be the same during all the iterations of the fitting process. For instance, if we have a model with a pronounced left illumination, and a model with uniform illumination, the change of appearance caused by an increase on one of the model parameters, for example the parameter of scale, is not the same in both cases. This ratio of appearance variation with respect to the model parameters is in fact a Jacobian whose value changes in each iteration. Therefore, if we want to fit an AAM to a face with any kind of lighting, a constant Jacobian is not the solution. On the other hand, recomputing the Jacobian in each iteration is an expensive computational task Cootes et al. (2001),Matthews & Baker (2004).
In this chapter, we introduce an innovative 3D extension of AAMs based on an illumination model. By using interpolation, we incorporate a dense set of surface normals to our sparse 3D AAM model. In this way, we can model lighting within the process of synthesizing faces, and also within the optimization process used for fitting the face model to an input image. We propose a fitting method based on an inexpensive way for updating the Jacobian in accordance to the illumination parameters recalculated in each iteration. Our method is able to encode separately four of the more relevant sources of appearance variation: 3D shape, albedo, 3D pose and lighting. This approach estimates 3D shape, 3D pose, albedo, and illumination simultaneously during each iteration. Since our model uses analysis by synthesis, it has an inherent ability of adaptation to the input image. Adaptation is a desirable characteristic because it provides the possibility of designing person-independent face interpretation systems. Experimental results show that the proposed approach not only can be extended to face recognition, but also demonstrate its ability for fitting to novel faces and performing interpretation. We implement a novel way to cope with an important source of appearance variation which affects significatively face images: *lighting*. We anticipate that this approach can be extended to face recognition under difficult conditions of lighting and can be generalized to the analysis and recovery of other types of sources of appearance variation such as age, gender, expression, etc., where lighting interferes seriously in the analysis process.

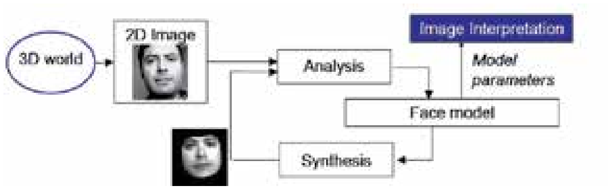


Fig. 1. Schematized flow of the analysis by synthesis approach.

Particularly, face interpretation has been faced through two paradigms: 3DMMs Blanz et al. (1999; 2003); Romdhani et al. (2005; 2006) and AAMs Cootes et al. (1998; 2001); Dornaika et al. (2003); Edwards et al. (1998); Kahraman et al. (2007); Legallou et al. (2006); Matthews & Baker (2004); Sattar et al. (2007); Xiao et al. (2004). 3DMMs cover a wide range of information recovery but are slow and cannot model properly every type of lighting. On the other hand, AAMs are fast but cannot model lighting and 3D information simultaneously. AAM models have been used for fast 2D face alignment under variable conditions of lighting Huang et al. (2004); Kahraman et al. (2007); Legallou et al. (2006), but not for estimation of 3D pose, 3*D* shape, albedo and illumination under non-uniform lighting conditions, which is still a challenging problem. In contrast, some authors Dornaika et al. (2003); Sattar et al. (2007); Xiao et al. (2004) have proposed 3*D AAM*s for estimating 3*D* pose and shape but do not include illumination. Finally, authors who reported lighting modeling for face recognition, do not propose methods for estimation of pose, shape, albedo and lighting simultaneously. This chapter describes a proposal for a complete 3*D* approach for an automatic and fast recovery of 3*D* shape, 3*D* pose, albedo and lighting of a face under non-uniform lighting and variable pose. This recovery is performed by fitting a parametric 3*D* Active Appearance Model based on the 9D subspace illumination model. Once we have finished the fitting process of the model to an input image, we obtain a compact set of parameters of shape, albedo, pose and lighting which describe the appearance of the original image. Because lighting parameters are not in a limited range, for faces with a pronounced non-uniform illumination, it is not possible to successfully use a constant Jacobian during all the fitting process as is done in original 2*D AAM* models Cootes et al. (2001). Instead of that, during the fitting stage, our algorithm uses the estimated lighting parameters, obtained in preceding iterations, for updating the Jacobian and the reference mean model on each iteration. The proposed method is called 3*D* Illumination-Based Active Appearance Models Ayala-Raggi et al. (2008), Ayala-Raggi et al. (2009) and is suitable for face alignment, pose estimation and synthesis of novel views (novel poses and lighting) of aligned faces. In this chapter, we explain the method, measuring its capability to recover 3*D* shape and albedo, and showing its capability to fit faces not included within the training set. Our experimental results, performed with real face images, show that the method could be extended to lighting-pose invariant face recognition.

2. Modeling lighting

Human face can be considered approximately as a convex surface with *Lambertian* reflectance Basri et al. (2003),Ramamoorthi & Hanrahan (2001). In Basri et al. (2003), Basri et al., propose using spherical harmonic functions to model lighting for face recognition. Spherical Harmonics are a set of functions which form an orthonormal basis which is able to represent all possible continuous functions defined in the sphere. The image of a face, illuminated by any lighting function can be expressed as a linear combination of harmonic reflectances (face images illuminated by harmonic lights),

$$I_i = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} I_{n,m} b_{n,m}(\mathbf{x}_i) \tag{1}$$

where $b_{n,m}$ are the set of harmonic reflectances and $\mathbf{x}_i$ is the i-th pixel of the object, in this case the face surface. In Basri et al. (2003), Basri et al. showed that the precision to approximate any function of light if we take a second order approximation $(n = 0, 1, 2)$ is at least 97.96%. From Equation (1) we see that this precision is achieved with only 9 harmonic images, and Equation (1) can be expressed in matrix notation as

$$\mathbf{I} = \mathbf{BL} \tag{2}$$

where $\mathbf{B}$ is a matrix with 9 columns. Each column is a harmonic image, and L is a column vector containing 9 arbitrary parameters.

2.1 Forcing the lighting model to be positive

By using Equation (2), we could obtain not physically realizable images if we take arbitrary linear combinations of the harmonic images. In fact, any arbitrary combination could produce an image with negative values. The harmonic images themselves have negative values, and as we know, light intensity is always positive. Therefore, different combinations of lightings must produce positive intensity values too. In Basri et al. (2003), the authors showed that the soft harmonic space spanned by the harmonic images can be discretized by using a sufficiently populated set of point light sources (delta functions) uniformly distributed around the sphere. Thus, Equation (2) can be modified as

$$\mathbf{I} = \mathbf{B}\mathbf{H}^T\mathbf{L} \tag{3}$$

where $\mathbf{L}$ is a column vector of arbitrary lighting parameters and $\mathbf{B}$ is a matrix with columns formed by the nine harmonic images. $\mathbf{H}$ is a matrix whose columns contain samples of the harmonic functions, whereas its rows contain the transform of the delta functions corresponding to the discrete number of point light sources.

This is a mathematical way of making discrete the smooth harmonic subspace by sampling the harmonic reflectance images. The more densely populated with deltas is $\mathbf{H}$, the better is the approach to the original space of the 9 harmonics. In order to obtain a good approximation to the original harmonic space, we should use a large set of point lights uniformly distributed around the sphere. However, in Lee et al. (2001), Lee et al., found an important result about how to approximate the illumination cone of lighting (see Georghiades et al. (1998)) with a small number of deltas. Only nine light point sources strategically distributed are necessary for approximating any reflectance on a face. Thus, $\mathbf{H}$ will be a constant 9×9 matrix.

In fact, the basis images can be obtained from two possible ways, the first one is the explained here, by using the compact notation through the spherical harmonics reflectances, and the second one is to explicitly render each one of the basis images, obtained from computing the intensity of each point by using the Lambert's law. This intensity can be computed if we know the surface normal, the albedo and the corresponding vector of the point light source.

3. Face synthesis using a $3D$ illumination-based active appearance model ($3D$-IAAM)

In this section, we describe an original method for face image synthesis based on the $3D-IAAM$ model proposed in this chapter. Our face synthesizer is capable of creating face images with arbitrary $3D$ pose, identity and illumination.

3.1 Construction of a bootstrap set of surfaces and albedo maps

In order to construct parametric models of shape and albedo, we need a bootstrap set of $3D$ face surfaces of different individuals, and their corresponding $2D$ albedo maps. This set of surfaces and albedo maps will be used to train models of $3D$ shape and $2D$ albedo, respectively.

3.1.1 Recovery of the face surface for each training identity

A bootstrap set of face surfaces can be obtained under well controlled laboratory conditions by using a set of distant directional lights which illuminate the face one at the time but all working during a short period of time, in such a way that there is not movement from one image to the next.

Surfaces can be recovered by using a technique known as *photometric stereo* Forsyth & Ponce (2002); Horn et al. (1978); Silver (1980); Woodham (1989). By using M ($M > 3$) different images per individual, each one illuminated by a different point light source, it is possible to simultaneously estimate the surface normals map and the albedo map of a face. This is

accomplished by using minimum squares for solving a linear system of M equations, each one expressing the pixel intensity as a function of the direction of the incident light (Lambert's cosine law) for each pixel. From surface normals maps, it is possible to reconstruct the surface of each face by using *shapelets* Kovesi (2005). This is done by correlating the surface normals with those of a bank of *shapelet* basis functions. The correlation results are summed to produce the reconstruction. The summation of shapelet basis functions results in an implicit integration of the surface while enforcing surface continuity.
On the other hand, a mean surface normals map, computed from the set of surface normals maps, is used as a deformable template for building basis reflectance images during the fitting stage.

3.2 Constructing the models of shape and albedo

In order to obtain a parametric $3D$ shape model, first of all, we have to capture the more significative modes of shape variation. This can be accomplished by using a statistical method such as PCA (*principal component analysis*) applied to a set of training faces with different identity. We can place $3D$ landmarks over the surface of N training faces. To be sure that we are only modeling variations in shape and not in pose, we have to align the $3D$ shape models first, by using an iterative algorithm based on Procrustes analysis (see Figure 2).

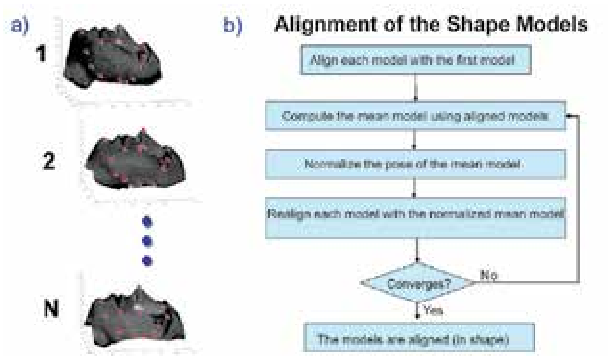


Fig. 2. The shape models (each one defined as the set of landmarks over a particular face surface) (a) must be aligned by using Procrustes Analysis (Ross (2004)) (b) before performing the statistical study of shape variation.

Then we apply PCA to the set in order to obtain the principal modes of variation of $3D$ shape. We can generate an arbitrary model using the following expression

$$\mathbf{s} = \bar{\mathbf{s}} + \mathbf{Q}_s \mathbf{c} \tag{4}$$

where $\bar{\mathbf{s}}$ is the mean shape model and $\mathbf{Q}_s$ is a matrix which contains the basis shapes (also known as *eigenshapes*) and $\mathbf{c}$ is a vector with arbitrary shape parameters. Similarly, we apply PCA to the set of shape-normalized $2D$ albedos maps. Before applying PCA, the albedos map of each training face must be shape-normalized (using the bidimensional projection of the mean shape frame) as is shown in Figure 3.
A triangulation is designed to warp original images into the mean shape frame. Finally, any shape-normalized albedo image can be generated with

$$\lambda = \bar{\lambda} + \mathbf{Q}_\lambda \mathbf{a} \tag{5}$$

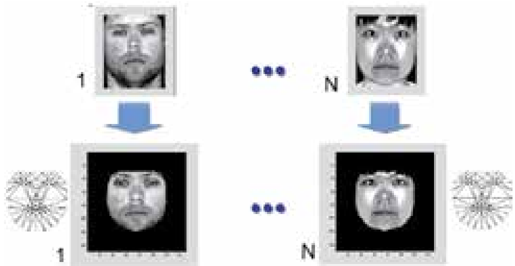


Fig. 3. Normalizing in shape the albedo images by warping the original albedo images into the $2D$ projection of the mean shape. Top: Original albedo images. Bottom: shape-normalized albedo images.

where $\bar{\lambda}$ is the mean albedo image, $\mathbf{Q}_{\lambda}$ is a matrix which contains principal albedo variation modes and $\mathbf{a}$ is a vector of arbitrary parameters.

3.3 Synthesizing faces with novel appearances

By using Equation (5), it is possible to synthesize an arbitrary albedo image λ and then warp it to the $2D$ projection of an arbitrary frontal shape generated with Equation (4). This new face is not illuminated yet. In the same process of warping the albedo image to the new shape, it is also possible to carry out a $2D$ warping from the $2D$ mean map of surface normals (calculated during the training stage) to the same new shape $\mathbf{s}$. So far, we have a new albedo image and a new map of surface normals, both of them shaped according to the new generated shape. With these two maps (albedos and normals), we can construct 9 basis reflectance images as is described in Section 2 by using Equation 3. Any illumination can be generated by a linear combination of these basis images. In order to give a $3D$ pose to the model, we use the $3D$ landmarks of the new generated $3D$ shape. By applying a rigid body transformation $(\mathbf{T}, \mathbf{R}, s)$ to these landmarks we give any pose and size to the created face.

If we suppose that the distance from the camera to the face is considerably greater than the depth of the face itself, then it is reasonable to use a simple orthographic projection model. Orthographic projection is the projection of a $3D$ object onto a plane by a set of parallel rays orthogonal to the image plane.

Finally, we warp the frontal face to the $2D$ orthographic projection of the transformed $3D$ shape. Figure 4 illustrates the synthesis process.

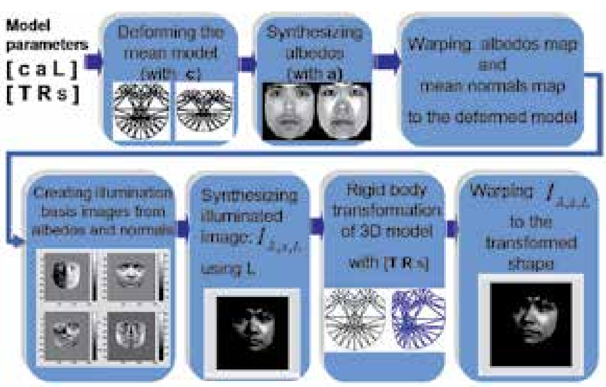


Fig. 4. Face synthesis process.

4. Face alignment using the $3D$-IAAM model

The original $2D$ *AAM* approach for face alignment presented in Cootes et al. (2001), consists of an iterative algorithm which minimizes the residual obtained by comparing a shape-normalized region (taken from the target image) with a reference mean-shape model which evolves in texture in each iteration. This method supposes a constant relationship between residuals and the additive increments to the model parameters. This approximation uses a constant Jacobian during all the fitting process, and works well when lighting is uniform because texture variation is small and residuals are always computed in the same reference frame, see Cootes et al. (2001). Since we know, in contrast to texture in human faces, lighting variation is not limited. Therefore, if the initial reference model is substantially different in lighting to that in the input image, it is not possible to consider a constant Jacobian for all the fitting process. Here, we propose an iterative fitting algorithm capable of correcting the Jacobian in each iteration by using the current estimation of lighting, which in turn, is used to update the reference model too.

4.1 Overview of the iterative fitting process

Once we have created the models of shape and albedo, we can use them in the face alignment process. The alignment process consists of an iterative algorithm which captures a region within the input image, performs a normalization of this region according to the current set of model parameters and compares this normalized image with a reference model. The comparison is always performed into a fixed reference shape. The reference model evolves only in lighting in each iteration. The resulting residual from that comparison is used in conjunction with a Jacobian for calculating suitable increments to be added to the current model parameters. During the following iteration the new set of model parameters are used again to capture and normalize a new region within the input image, and so on. At the beginning of the alignment process, a set of initial model parameters is defined by the user. Commonly, shape, albedo and rotation parameters are initialized with zero, illumination parameters are initialized to a medium illumination, and translation and scale parameters are initialized to a rough value near to the real $2D$ position and size of the face. In other words, initial parameters are initialized in such a way that they would produce a frontal mean face placed over the face in the input image.

On the other hand, at the end of the alignment process, the final set of model parameters should be capable of synthesizing a face image similar to the original in the input image by using the synthesis process described in section 3.3. The normalization process over the input image is composed by a pose normalization, a shape normalization and an albedo normalization, all described in the following subsections.

4.2 Pose and shape normalization

In each iteration the model parameters of $3D$ shape and $3D$ pose determine a $3D$ structure whose orthographic $2D$ projection is used to define a region within the input image. This region can be mapped to a reference shape-normalized frame.

By using the rigid body transformation parameters $(\mathbf{T}, \mathbf{R}, s)$ and the shape parameters $\mathbf{c}$, a region in the image is sampled and warped to the $2D$ mean shape frame. This new shape-normalized image is denoted as $\mathbf{I}_{shape\,aligned}$.

4.3 Albedo normalization

A novel contribution of this work is a method for normalizing albedo when we have an estimate of lighting and albedo parameters. In fact, at the beginning of the fitting process, albedo parameters have a zero value, then the normalization will produce the same image before normalization, see Equation 13. In contrast, as the albedo and illumination parameters get closer to the ideal values for synthesizing a face equal to the original, then normalization will produce an image more similar to a face with mean albedo illuminated by the actual lighting present in the original image. The image normalized in pose, shape and albedo, can be compared with a reference mean-shape mean-albedo face which evolves in lighting each iteration. The residual obtained from this comparison will give us the possibility to use a gradient matrix, or simply a Jacobian which is almost constant and is easily updated by using the estimated illumination parameters.

4.3.1 Albedo normalization by using a current estimation of parameters of albedo and illumination

In Section 2 we have showed that every illumination over a face can be synthesized by using the following expression

$$\mathbf{I} = \mathbf{B}\mathbf{H}_{9PL}^{T}\mathbf{L} \tag{6}$$

as explained before, $\mathbf{B}\mathbf{H}_{9PL}^{T}$ represents a matrix with nine columns each one being a real and positive basis reflectance image. In order to compact the notation, we can denote that matrix as

$$\beta_{9PL} = \mathbf{B}\mathbf{H}_{9PL}^{T} \tag{7}$$

then Equation 6 can be rewritten as

$$\mathbf{I}_{illuminated\,face} = \beta_{9PL}\mathbf{L} = ([\lambda..\lambda] \cdot \mathbf{\Phi})\mathbf{L} \tag{8}$$

where λ is the albedos map represented as a column vector repeated in order to form a matrix with the same dimensions as the basis reflectances matrix without albedo, represented by $\mathbf{\Phi}$. These two matrices are multiplied in an element-wise fashion (Hadamard product). Then, $\mathbf{I}_{illuminated\,face}$ can be rewritten as

$$\mathbf{I}_{illuminated\,face} = \lambda \cdot (\mathbf{\Phi}\mathbf{L}) \tag{9}$$

Now, suppose that the fitting algorithm has successfully recovered the shape and pose parameters corresponding to the input image. In that situation, the process of pose and shape normalization explained in the preceding section would produce a frontal shape-normalized face.

On the other hand, if we would know the correct illumination parameters $\mathbf{L}$ of that face, we could solve for the albedo by manipulating Equation 9 and using $\mathbf{I}_{shape\,aligned}$ instead of $\mathbf{I}_{illuminated\,face}$,

$$\hat{\lambda} = \frac{(\mathbf{I}_{shape\,aligned})}{(\mathbf{\Phi}\hat{\mathbf{L}})} \tag{10}$$

where the division denotes an element-wise division.

Suppose now, that we have a correct estimation of the albedo parameters $(\mathbf{a})$. Then, by using $\hat{\lambda}$ and the albedo parameters $(\mathbf{a})$ we can derive an approximated mean albedo by using Equation 5,

$$\tilde{\lambda} \approx \hat{\lambda} - \mathbf{Q}_{\lambda}\mathbf{a} \tag{11}$$

Finally, we can normalize the image in albedo by using $\tilde{\lambda}$,

$$\mathbf{I}_{aligned} = (\tilde{\lambda}) \cdot (\mathbf{\Phi}\hat{\mathbf{L}}) \tag{12}$$

where $\hat{\mathbf{L}}$ is a vector containing the current estimated illumination parameters. We can rewrite Equation 12 as

$$\mathbf{I}_{aligned} = [I_{shape\,aligned} . / (\mathbf{\Phi}\hat{\mathbf{L}}) - \mathbf{Q}_{\lambda}\mathbf{a}] \cdot (\mathbf{\Phi}\hat{\mathbf{L}}) \tag{13}$$

The residuals vector can be calculated as

$$\mathbf{r} = \mathbf{I}_{aligned} - \bar{\lambda} \cdot (\mathbf{\Phi}\hat{\mathbf{L}}) \tag{14}$$

The energy of this residual image is a quantity to minimize by the iterative optimization algorithm

$$\|\mathbf{r}\|^2 = \|\mathbf{I}_{aligned} - \bar{\lambda} \cdot (\mathbf{\Phi}\hat{\mathbf{L}})\|^2 \tag{15}$$

where $[\bar{\lambda} \cdot (\mathbf{\Phi}\hat{\mathbf{L}})]$ represents the reference model with mean shape, mean pose, mean albedo, but illumination determined by the last estimated lighting parameters $\hat{\mathbf{L}}$. The process for obtaining residuals in each iteration is shown in Figure 5, where the reference model $[\bar{\lambda} \cdot (\mathbf{\Phi}\hat{\mathbf{L}})]$ is denoted by $\bar{\mathbf{f}}$.

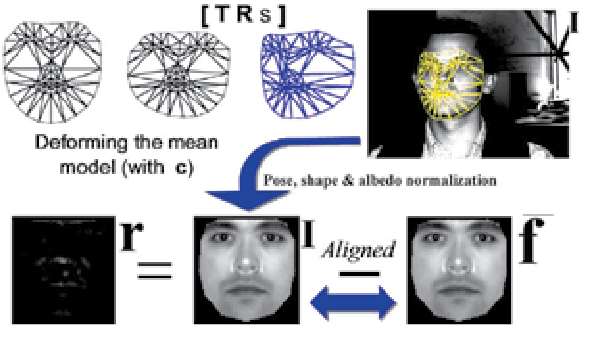


Fig. 5. Estimation of residuals during a step of the fitting process. The mean shape is deformed by using the current parameters **c** and **ø** (top). Then, the region within the $2D$ projection of this new structure is warped from the test image to the reference mean shape frame (in the bottom and in the middle) in order to apply the process of albedo normalization. The resulting image called $I_{Aligned}$ is compared with a reference model in order to obtain a residual image.

In order to work with a more compact notation, we can view the pose-shape and albedo normalization as an inverse transformation to the $3D$-IAAM synthesis process. Therefore, we can denote that process as

$$\mathbf{I}_{aligned} = T_{\mathbf{p}}^{-1}(\mathbf{I}_{input}) \tag{16}$$

where $\mathbf{I}_{input}$ represents the input image and $\mathbf{p}$ is a vector containing the model parameters $\mathbf{p} = (\mathbf{T}^T, \mathbf{R}^T, s, \mathbf{c}^T, \mathbf{L}^T, \mathbf{a}^T)^T$. The initial parameters for the start of a fitting process are denoted as

$$\mathbf{p}_0 = (\mathbf{T}_0^T, \mathbf{R}_0^T, s_0, \mathbf{c}_0^T, \mathbf{L}_0^T, \mathbf{a}_0^T)^T \tag{17}$$

where $\mathbf{T}_0^T$ is the initial position vector (x_0, y_0) given by the user. $\mathbf{R}_0^T = (0, 0, 0)$ is the initial rotation vector, and s_0 the initial scale factor (commonly equal with 1). $\mathbf{c}_0^T = (0, 0, 0, 0, ...)$ is the initial shape parameters vector. $\mathbf{L}_0^T = (L0_1, L0_2, L0_3, L0_4, L0_5, L0_6, L0_7, L0_8, L0_9)$ is the initial illumination parameters vector. Finally, $\mathbf{a}_0^T = (0, 0, 0, 0, ...)$ is the initial albedo parameters vector.

4.4 Modeling the residuals

During the fit, according to the last estimated parameters, the pixels inside of a region in the image are sampled and transformed. So, the residuals image computed with (14) is a function of the model parameters $\mathbf{p}$, that is $\mathbf{r} = \mathbf{r}(\mathbf{p})$. The first order Taylor expansion of (14) gives $\mathbf{r}(\mathbf{p} + \delta\mathbf{p}) = \mathbf{r}(\mathbf{p}) + \frac{\delta\mathbf{r}}{\delta\mathbf{p}}\delta\mathbf{p}$, here, $\mathbf{p}^T = (\mathbf{T}^T|\mathbf{R}^T|s|\mathbf{c}^T|\mathbf{a}^T|\mathbf{L}^T)$, and the $ij - th$ element of the matrix $\frac{\delta\mathbf{r}}{\delta\mathbf{p}}$ is $\frac{\partial r_i}{\partial p_j}$. We desire to choose $\delta\mathbf{p}$ such that it minimize $||r(\mathbf{p} + \delta\mathbf{p})||^2$. Equating $\mathbf{r}(\mathbf{p} + \delta\mathbf{p})$ to zero leads to the solution

$$\delta\mathbf{p} = -\mathbf{J}^{-1}\mathbf{r}(\mathbf{p}) \tag{18}$$

and $\mathbf{J}^{-1}$ can be calculated by pseudo-inverting the Jacobian matrix (Moore-Penrose pseudo-inverse), or by using the normal equations:

$$\mathbf{J}^{-1} = \left(\frac{\delta\mathbf{r}^T}{\delta\mathbf{p}}\frac{\delta\mathbf{r}}{\delta\mathbf{p}}\right)^{-1}\frac{\delta\mathbf{r}^T}{\delta\mathbf{p}} \tag{19}$$

where $\frac{\delta\mathbf{r}}{\delta\mathbf{p}}$ is actually a gradient matrix or Jacobian changing in each iteration. Recalculating it at every step is expensive. Cootes et al. in Cootes et al. (2001), assume it to be constant since it is being computed in a normalized reference frame. This assumption is valid when we are only considering variations of texture, and lighting is ignored because it is uniform. Since texture parameters do not present a large variation between training faces, then, it is possible to compute a weighted average of the residuals images for each displaced parameter in order to obtain an average constant Jacobian. In our case, we are dealing with non-uniform illumination, therefore we propose to construct an adaptive Jacobian as is explained later.

4.5 Iterative fitting algorithm

In Cootes et al. (2001), authors propose to utilize a precalculated constant Jacobian matrix which is used during all the fitting process. Each iteration, a sampled region of the image is compared with a reference face image normalized in shape which is updated only in texture according to the current estimated parameters. Ideally, this reference image constitutes a reference model evolving in texture which should be associated to a Jacobian evolving in texture too. However, in practice, a mean constant Jacobian, computed from the different textures found in the training set, is used. This constant Jacobian works well in uniform lighting conditions, because texture variation between training faces is relatively small. Nevertheless, using a constant Jacobian would produce bad alignments in both, the approach described in Cootes et al. (2001) and in our $3D$ approach Ayala-Raggi et al. (2008) when the lighting of the input face is considerably different from the lighting used during the training stage. In our $3D$ approach, an ideal procedure to achieve good convergence results, at a high computational cost, would be to recalculate completely the Jacobian each iteration. This operation could be performed each iteration by displacing the parameters of the reference model. The parameters of albedo and illumination would be displaced from their current estimated values, and the $3D$ shape and pose parameters from their mean state values. All these parameter displacements would be used to synthesize displaced face images which, in turn, would be used for computing residuals by subtracting the images without displacement from the displaced images. Finally, residual images and their respective parameter displacements would be used to calculate the Jacobian. This process of synthesis of multiple images should be performed on-line during the fitting stage and certainly would be an extremely expensive operation.

In contrast, we propose a computationally inexpensive way to update the Jacobian by using the current illumination parameters. Each iteration, our optimization algorithm samples a region of the image and normalizes it in pose, shape and albedo. Albedo normalization is performed by using the current estimated illumination parameters. Thus, a comparison should be computed between this normalized image and the reference mean model (a model with mean shape and albedo) illuminated by using the same current illumination parameters. The estimated residuals and an updated Jacobian (*re-illuminated* by using the current estimated lighting) can be used to compute the new parameters displacements.

Updating the Jacobian with the current estimated illumination parameters is an easy and computationally inexpensive step, because we use the fact that lighting and albedo are separated vectors and they are independent of basis reflectance images, see Equation 9. In training time, we construct a set of displaced images that will be used during the fitting stage to update the Jacobian. We know that basis reflectances $\mathbf{\Phi}$ (without albedo) are not affected by albedo displacements, but they can be modified by pose and shape increments. Our model uses 33 parameters: 6 for pose, 9 for $3D$ shape, 9 for illumination, and 9 for albedo. We construct 15 ($6 + 9 = 15$) basis reflectance matrices $\mathbf{\Phi}_{p_i+\Delta p_i}$ by displacing, in a suitable quantity, each one of the 15 parameters of pose and shape. That is, by using face synthesis (through our model), we synthesize each reflectance image represented as a column within the matrix $\mathbf{\Phi}_{p_i+\Delta p_i}$ by giving the following synthesis parameters:

$$\mathbf{p} = (p_1, p_2, ..., p_i + \Delta p_i, ..., p_{15})^T \tag{20}$$

For instance, if $i = 8$, i.e. we are constructing the matrix for the second shape parameter, then the generating parameters $\mathbf{p}$ will be:

$$\mathbf{p} = (\mathbf{T}_0^T, \mathbf{R}_0^T, s_0, [0 \quad (0 + \Delta p_8) \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0])^T \tag{21}$$

In practice, we construct 30 basis reflectance matrices because we consider 15 positive displacements and 15 negative displacements. In a similar way, by displacing each parameter with a suitable increment $p_i + \Delta p_i$ (positive and negative), we obtain 30 albedo images for positive and negative increments in pose and shape parameters, and 18 albedo images for positive and negative increments in albedo parameters. These albedo images do not have information about lighting.

These 30 reflectance matrices and 48 albedo images are created during training time (*off-line*). During the alignment stage, we can create a Jacobian *on-line* according to the current parameters of illumination $\mathbf{L}$:

$$\frac{\delta \mathbf{r}}{\delta \mathbf{p}} = [\frac{\partial \mathbf{r}_1}{\partial \mathbf{p}_1} \dots \frac{\partial \mathbf{r}_{33}}{\partial \mathbf{p}_{33}}] \tag{22}$$

where each column can be calculated as:

$$\frac{\partial \mathbf{r}_i}{\partial \mathbf{p}_i} = [\frac{\partial \mathbf{r}_i}{\partial \mathbf{p}_{i\ (\Delta+)}} + \frac{\partial \mathbf{r}_i}{\partial \mathbf{p}_{i\ (\Delta-)}}] \times \frac{1}{2} \tag{23}$$

with $i = 1, 2, ..., 33$. Here, $\frac{\partial \mathbf{r}_i}{\partial \mathbf{p}_{i\ (\Delta+)}}$ and $\frac{\partial \mathbf{r}_i}{\partial \mathbf{p}_{i\ (\Delta-)}}$ can be computed as:

$$\frac{\partial \mathbf{r}_i}{\partial \mathbf{p}_{i\ (\Delta+)}} = \frac{\lambda_{p_i+\Delta p_i} \cdot [\mathbf{\Phi}_{p_i+\Delta p_i}\mathbf{L}] - \lambda_0 \cdot [\mathbf{\Phi}_0\mathbf{L}]}{\Delta p_i} \tag{24}$$

$$\frac{\partial \mathbf{r}_i}{\partial \mathbf{p}_{i\ (\Delta-)}} = \frac{\lambda_{p_i-\Delta p_i} \cdot [\mathbf{\Phi}_{p_i-\Delta p_i}\mathbf{L}] - \lambda_0 \cdot [\mathbf{\Phi}_0\mathbf{L}]}{-\Delta p_i} \tag{25}$$

where λ_0 is the mean albedo, and $\mathbf{\Phi}_0$ is the matrix which columns are the mean basis reflectances (without albedo information). When p_i corresponds to an albedo parameter, then we use $\mathbf{\Phi}_{p_i+\Delta p_i} = \mathbf{\Phi}_0$, since the reflectance matrices are not affected by albedo variations.
Because the Jacobian is constructed using the last estimated lighting parameters, we denote it as $\mathbf{J}(\hat{\mathbf{L}})$,

$$\mathbf{J}(\hat{\mathbf{L}}) = \frac{\delta \mathbf{r}}{\delta \mathbf{p}} \tag{26}$$

The iterative fitting algorithm is outlined in Figure 6.
Basically, the algorithm can be summarized as follows: When the fitting process begins, $I_{aligned}$ is an unprocessed region of the test image delimited only by the position of the initial model over the image. There is not shape or albedo normalization at this moment, so that the residual (step 2) will be computed between the region (without transformation) and the model in a similar way such as it is done in the $2D\ AAM$ algorithm Cootes et al. (2001). This first residual in combination with the Jacobian (which is a precalculated constant the first time) produces (such as it happens in Cootes et al. (2001)) an additive increment vector $\delta\mathbf{p}$ to be added to the initial parameters. $\delta\mathbf{p}$ is iteratively reduced by re-scaling it (step 15) until the energy of the residual is lower than its initial estimate. If this value does not decrease after a fixed number of reductions, the algorithm claims that convergence was not reached and stops. Otherwise, if the value is lower than the initial, then the new set of model parameters is used again to normalize a new region within the test image. The new residual in combination with a new Jacobian is used to compute a new set of increments to the parameters, and so on. Figure 7 illustrates two consecutive iterations of the fitting process.
On the other hand, Figure 8 shows the evolution of the model during the fitting process. Figure 8 is illustrative and shows only five representative iterations in both alignments. Actually, the algorithm takes an average of 14 iterations to reach convergence.
In practice, we have implemented this algorithm using a pyramid of two resolution levels. A multi-resolution approach overcomes to the single resolution method and improves the convergence of the algorithm, even if we place the initial model farther from the actual face. On the other hand, the columns within the Jacobian matrix which correspond to illumination parameters, are maintained fixed during the fitting process and they are precalculated from a mean state of uniform lighting.

1. Project the sampled region from the input image $\mathbf{I}_{input}$ to the mean-shape model frame by applying pose-shape-albedo normalization $\mathbf{I}_{aligned} = T^{-1}_{\mathbf{p}_0}(\mathbf{I}_{input})$ with parameters $\mathbf{p} = \mathbf{p}_0$.
2. Compute the residual, $\mathbf{r} = \mathbf{I}_{aligned} - \bar{\lambda} \cdot (\mathbf{\Phi L}_0)$
3. Compute the predicted displacements, $\delta\mathbf{p} = -[\mathbf{J}_0]^{-1}\mathbf{r}(\mathbf{p})$. Where $\mathbf{J}_0$ is a Jacobian computed in the training stage by taking little displacements of the parameters from their initial values $\mathbf{p}_0$. $[\mathbf{J}_0]^{-1}$ is the Moore-Penrose pseudoinverse matrix of the Jacobian.
4. Take only the new estimate of illumination parameters and put the other parameters in their initial values ignoring the estimates, $\mathbf{p}_0 = (\mathbf{T}_0, \mathbf{R}_0, s_0, \mathbf{c}_0, \hat{\mathbf{L}}, \mathbf{a}_0)$
5. Set $\mathbf{p} = \mathbf{p}_0$.
6. Project the sampled region from the input image $\mathbf{I}_{input}$ to the mean-shape model frame by applying pose-shape-albedo normalization $\mathbf{I}_{aligned} = T^{-1}_{\mathbf{p}}(\mathbf{I}_{input})$
7. Compute the residual, $\mathbf{r} = \mathbf{I}_{aligned} - \bar{\lambda} \cdot (\mathbf{\Phi}\hat{\mathbf{L}})$
8. Compute the current error, $E = ||\mathbf{r}||^2$
9. Compute the predicted displacements, $\delta\mathbf{p} = -\mathbf{J}^{-1}\mathbf{r}(\mathbf{p})$. Here $\mathbf{J}^{-1} = [\mathbf{J}(\hat{\mathbf{L}})]^{-1}$. Jacobian $\mathbf{J}(\hat{\mathbf{L}})$ is assembled by using the precomputed images of basis reflectance and albedo in combination with the estimated parameters $\hat{\mathbf{L}}$ computed in last iteration, see Equations 24 and 25.
10. Update the model parameters $\mathbf{p} \longrightarrow \mathbf{p} + k\delta\mathbf{p}$, where initially $k = 1$.
11. Using the new parameters, calculate the new face structure $\mathbf{X}$ and the new mean-shape reference model $\bar{\lambda} \cdot (\mathbf{\Phi}\hat{\mathbf{L}})$.
12. Compute $\mathbf{I}_{aligned} = T^{-1}_{\mathbf{p}}(\mathbf{I}_{input})$
13. Calculate a new residual $\mathbf{r} = \mathbf{I}_{aligned} - \bar{\lambda} \cdot (\mathbf{\Phi}\hat{\mathbf{L}})$
14. If $||\mathbf{r}||^2 < Threshold$ then terminate else go to the next step
15. If $||\mathbf{r}||^2 < E$, then accept the new estimate, make $k = 1$ and go to step 8; otherwise go to step 10 and try at $k = 0.5, k = 0.25$, etc.. (In practice, after 7 attempts of reducing k, if $||\mathbf{r}||^2 \geq E$ then the fitting process is finished.)

Fig. 6. Fitting algorithm.

5. Experimental results

5.1 Individuals used

We evaluated our approach on two different datasets. The first one was called set A and is composed by the 10 identities contained in the Yale B Database. Each subject in the database is photographed in six different poses. For each pose many different illuminations are available. A second dataset, that we call set B is composed by 20 individuals. This second dataset is composed by the 10 identities from Yale B Database plus other 10 identities randomly selected from the extended Yale B database (which contains 28 identities from different ethnicity).

5.2 Setup for experiments

The test set for this experiments was composed by 60 real images (with a size of 320×240 pixels) taken from Yale database B in the following manner: all images have the pose number 6 which presents a similar angle in azimuth to the left and elevation up. This pose has an angle of 24 degrees from the camera axis. We choose 6 different illuminations for using with each one of the identities. Each illumination is generated by a single point light source, and its

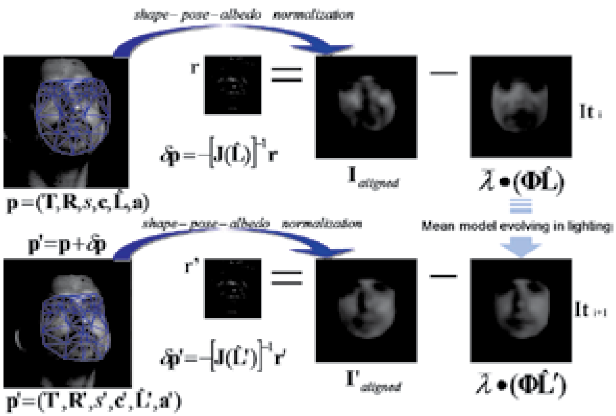


Fig. 7. Two consecutive iterations of the fitting process. During the iteration It_i a region in the test image is captured and normalized according to the current parameters $\mathbf{p}$ producing the image $I_{aligned}$. A residual $\mathbf{r}$ is calculated by comparing $I_{aligned}$ with a reference shape-normalized model illuminated by the current parameter $\hat{\mathbf{L}}$. An additive increment vector $\delta\mathbf{p}$ is computed. $\delta\mathbf{p}$ is iteratively reduced by re-scaling it (step 15) until the energy of the residual is lower than its initial estimate. When this event occurs, the new set of model parameters $\mathbf{p}'$ is used again to normalize a new region within the test image. The new residual $\mathbf{r}'$ in combination with a new Jacobian $\mathbf{J}(\hat{\mathbf{L}}')$ is used to compute a new set of increments to the parameters, and so on.

Fig. 8. Evolution of the synthetic face produced by the model during the fitting process, from initialization to convergence.

direction is specified with an azimuth angle and an elevation angle with respect to the camera axis, see table 1.

L1	L2	L3	L4	L5	L6
$A+50E+00$	$A+35E+15$	$A+10E+00$	$A-10E+00$	$A-35E+15$	$A-50E+00$

Table 1. Illuminations used for experiments.

The initial conditions of the model at the beginning of the fitting process were manually setup only in translation and scale. The rest of the parameters: rotation, $3D$ shape, illumination and albedo were always initialized in their mean state for all the alignments, i.e., rotation: $\mathbf{R}_0^T = [0,0,0]$, $3D$ shape: $\mathbf{c}_0^T = [0,0,0,0,0,0,0,0,0]$, albedo: $\mathbf{a}_0^T = [0,0,0,0,0,0,0,0,0]$, and illumination: $\mathbf{L}_0^T = [0.6, 0.6, 0.6, 0.4, 0.4, 0.9, 0.9, 0.4, 0.4]$ (this configuration of the intensity of the light sources produces a *mean lighting* which illuminates uniformly the face).
In all the alignments, the translation and scale parameters were initialized with the output values of a manual pose estimator which uses three landmarks manually placed on the two external eye corners and on the tip of the nose. The output of this manual estimator are rigid

body parameters $(\mathbf{T}, \mathbf{R}, s)$ computed by using $3D$ geometry. From those parameters, we only used the translation and scale values in order to initialize the fitting process.
Our fitting algorithm is a local optimization and can fall into local minima, particularly if the initial model is placed far from the face to fit. We observed that the algorithm converges if we give an initial translation value with a maximum difference of ± 10 pixels far from the ideal initial position. Therefore, the algorithm tolerates up to certain degree of imprecision in the initial position of the model over the test image.
Over the test set (the 60 images) we performed 180 alignments distributed within the following groups:

1. Group 1: 60 alignments using the fitting algorithm programmed with 4 computations of the adaptive Jacobian. That is, the algorithm has been allowed to recompute the Jacobian only during the first 4 consecutive iterations.
2. Group 2: 60 alignments using the fitting algorithm programmed with 2 computations of the adaptive Jacobian. That is, the algorithm has been allowed to recompute the Jacobian only during the first 2 consecutive iterations.
3. Group 3: 60 alignments using the fitting algorithm programmed with a constant Jacobian.

Figure 9 shows the alignments belonging to Group 1 (4 computations of the Jacobian) for each one of the 6 illuminations for identity 7.

Fig. 9. Alignments for identity 7 with each one of the 6 different illuminations.

5.3 Recovery of $3D$ shape and Albedo and measuring its quality through identification

In order to measure the quality of the recovered $3D$ shape and albedo, we have considered that this quality is encoded into the recovered shape and albedo parameters. These estimated shape and albedo parameters are directly related to identity. Therefore, it is reasonable to compare them with those stored within a gallery containing parameters of all the training identities. In fact, PCA allows the computation of the generative parameters for each training identity when the models of shape and albedo are created (see Section 3.2).
As a previous step before performing the comparison between estimated and stored parameters, they have to be re-scaled by dividing them by their respective standard deviations. Then, we measure the distance between the recovered parameters and the original parameters from the gallery. An appropriate distance measure in this case is the cosine of the angle between both vectors. This metric has the advantage of being insensitive to the norm of both vectors. In fact, that norm does not modify the perceived identity (see Romdhani (2005)). This operation was performed separately for vectors of albedo and for vectors of shape.

If we denote as $\sphericalangle(\hat{\mathbf{a}}, \mathbf{a}_i)$ the angle between the vector $\hat{\mathbf{a}}$ (estimated albedo parameters) and the vector $\mathbf{a}_i$ (stored albedo parameters for identity i), then the cosine can be computed with the following expression for albedo:

$$\Omega_i^a = cos(\sphericalangle(\hat{\mathbf{a}}, \mathbf{a}_i)) = \frac{\hat{\mathbf{a}}^T \mathbf{a}_i}{\sqrt{(\hat{\mathbf{a}}^T \hat{\mathbf{a}})(\mathbf{a}_i^T \mathbf{a}_i)}} \tag{27}$$

and the following expression for $3D$ shape parameters:

$$\Omega_i^s = cos(\sphericalangle(\hat{\mathbf{c}}, \mathbf{c}_i)) = \frac{\hat{\mathbf{c}}^T \mathbf{c}_i}{\sqrt{(\hat{\mathbf{c}}^T \hat{\mathbf{c}})(\mathbf{c}_i^T \mathbf{c}_i)}} \tag{28}$$

where $\hat{\mathbf{c}}$ are the estimated shape parameters vector, and i is an index which indicates the identity of the parameters vector stored in the gallery. Because the cosine function might be negative, Ω_i^s and Ω_i^a are equated to zero in such a case. This positive cosine function works fine because we are interested on detecting only small angles related with the presence of high similarity between faces.

In order to perform the identification, we have to combine these two results (cosines for shape and cosines for albedo) to obtain a single identification result. An appropriate approach to combine both cosines is to convert them in likelihood values.

Using the known probability property which states that the sum of all likelihoods must be 1, we can normalize the computed cosines for albedo:

$$IL_i^a = \frac{\Omega_i^a}{(\Omega_1^a + \Omega_2^a + \Omega_3^a + \cdots + \Omega_{10}^a)} \tag{29}$$

and normalize the computed cosines for shape:

$$IL_i^s = \frac{\Omega_i^s}{(\Omega_1^s + \Omega_2^s + \Omega_3^s + \cdots + \Omega_{10}^s)} \tag{30}$$

where IL_i^a (with $i = 1, 2, ..., 10$) represents the identity likelihood of the estimated albedo for each one of the ten identities stored in the gallery. Similarly, IL_i^s (with $i = 1, 2, ..., 10$) represents the identity likelihood of the estimated shape for each one of the ten identities stored in the gallery.

Now, we can combine both likelihoods using a weighted sum. By experimentation, we found that weights with better identification rates are 0.6 for albedo, and 0.4 for shape. This experimental result can be explained by the following fact: $3D$ shape information of the original face is lost when the $2D$ image is formed. In fact, our fitting approach has to infer a probable shape. On the other hand, albedo which can be considered as $2D$ is recovered with more accuracy. In our experimental results we saw, that in some cases, the values of the cosine measured between the estimated shape vector and the shape vectors from the gallery, were very similar. These similar values of the cosine can produce confusion in the decision of the identity based only on the shape. Therefore, we considered that using probability functions instead of the cosine values is a more appropriate way to obtain a correct decision of the identity.

The conditional likelihoods IL_i^a and IL_i^s, for albedo and shape respectively, can be combined to obtain a single likelihood IL_i:

$$IL_i = 0.6(IL_i^a) + 0.4(IL_i^s) \tag{31}$$

For instance, if the face of the test image corresponds to the identity $i = 2$, then, we would expect a higher value for IL_2 (theoretically $IL_2 = 1$) with respect to the values for IL_i with $i = 1, 3, 4, 5, 6, 7, 8, 9, 10$ (theoretically 0).
In order to show the probability of the algorithm to select the correct identity under a specific illumination, we have used (from Group 1) ten alignments for test images of individuals $i = 1, 2, 3, ..., 10$ under the same illumination. Then, for each alignment a single value IL_i (with i being the test identity) was computed. For each illumination an average of the IL_i values was computed and plotted in Figure 10 (b). The little vertical segments represent the associated standard deviation. We see that the mean IL is greater when lighting is frontal to the face (illumination 5). Figure 10 (c) shows the identification rate for each illumination. The identification rate for each illumination is computed by summing the number of correct identifications and dividing this result by the total number of alignments for that specific illumination. In this graph we have plotted the identification rate computed for Group 1, Group 2, and Group 3 of alignments.

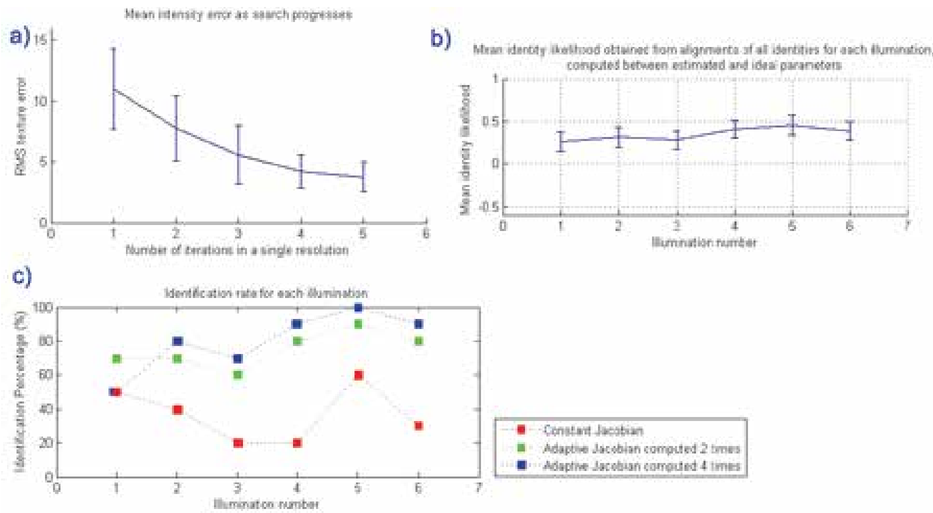


Fig. 10. a) Evolution of RMS error in intensity difference. b) Average (over the 10 identities) of the identity likelihood measured between estimated and ideal parameters. c) Identification rates for each one of the six illuminations.

In the case of fitting with four computations of an adaptive Jacobian we see the worst identification rate (50%) with the illumination number 1, and the best identification rate (100%) using the illumination number 5 which is nearly frontal to the face. A similar relation among identification rates for all the six lightings is conserved for the case of fitting with two computations of the adaptive Jacobian (plot in the middle). The phenomenon is repeated again for the case of fitting with a constant Jacobian (plot in the bottom). Anyway, we can see an important improvement on the quality of the reconstructions when the adaptive Jacobian is computed more times.
In a similar way, we have evaluated the fitting algorithm now trained with the 20 identities from set B.
For this test we used 6 images (with a size of 320×240 pixels) with the same pose and different lighting for each one of the 20 individuals from the set B. Hence, our test set is composed by 120 real images. Again, all images have the pose number 6 which presents a similar angle in azimuth to the left and elevation up. This pose has an angle of 24 degrees from the camera

axis. We choose the same 6 different illuminations for using with each one of the identities. See table 1.
Over the test set (the 120 images) we performed 360 alignments distributed within the following groups:

1. Group 1B: 120 alignments using the fitting algorithm programmed with 4 computations of the adaptive Jacobian.
2. Group 2B: 120 alignments using the fitting algorithm programmed with 2 computations of the adaptive Jacobian.
3. Group 3B: 120 alignments using the fitting algorithm programmed with a constant Jacobian.

Figure 11 shows the identification rate for each illumination. In this graph we have plotted the identification rate computed for Group 1B, Group 2B, and Group 3B of alignments.

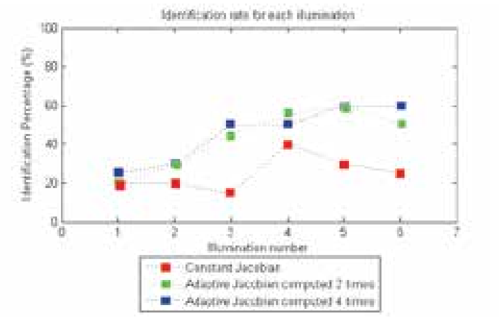


Fig. 11. Identification rates for each one of the six illuminations

In the case of fitting with four computations of an adaptive Jacobian we see the worst identification rate (25%) with the illumination number 1, and the best identification rate (60%) with the illumination number 5 and 6 which are nearly frontal to the face. In a similar way as in the case of experiments for set A, a similar relation among identification rates for all the six lightings is conserved for the case of fitting with two computations of the adaptive Jacobian (plot in the middle). Again, we can see an important improvement on the quality of the reconstructions when the adaptive Jacobian is computed more times.
In this test we used a training set of 20 individuals. In a similar way as in all experiments, model parameters have been limited to 9 shape parameters and 9 albedo parameters.
We used Principal Component Analysis for reducing the dimensionality of shape and albedo variation. Model parameters of shape and albedo are weights of a weighted sum of eigenvectors, see 4 and 5. Eigenvectors of shape or albedo represent variation modes (20 modes) and they are sorted according to their associated variances, from the higher to the lower value of these variances. Each variance associated to each eigenvector represents the relevance of the eigenvector into the weighted sum. The greater the variance, the more relevant the eigenvector (variation mode).
In order to reduce the dimensionality of the training set and using the same number of parameters, we have taking into account only the first 9 relevant eigenvectors of shape and albedo. We can compute the percentage of total variance that can be represented by the model using only 9 parameters of shape as

$$\Xi_{\sigma^2} = \frac{\sum_{i=1}^{9} \sigma_i^2}{\sum_{i=1}^{20} \sigma_i^2} \times (100) = 83.5\% \qquad (32)$$

where Ξ_{σ^2} is the shape representation capability of the model relative to the training set. In a similar way, since we only used the first 9 eigenvectors of albedo, we can compute the percentage of total variance that can be represented by the model using only 9 parameters of albedo as

$$\Xi_{\eta^2} = \frac{\sum_{i=1}^{9} \eta_i}{\sum_{i=1}^{20} \eta_i} \times (100) = 87.1\% \tag{33}$$

where Ξ_{η^2} is the albedo representation capability of the model relative to the training set. These percentages are lower than those corresponding to the experiments using 10 training individuals where the number of model parameters of shape and albedo is also 9 and 9 respectively, where model parameters cover 100% of the total variance. Therefore, we can conclude that using a bigger set of training faces while keeping a fixed number of model parameters decrements the ability of representing the 100% of shape and albedo variation contained into the training set. In turn, that conclusion explains the lower identification rates observed on Figure 11 with respect to those observed on Figure 10 (c).
Figure 12 illustrates the difference in fitting with a constant Jacobian in contrast to fit with an adaptive one. Here we show reconstructions for two different lightings.

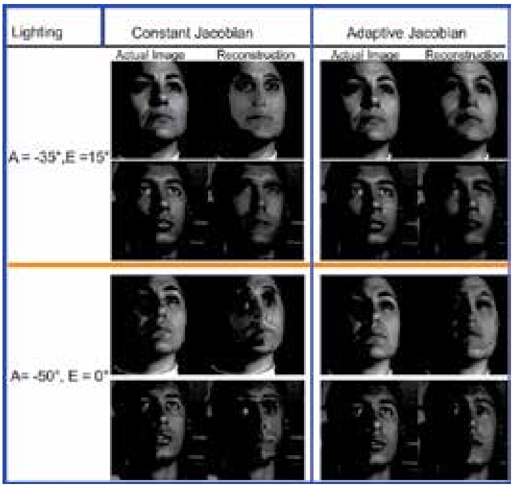


Fig. 12. Reconstructions of two individuals from set B under two different lightings. The reconstructions obtained with the fitting algorithm which uses an adaptive Jacobian are visually better than those obtained from using a fitting algorithm which uses a constant Jacobian.

5.4 Face alignment of faces not included into the training set: Fitting novel faces

The $3D - IAAM$ model trained with the set of 20 individuals has been tested for fitting to novel faces not contained within the training set. Again, we used 33 model parameters: 6 for 3D pose, 9 for 3D shape, 9 for illumination, and 9 for albedo.
We selected 5 individuals not contained within the training set and captured in 3 poses each one (-24ž, 0, and +24ž with respect to the camera axis). For all the images, the 3D pose only varies in azimuth: -25, 0, and 25 degrees with respect to the camera axis. Figures 13, 14, 15, 16, and 17 show alignments for novel faces take from the extended Yale B database and originally numbered as 18,25,35,36. The fifth face belongs to the author of this work.

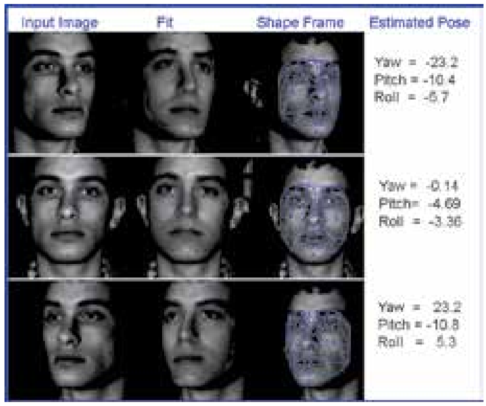


Fig. 13. Alignments for the face number 18 from the extended Yale B database. This face is not included in the training set. The recovered 3D pose angles are specified in degrees

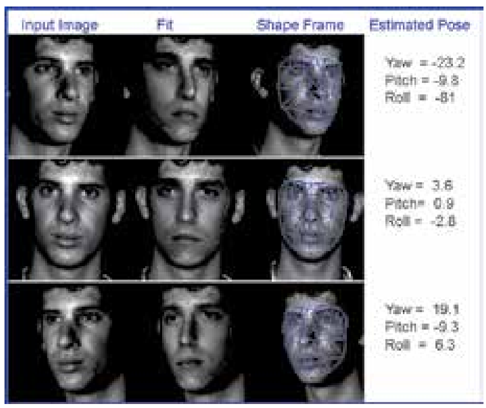


Fig. 14. Alignments for the face number 25 from the extended Yale B database. This face is not included in the training set. The recovered 3D pose angles are specified in degrees

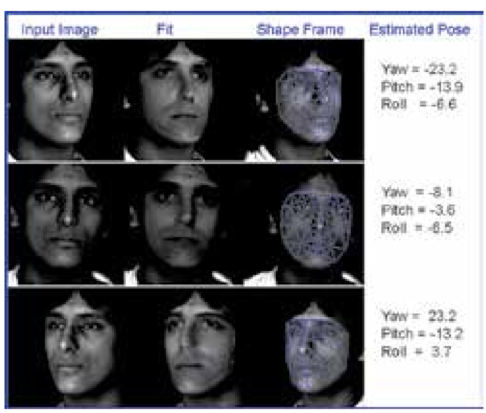


Fig. 15. Alignments for the face number 35 from the extended Yale B database. This face is not included in the training set. The recovered 3D pose angles are specified in degrees

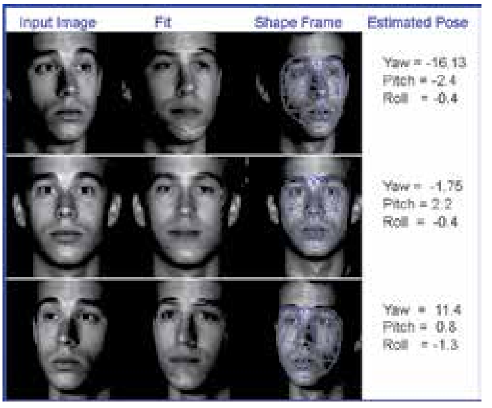


Fig. 16. Alignments for the face number 36 from the extended Yale B database. This face is not included in the training set. The recovered 3D pose angles are specified in degrees

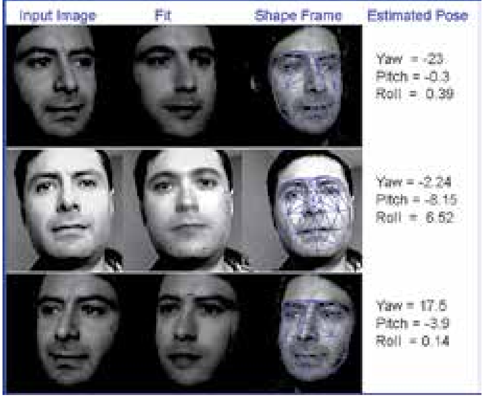


Fig. 17. Alignments for the author's face. This face is not included in the training set. The recovered 3D pose angles are specified in degrees

The experiments performed over faces not included within the training set give us with signs about the ability of the method for adapting to novel faces, and also demonstrate that it is possible to estimate relevant information about a new face. That information is provided at the end of the fitting process, and is delivered to us through the model parameters. We think that a possible generalization of the method consisting on the capability of fitting any human face may be feasible. The solution will be based on making a careful and systematic selection of the training faces according to desired characteristics. That could be another research problem.

6. Conclusions

Shape and albedo are estimated more accurately when an adaptive Jacobian is used. The adaptive Jacobian is a way to express the appearance variation produced by parameters variation as a function of the lighting parameters computed in each iteration. Hence, the adaptive Jacobian works better than the constant one when the initial model is different (in lighting) from the test image, as it actually happens in the most of cases. The improvement provided by the use of an adaptive Jacobian was confirmed when we obtained better estimations of shape and albedo whenever we were increasing the times that this Jacobian was computed. On the other hand, we determined that the computational time used in calculating the Jacobian is linear with respect to the number of times that this Jacobian is computed. In contrast, the improvement in the recovery of the parameters was not significative when the Jacobian was computed more than two times. Therefore, we conclude that four computations of the Jacobian is sufficient to obtain acceptable reconstructions. On the other hand, the capability of the fitting algorithm to reconstruct novel faces not contained within the training set was demonstrated in this chapter. Finally, our proposed interpretation approach not only provides information from a face image, also it is capable of creating new information by reconstructing unseen novel views of a recovered face. This work has addressed the problem of automatic and fast interpretation of a face which exhibits any pose and any lighting. Modern approaches have important limitations regarding processing speed, fully automatic operation, 3D, lighting invariant and simultaneous handling of multiple appearance variation sources. We introduced a novel and fast method for automatic interpretation of face images from a single image. Pose, shape, albedo, and lighting are sources of appearance variation which modify the face image simultaneously. For that reason, trying to estimate only one of these factors without considering the others would produce inaccurate estimates. In order to avoid an inaccurate estimation of each one of these sources of appearance variation, our fitting method estimates simultaneously, in each iteration, the appropriate increments for parameters of 3D shape, 3D pose, albedo and lighting. At the end of the fitting process our proposed algorithm provides us with a compact set of parameters of 3D pose, 3D shape, albedo and lighting which describe the test image. The fitting algorithm is based on *a priori* knowledge of the relationship between the appearance variation (of the model) and the parameters. The appearance variation of the model is produced by changes in pose, shape, albedo and lighting. This appearance variation maintains a non-linear relationship with respect to the model parameters. However, in the case of pose, shape, and albedo, the appearance variation range is sufficiently small so that we can approximate this non-linear relationship with a linear relationship which can be easily learned. On the other hand, the range of appearance variation produced by changes in lighting is unlimited. Then, it is not possible to approximate the appearance variation with respect to the lighting parameters with a simple linear relationship. Fortunately, we found a way to separate lighting from the other sources of appearance variation, in such a way that we can learn a linear relationship between a set of parameters (pose, shape, and albedo) and the appearance variation caused by these parameters. This learned linear relationship is completely independent from lighting. By incorporating a particular lighting to this linear relationship in each iteration of the fitting process, it is possible to reconstruct a new relationship between the full appearance variation and the changes of all the model parameters, i.e. pose, shape, albedo and lighting. This new relationship is represented in our fitting algorithm by the adaptive Jacobian which is reconstructed in each iteration according to the current estimated lighting parameters. Our results, both quantitative and qualitative, show that the method is able to align a 3D deformable model not only in shape but also in albedo, pose and lighting simultaneously. The identification results lead us to think that our approach could be extended to automatic face recognition under arbitrary pose and non-uniform illumination. Besides, the model can synthesize unseen face images of people not used to train the model

7. Future work

In our approach, the process of creating synthetic faces is used during the synthesis of the basis reflectance images created during the training time. This set of resulting images is utilized later for the on-line construction of the Jacobian during the test stage. We could improve the accuracy in the synthesis of lighting by refining the mapping of the normals from the mean model to the new deformed model. Presently, this mapping is purely 2D, but because our shape model is 3D, normals can be reoriented according to the new 3D position of each triangular facet. A more accurate representation of lighting should improve the recovery of 3D shape and albedo, and therefore the identification rate. We think that our method can also be optimized in fitting speed by reducing the times that the Jacobian is updated. According to the initial estimated lighting it would be possible to establish a criterium to determine the minimum necessary number of Jacobian updates, while is preserved an acceptable alignment. Also, a robust face recognition scheme can be implemented if we increase the number of identities for training, in such a way, that they have the enough kinds of extreme variations in shape and albedo for modeling all intermediate possibilities. There are many interesting avenues of feature work. With a careful and systematic selection of the faces for the training set, our method can be extended to a generic person-independent automatic 3D face interpretation system, useful for face recognition in difficult conditions of lighting and pose. Combined with other methods for identification, this kind of generic approach could be a suitable part of a complete biometric system for identity recognition.

8. References

Ayala-Raggi, S., Altamirano-Robles, L., Cruz-Enriquez, J. (2008). Towards an Illumination-Based 3D Active Appearance Model for Fast Face Alignment, *CIARP 2008*, pp. 568-575

Ayala-Raggi, S., Altamirano-Robles, L., Cruz-Enriquez, J. (2009). Recovering 3D Shape and Albedo from a Face Image under Arbitrary Lighting and Pose by Using a 3D Illumination-Based AAM Model, *ICIAR 2009*, Halifax, vol. 5627, pp. 584-593

Basri, R., Jacobs, D.W. (2003). Lambertian Reflectance and Linear Subspaces, In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 218-233

Blanz, V., Vetter, T. (1999). A Morphable Model for the Synthesis of 3D Faces, *Siggraph 1999*, pp. 187-194

Blanz, V., Vetter, T. (2003). Face Recognition Based on Fitting a 3D Morphable Model, In: *IEEE Transaction Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1063–1074

Cootes, T.F., Edwards, G.J., Taylor, C.J. (1998). Active Appearance Models, *ECCV 1998*. LNCS, vol. 1407, pp. 484–498. Springer, Freiburg

Cootes, T.F., Edwards, G.J., Taylor, C.J. (2001). Active Appearance Models, In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 681-685

Dornaika, F., Ahlberg, J. (2003). Fast And Reliable Active Appearance Model Search For 3d Face Tracking, *Proceedings of Mirage 2003*, pp. 10-11. INRIA Rocquencourt, France

Forsyth, D. A., Ponce, J. (2002). Computer Vision: A Modern Approach, *US ed Ed. Prentice Hall*

Edwards, G. J., Taylor, C. J., Cootes T. F. (1998). Interpreting face images using active appearance models, *Proc. 3rd IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pp. 300Ű305

Georghiades, A. S., Kriegman, D.J., Belhumeur, P.N. (1998). Illumination Cones for Recognition under Variable Lighting: Faces, *IEEE CVPR 1998*, pp. 52

Horn, B.K.P., Woodham R.J., Silver (1978). Determining shape and Reflectance Using Multiple Images, In: *A.I. Laboratory Memo 490*, MIT. Cambridge, Mass. (August 1978)

Huang, Y., Lin, S., Li, S.Z., Lu, H., Shum, H.Y. (2004). Face Alignment Under Variable Illumination, *Proceedings of the FGR 2004*, pp. 85-90

Kahraman, F., Gökmen, M., Darkner, S., Larsen, R. (2007). An Active Illumination and Appearance (AIA) Model for Face Alignment, *CVPR 2007*

Kovesi, P. (2005). Shapelets Correlated with Surface Normals Produce Surfaces, *ICCV 2005*, pp. 994-1001

Lee, K.C., Ho, J., Kriegman, D.J. (2001). Nine Points of Light: Acquiring Subspaces for Face Recognition under Variable Lighting, *CVPR 2001*, pp. 519-526

Le Gallou, S., Breton, G., García, C., Séguier, R. (2006). Distance Maps: A Robust Illumination Preprocessing for Active Appearance Models, *VISAPP'06*, vol. 2, pp. 35-40

Matthews, I., Baker, S. (2004). Active Appearance Models Revisited, In: *International Journal on Computer Vision*, vol. 60, pp. 135-164

Ramamoorthi, R., Hanrahan, P. (2001). An Efficient Representation for Irradiance Environment Maps, *Proc. ACM SIGGRAPH*, pp. 497-500

Romdhani, S., Pierrard, J.S., Vetter, T. (2005). 3D Morphable Face Model, a Unified Approach for Analysis and Synthesis of Images, In: *Face Processing: Advanced Modeling and Methods*. Elsevier.

Romdhani, S. (2005). Face image analysis using a multiple feature fitting strategy, In: *Ph.D. dissertation*, Univ. Basel, Basel, Switzerland.

Romdhani, S., Ho, J., Vetter, T., Kriegman, D.J. (2006). Face Recognition Using 3-D Models: Pose and Illumination. *Proceedings of the IEEE*, vol. 94, pp. 1977-1999

Ross, A. (2004). Procrustes analysis, In: *Technical Report*, Department of Computer Science and Engineering, University of South Carolina, SC 29208, www.cse.sc.edu/ songwang/CourseProj/proj2004/ross/ross.pdf.

Sattar, A., Aidarous, Y., Le Gallou, S., Séguier, R. (2007). Face Alignment by 2.5D Active Appearance Model Optimized by Simplex, *ICVS 2007*, Bielefeld University, Germany

Silver, W. (1980). Determining Shape and Reflectance Using Multiple Images, In: *Ph.D. dissertation*, Massachusetts Inst. of Technology, Cambridge.

Woodham, R. J. (1989). Photometric Method for Determining Surface Orientation from Multiple Images, In: *Shape From Shading*, B. K. Horn, Ed. Mit Press Series Of Artificial Intelligence Series. MIT Press, Cambridge, MA, pp. 513-531

Xiao, J., Baker, S., Matthews, I., Kanade, T. (2004). Real-Time Combined 2D+3D Active Appearance Model, *CVPR 2004*, vol. 2, pp. 535-542

Zhang, L., Samaras, D. (2006). Face Recognition from a Single Training Image under Arbitrary Unknown Lighting Using Spherical Harmonics, In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 351-363

5

Processing and Recognising Faces in 3D Images

Eyad Elyan and Daniel C Doolan
School of Computing, Robert Gordon University
United Kingdom

1. Introduction

Face recognition is one of the most active research areas in computer vision, statistical analysis, pattern recognition and machine learning (Huq et al., 2007). Significant progress has been made in the last decade, in particular after the FRVT 2002 (Phillips et al., 2003). For example (O'Toole et al., 2007) showed that face recognition systems surpassed human performance in recognizing faces under different illumination conditions. In spite of recent progress the problem of detecting and recognizing faces in un-controlled biometric environments is still largely unsolved.

The use of other biometric techniques, such as fingerprinting and iris technology appear to be more accurate and popular from a commercial point of view than face recognition (Abate et al., 2007). This is due to the inherent problems with 2D-image based FR systems. These include the viewing point of the face, illumination and variations in facial expression. These problems exhibit a great challenge for such systems and significantly affect performance and accuracy of algorithms.

In an overview of the Face Recognition Grand Challenge (FRGC) Phillips et al. (2006), the authors pointed out some of the new techniques used in Face Recognition that essentially hold the potential to improve performance of automatic face recognition significantly over the results in FRVT 2002. Among these techniques the use of 3D information to improve the recognition rates and overcome the inherent problems of 2D image based face recognition has become a current research trend.

In this chapter we present a novel technique for 3D face recognition using a set of parameters representing the central region of the face. These parameters are essentially vertical and cross sectional profiles and are extracted automatically without any prior knowledge or assumption about the image pose or orientation. In addition, these profiles are stored in terms of their Fourier Coefficients in order to minimize the size of the input data. The algorithm accuracy is validated and verified against two different datasets of 3D images covers a sufficient variety of expression and pose variation. Our computational framework is based on concepts of computational geometry which yield fast and accurate results. Here, our first goal is to automatically allocate the symmetry profile along the face. This is undertaken by means of computing the intersection between the symmetry plane and the facial mesh, results in a planner curve that accurately represents the symmetry profile.

Once the symmetry profile and few features points are allocated, then it is used to align the scanned images within the Cartesian coordinates with the tip of the nose residing at the origin.

Aligning the 3D images within the same Cartesian coordinates makes it possible to compare images against each other. Finally, profile-based comparisons are carried out, where profiles (space curves) of different faces are compared. Here, only part of these profiles are considered for the comparisons. These parts are assumed to be less sensitive to facial expression variation. This chapter is organised as follows: section 2 examines the current state of face recognition research. In section 3 the algorithm for processing 3D facial data and extracting facial features will be presented. Following on from this in section 4 profile-based comparisons will be briefly introduced. Experiments and results will be shown and discussed in the second last section. Finally the conclusions and limitations of our method and direction for future work are presented.

2. Previous work

Formally, face recognition maybe defined as: "given a still or video images of a scene, identify or verify one or more persons in the scene using a stored database of faces" (Zhao et al., 2003). In other words, for any facial recognition system, it is usually initialized with a set or a database of images of known persons. This image repository is usually termed as the "gallery". In a recognition scenario an incoming image of a certain person termed as "probe" is matched against the gallery for recognition purposes. This matching scenario is a one-to-many relation where the probe is matched against the entire set of images in the gallery to find out the best match based on some criterion or threshold.

The very initial step in an automated face recognition system is to detect the face in an image. Although it is very likely that more than one face might exist per image, usually it is assumed that only one face exists per image. Detecting the face, and identifying the region of interest on that face (the region which contains main facial characterises such as eyes, nose, and mouth) is essentially a critical step in order to align the face image with a certain coordinate system (often called image registration), and thereby make comparisons between faces feasible, and more likely to produce accurate results. This step is also important to allow the extraction of some facial features that will be used for comparisons purposes.

2.1 Types of face recognition systems

The vast majority of work that has been done in the area of FR has been based on 2D intensity images. In other words face recognition techniques that are solely based on 2D intensity images, usually acquired by 2D digital cameras. Such systems have several advantages, including the availability of cheap over the counter equipment and wide range of algorithms and existing solutions. Although it is difficult to categorize face recognition systems, it is often found in the literature that they are sometimes categorized based on the type of images used, for example 2D or 3D image-based FR systems. Following (Zhao et al., 2003) 2D-image based FR systems can be broadly categorized as holistic, feature-based or hybrid approach approaches. This categorization of 2D-image based FR system could also be generalized on systems that utilize 3D images.

Holistic approaches use the whole face region as an input to recognition. The work proposed early by Turk & Pentland (1991) serve as a corner stone for holistic-based face recognition approaches which is based on Principal Component Analysis (PCA). This in turn, is a dimensionality reduction technique, which treats the image as a point or a vector in a high dimensional space.

Other methods include the use of spatial-frequency techniques, such as Fourier transformations. In these methods, face images are transformed to the frequency domain, and only the coefficient in the low frequency band are preserved for recognition purposes. These approaches have been successfully applied to face recognition, however, the accuracy of such algorithms drops significantly under pose or light variation. In addition, for real-life applications with large databases, holistic methods may not provide sufficient discriminant information.
Unlike holistic-based methods, feature-based approaches utilize facial features such as eyes, nose, and mouth as an input for recognizing faces. In feature-based approach, the recognition rate is highly dependent on the accuracy of the face and facial feature localizations techniques. An example of such work is demonstrated by (Asteriadis et al., 2009) who used geometrical information to localize faces in images and several facial features, such as eyes, nose and mouth. First the face in an image was detected using the Boosted Cascade Method, then a Distance Vector Field was used to detect facial features such as eyes and mouth were geometric information about each pixel are encoded in the feature space.
Hybrid approaches utilize both holistic and local feature representation of the face images (Su et al., 2009). This technique is inspired by psychophysics and neuroscience literature which shows that human beings perceive faces based on both global and local features (Sinha et al., 2006). An example of such an approach is presented by (Su et al., 2009). In this work, the face image has been globally represented by means of Fourier transform defined by the authors as Global Fourier Feature Vector (GFFV).Gabor wavelets were used to extract local features from faces to form a vector space called Local Gabor Feature Vector (LGFV). Thus, representing each face image by one GFFV and multiple LGFVs. With such form of hybrid representation of the facial data, more diverse discriminatory information was encoded in the feature space.

2.2 Challenges

The accuracy of face recognition systems are significantly affected by various challenges. Although the revision of each of these challenges is beyond the scope of this chapter, it is worth highlighting some of these:

- Illumination is considered as one of the challenges that may hinder the robustness of 2D FR in an unconstrained environment.
- Pose variation is another important issue that has been the subject of extensive research. In spite of the major advances that took place in the past decade, handling varying head poses is still considered as one of the major challenges encountered by face recognition techniques. (See for a good review of various techniques that address this issue (Zhang & Gao, 2009)). Various solutions have been proposed to address this problem. One of the simplest solutions is to enrol the gallery with different images per individual that correspond to various poses. Several experiments show that enrolling more than one image, increases recognition rates. Eigenfaces, self organizing map and convolution network approaches both performed better when five gallery images per person were available rather than just one (Zhang & Gao, 2009). However, such an approach is not always possible due to the difficulties in obtaining suitable image data. In addition, it is quite impossible to represent all face poses in a database. Moreover, enrolling the gallery with multiple images per individual would add computational and storage costs, thereby impacting the overall performance of the system.

- Facial expression variation is still considered as one of the challenging problems for FR it has been estimated that the face could generate up to 55,000 different actions (Zhiliang et al., 2008). Addressing this problem will not only improve recognition rates, but will also have a positive impact on other domains, such as facial modelling, animation, and speech synthesis.
- Occlusions due to other objects such as sunglasses and hats.
- Aging, is considered as one of the main challenges to Face Recognition, as it causes significant alteration to the appearance of faces (Lanitis et al., 2002).

Due to the above challenges, the trend has shifted toward using 3D images in FR systems. It is strongly believed in the research community that using 3D information will improve recognition rates and overcome some of these challenges.

2.3 3D face recognition

(Phillips et al., 2006) point out to some of the new techniques used in face recognition. These include recognition from 3D images, high resolution still images, multiple still images and multi-modal techniques. Such techniques essentially hold the potential to improve performance of automatic face recognition significantly over the results in FRVT 2002. Among these techniques, possibly due to the recent development in 3D capturing devices and in order to overcome inherent problems of 2D-based face recognition systems, the trend is shifting toward utilizing 3D information to improve recognition rates. Here, the recognition is performed by matching the 3D models representing the shape of the faces. It is believed that the 3D representations of facial data will essentially overcome problems such as pose and illumination variations.

3D face recognition is attracting more attention in the recent years due to two important factors. Firstly because of the inherent problems with 2D face recognitions systems that appear to be very sensitive to facial pose variations, variant facial expressions, lighting and illumination. Xu et al (Chenghua et al., 2004) compared 2D intensity images against depth images and from their experiments they concluded that depth maps give a more robust face representation, because intensity images are significantly affected by changes in illumination. Secondly, due to the recent development in 3D acquisition techniques such as 3D scanners, infrared, and other technologies that makes obtaining 3D data acquisition relatively easy and accurate (Bowyer et al., 2006).

Although the utilization of 3D data in the area of face recognition started early (Y. et al., 1989), in comparison with image-based face recognition, the use of 3D information is relatively new in terms of literature, algorithms, commercial applications, and datasets used for experimentations (Bowyer et al., 2004). The number of persons represented in datasets for 3D face recognition experiments didn't reach 100 until 2003 (Bowyer et al., 2006), with little experimentation explicitly incorporating pose and expression variations. In this review paper (Bowyer et al., 2006) the authors surveyed some techniques which reported 100% recognition rates. However, the authors pointed out that this is due the limited size of the databases used. In addition, it was clear from the review that in early work only few published results have dealt with datasets that explicitly incorporate pose and/ or expression variation. In the past few years, this has changed, data sets have become larger and algorithms become more sophisticated. Therefore, it is not surprising to see that recent reported recognition rates are not as high as early work.

Several approaches are used in the literature for 3D face recognition. Some of these are based on the segmentation of the face into meaningful points, lines and regions. Others are considered as model based approaches using information about texture, edges, and colours. Profile-based techniques where multiple profile comparisons are carried out, by which a set of profiles are compared against each other. Such profiles might be symmetry ones, transverse, vertical or even cross-sectional.

Among the existing approaches for addressing 3D face recognition systems is the use of Extended Gaussian Image (EGI). Early work by (Lee, 1990) segment convex regions in a range image based on the sign of the mean and Gaussian curvature, and creates an extended Gaussian image. The matching algorithm is done by correlating the EGIs between the probe and an image in the gallery. The EGI in turn, describes the shape of an object by distribution of surface normal over the object structure.

A 2005 article (Gökberk et al., 2005) compared five approaches to 3D face recognition. They compared methods based on EGI, ICP matching, Range Profile, PCA, and Linear Discriminate Analysis (LDA). They used a database of 160 people consisting of 571 images. They found out that ICP and LDA approaches offer the best performance, although performance is relatively similar among all approaches but PCA.

One of the earliest attempts to utilized depth information was possibly proposed by Gordon (Gordon, 1992) who segmented the face based on curvature description, he then extracted a set of features that describes both the curvature and metric size properties of the face. Thus, each face becomes a point in the feature space and the comparisons were carried out using the nearest neighbouring algorithm.

Nagamine (Nagamine et al., 1992) extracted five feature points and used it to standardize face pose, matching various curves or profiles though the face data. According to this experiment the best recognition rates were achieved using vertical profiles that pass through the central region of the face.

Achermann (Achermann & Bunke, 2000) approached 3D face recognition based on an extension of Hausdorff distance matching. A database of 240 images were used, and 100% recognition rate was reported.

Lee et al. (Lee et al., 2005) Approached the problem based on the curvature values at eight feature points on the face. Using support vector machine for classifications they reported a rank-one recognition rate 96% for a data set representing 100 persons. The feature points were manually allocated.

Mahoor et al. (Mahoor & Abdel-Mottaleb, 2009) presented an approach for 3D face recognition from frontal range data. In this work each image was represented by ridge lines. These are points around the eyes, the nose and the mouth. The lines were defined based on the mean and Gaussian curvature computation, and used for representing the face images. Then for matching ridge images, robust Hausdorff Distance (HD) and ICP were used. It was reported by the authors that ICP outperformed HD in experiments carried out using GavabDB and FRGC 2.0 databases (third experiment) where neutral 3D face images of the FRGC2.0 were used, 58.9% rank-one identification using HD was reported while 91.8% using the ICP. Note that the authors here only used frontal images from GavaDB and FRGC 2.0 databases. In other words, results presented here may change by incorporating more pose and expression variation in the experiments.

Facial profile (vertical, cross-sectional, etc.) were also explored in 3D face recognition. Zhang et al. (Zhang et al., 2006) approached face recognition by utilizing 3D triangulated polygonal meshes. Their approach starts by first identifying the symmetry plane (assuming that the facial data is symmetric), and then by computing the symmetry profile. Based on the mean curvature plot of the facial surface, and symmetry profile, they recovered 3 feature points on the nose area to define what they called facial intrinsic system (namely, the nose tip, nose bridge, and nose point at the lower nose edge), which were used to standardize the faces. For detection purposes the symmetry profile with another two transverse profiles provide a compact representation of the face and were used for comparison purposes. A database of 382 different scans was used consisting of 166 individuals of which 32 individuals have multiple scans, and others have just a single. EER for face authentication with variant facial expression reported was 10.8%. For scans with normal expressions 0.8% EER was reported. The symmetry profiles of two models to be compared are fist registered by mean of ICP algorithm. Then translation is done to make the cheek, forehead, and symmetry profiles coincide in the two models. The comparison is done by a set of sampling points on the corresponding profiles. A semiautomatic pre-processing procedure is used to trim of the non-facial regions in the raw mesh.

3. Automatic feature extraction

One of the main challenges in processing and determining certain facial features for a given raw 3D facial mesh is due to the resulting scanned image, which usually contains unwanted geometry that need to be identified and discarded at a pre-processing stage as shown in Figure 1. In certain applications semi-automatic approaches have been introduced to overcome this problem. For example (BenAbdelkader & Griffin, 2005) used seven manually selected land-mark points. Similarly in (Nagamine et al., 1992) it is necessary to identify several landmark points for example nose tip and eye corners which can then be used to register the face.

Fig. 1. Typical Examples of 3D Face Images

A key component within facial data is the symmetry characteristic that is defined by a symmetry plane which divides the face into two similar halves. Wide range of methods are available in the literature that deals with symmetry detection, in particular for 3D face shapes (Zhang et al., 2006), (Colbry & Stockman, 2007), (Pan et al., 2006)-(Gökberk et al., 2006).
Sun et. al. (Sun & Sherrah, 1997), for example, assume that the symmetry plane passes through the center of mass of a given object and uses Extended Gaussian Image (EGI) based technique to detect reflection, and rotational symmetry of objects. For facial data such an assumption might not hold, especially that 3D facial data acquired by laser scanners might be highly asymmetric since it would contain noise, and undesired geometry such as neck and the shoulder.

Zhang et al. (Zhang et al., 2006) detected the pose of a raw mesh by means of PCA, and then detected the symmetry plane by determining certain facial features (e.g. nose ridge points). They reported that out of 120 images, 117 model were correctly characterized by its symmetry profiles and few feature points along the nose area, with an average processing time of 10 seconds.
Colbry and Stockman (Colbry & Stockman, 2007) identified the symmetry plane of a facial scan by matching that scan with a mirror image of itself using face surface alignment algorithm assuming that pose variation is up to 10 degree in roll and pitch and up to 30 degree in yaw.

3.1 Main Method

In this section we will describe our technique to automatically extract the main facial features. First, we will give some definitions and terminologies that are used from this point on.

3.1.1 Definitions

3D images are either produced as point clouds or polygonal meshes (usually triangular). A point cloud is simply a set of n vertices $V = \{p_i \mid p_i \in R^3, 1 \leq i \leq n\}$. A triangular mesh S on the other hand, includes the set of vertices and adjacency information and is defined as $S = \{V, E, F\}$, where E is a set of edges defined as $E = \{(p_i, p_j) | p_i, p_j \in V\}$ and F is a set of facets defined as $F = \{(p_i, p_j, p_k) | p_i, p_j, p_k \in V\}$. The Euclidean distance between two points v_1, v_2 denoted by $v_1 = (x_1, y_1, z_1)$, and $v_2 = (x_2, y_2, z_2)$ is defined as $d(v_1, v_2) = ||v_1 - v_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$. If we Let $f_i \in F$ be a facet on the surface mesh defined by the triplets $f_i = \{v_0, v_1, v_2\}$ then the circumference of the f_i is defined as $d(f_i) = d(v_0, v_1) + d(v_1, v_2) + d(v_2, v_3)$. Based on this arrangement we could approximate the tolerance value of the surface mesh as,

$$S_t = \frac{1}{C} \sum_{i=1}^{nf} d(f_i) \tag{1}$$

where nf represents the number of facets in the triangular mesh, and $d(f_i)$ is the circumference of the i^{th} facet and C is a constant computed based on an average estimation of the number of common edges between adjacent facets on the surface mesh. A normalized and registered raw mesh means that all values of the vertices are scaled to be in the range between 0.0 and 1.0. In addition, the facial data is aligned with the Cartesian coordinate system, such that the nose tip is located at the origin and the face is looking towards the positive z-axis.
A plane is defined by a point and its normal vector, hence a plane will be denoted in the form of $\prod(p_0, n)$ where p_0 is a point on the plane, and n is its unit normal vector. A reference depth plane is used as the reference for measuring the depth of a given surface point on the mesh. The depth of any point denoted by $p_0 = (x_0, y_0, z_0)$ on the surface mesh is measured as the distance between that point and its projection on the depth plane $\prod$ which is defined as

$$d(p_0, \prod) = \frac{n_x x_0 + n_y y_0 + n_z z_0}{\sqrt{n_x^2 + n_y^2 + n_z^2}} \tag{2}$$

where the normal vector of the plane is defined as $n = (n_x, n_y, n_z)$. A planner curve is defined as a set of points in the 3D space that belongs to the mesh, and intersects a certain plane. The length of a planner curve is defined as $\sum_{i=1}^{m} d_i(v_i + v_{i+1})$, where $d_i(v_i + v_{i+1})$ is the Euclidean distance between the two position points $v_i + v_{i+1}$ where $v_i + v_{i+1}$, are point positions on the planner curve, and v_i, v_m are the first and last point respectively on the curve.

The 3D face is said to be symmetric, if there is a plane, such that the face is invariant under reflection about it. Essentially a symmetry plane will pass through the tip of the nose. Thus, if the tip of the nose and another two position points are identified on the face then one could define the symmetry plane.

The centroid position of a facial surface mesh with vertices is denoted by $c = (c_x, c_y, c_z)$ where $c_x = \frac{1}{n}\sum_{i=1}^{n} x_i, c_y = \frac{1}{n}\sum_{i=1}^{n} y_i, c_z = \frac{1}{n}\sum_{i=1}^{n} z_i$. For a well characterized facial data set, the centroid point of a mesh usually lies within the region of interest which includes the nose, eyes and mouth features. Thus, it is highly unlikely that such a point would lie outside this region, for example near the neck area or the hair. Figure 1 shows various 3D facial scans (Moreno & Sánchez, 2004) with irregular outliers where the above assumption about the centroid position is still true.

3.1.2 Method outline

The tip of the nose is considered as one of the easiest feature points to recover from a facial image. In addition, we assume that the symmetry plane of the face passes through the tip of the nose. For human faces this is a very reasonable assumption which is widely accepted in the research community (Zhang et al., 2006), (Colbry & Stockman, 2007), (Sun & Sherrah, 1997). Our methodology is focused on identifying the symmetry plane based on the determination of the tip of the nose. The basic structure of the proposed algorithm is as follows,

- The central region of a 3D scan is initially approximated based on the center of mass and some extreme points.
- The tip of the nose is determined as the point on the facial surface with maximum perpendicular distance from a certain depth plane.
- The symmetry plane that passes through the pre-determined nose tip is then determined.
- A planner curve that accurately represents the symmetry profile is then extracted.
- Some feature points are then automatically determined on the symmetry profile. These feature points include the nose bridge and lower part of the nose.
- The central region is then extracted, based on approximating the positions of the outer corners of the eyes.

3.1.3 Nose tip identification

The first step in this process is to identify the tip of the nose. This is considered as the easiest point to recover on a facial scan. In order to determine this point, we fit a bilinear blended Coon's surface patch. Coon's patch is simply a parametric surface defined by four boundary curves(Farin & Hansford, 1999). The four boundaries of the Coon's patch are determined based on the boundary curves that enclose an approximated central region of the face.

In order to approximate the region of interest we take the centroid and all points that lie within a pre-determined distance from that point. It is important to highlight that the central region identified here is not an accurate representation of central region of the face. Rather it is an approximation which can be used to identify a "minimum" region of the face which can provide a smooth boundary on which it includes certain facial features, in particular the nose region. Once this region is approximated, its boundary is sorted and organized so that it represents the four boundary curves of a Coon's patch. Finally, a surface patch within the boundary curves is interpolated based on Coon's patch definition (see (Farin & Hansford, 1999) for more information).

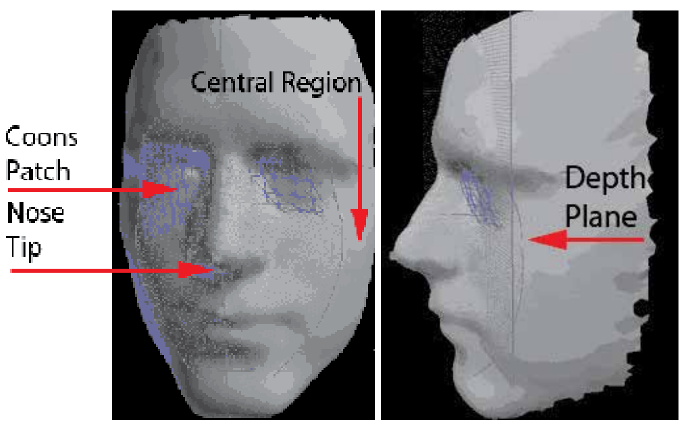
Having the Coon's surface generated as a reference to the facial points on an approximated central region, it becomes straightforward to recover an initial estimation of the nose tip as the one with the maximum depth from the patch. If we let $'$ to denote the set of all vertices within the approximated region of interest of the facial data and let C denote the set of vertices of the Coon's surface patch, then the initial approximation of the nose tip could be formulated as follows,

$$NTIP_{init} = \max\{d(p_i, e_j) : \forall p_i \in \forall', e_j \in C\} \tag{3}$$

Provided that, $e_j = \min\{d(p_i, e_j) : \forall e_j \in C\}$. Since the Coon's surface is composed of relatively small number of vertices in order to keep computation to a minimum, the above formulation only gives an approximation of the nose tip position. To improve our approximation we fit a plane using the points e_j recovered in Equation 3 and its neighbors e_{j0}, e_{j1} and compute the nose tip position as the point with maximum depth from the constructed plane. Figure 2(b) illustrates this concept. Assuming that the nose tip is denoted by N_{TIP} , the constructed depth plane fitted is defined as $\prod_{depth}$ and n is the normal unit vector to the plane, then the tip of nose is formulated as follows,

$$N_{TIP} = \max\{d(v_i, \prod_{depth}) : \forall v_i \in \forall'\} \tag{4}$$

where $d(v_i, \prod_{depth})$ is the Euclidean distance between a point v_i on the surface of the face and the constructed depth plane $\prod_{depth}$.



(a) (b)

Fig. 2. Nose tip identification. (a) Initial estimation of nose tip based on depth measured relative to the Coon's patch. (b) Improving accuracy of nose tip positing based on fitting a plane

This procedure enables us to neutralize the facial data with the tip of the nose residing at the origin of a right hand coordinate system. In addition, the facial data can now be transformed in the Cartesian coordinate system with a rotation vector r defined by two points $N_{TIP}, Nproj_u$

that respectively represents the nose tip and its projection on the depth plane with normal unit vector u. Thus, once we identify the nose tip correctly, we then rotate the facial data such that r becomes aligned with the z-axis of the Cartesian coordinate system.

3.1.4 Symmetry plane detection

To identify the symmetry plane, we assume that N_{TIP} point lies on the symmetry plane. In addition, we let a point p_{s1} be any arbitrary point that lies on the depth plane such that $n_s = (N_{TIP} - p_{s1}) \times (N_{TIP} - Nproj_u))$ where $(N_{TIP} - p_{s1}), (N_{TIP} - Nproj_u))$ are two vectors such that $Nproj_u$ is the projection of N_{TIP} into the depth plane and n_s is the normal unit vector resulting from their cross-product. Figure 3 illustrates this arrangement. Clearly both depth plane and the initial symmetry plane with normal n_s are perpendicular to each other. Assuming that the initial symmetry plane defined by the point p_{s1} and its normal unit vector n_s denoted as $\prod(p_{s1}, n_s)$ and recalling that p_{s1} is one of the points lying on the depth plane then we make the following observations:

1. For a human face, the height dimension of the face is greater than its width.
2. It is clear that if the upper part of the face was considered, and the initial symmetry plane was rotated around the z-axis, then the planner curve that is identified as the intersection between the facial points and the initial symmetry plane with the minimum length will be the symmetry profile.

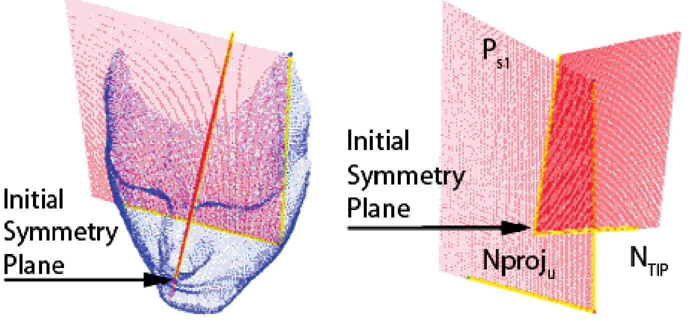


Fig. 3. Symmetry plane identification. (a) Facial surface with depth plane, and initial symmetry plane. (b) Initial symmetry plane results from the nose tip, its projection into depth plane, and an arbitrary point on the depth plane

Based on this arrangement, the initial symmetry plane is rotated by 2π around the z-axis and computation is performed to verify the correct allocation of the symmetry plane. In order to perform the rotation, we compute an angle θ where θ is the angle by which the symmetry plane should be rotated each time. Recall that, the facial data is already aligned within the Cartesian coordinate system with the nose tip residing at the origin. Therefore, if we assume that the initial symmetry plane is defined by the three points $N_{TIP}, p_{s1}, Nproj_u$ and recalling that p_{s1} is one point on the depth plane then θ could be defined as $\theta = \cos^{-1}(\frac{d^2}{S_t^2 + d^2})$, where $d = d(p_{s1}, NTIP)$, and S_t is the tolerance value of the mesh. Defining θ to be dependent on the tolerance value of the mesh makes it more accurate regardless of how the meshes vary in terms of their density. In addition, rotation of the initial symmetry plane based on a very small value for θ minimizes the error value of the detected symmetry plane. Based on θ, the number of rotations that need to be performed is then approximated as $n = \frac{2\pi}{\theta}$, Working out the degree of rotations and the number of rations to validate the symmetry plane, the algorithm proceeds as shown in Algorithm 1. Figure 4, provides an illustration for symmetry plane detection algorithm.

Algorithm 1: Approximating Symmetry Plane

Let $height = -100.0, length = 100.0$ Let V' be a subset of the facial data that represents an approximated central region of the face. ;

while *Number of rotations* $\leq n$ **do**

Find $v_0, v_1 \in V'$ such that they both intersect initial symmetry plane at both ends of the central region.;

Let v_{0p}, v_{1p} be the projected point of v_0, v_1 respectively into the depth plane, and construct an initial symmetry plane based on the three points v_{0p}, N_{TIP}, v_{1p}. ;

Find the planner curve $p(l)$ that is resulting from the intersection of the Facial points of the central region with the initial symmetry plane, and let p_{lenght} be the length of its upper part. ;

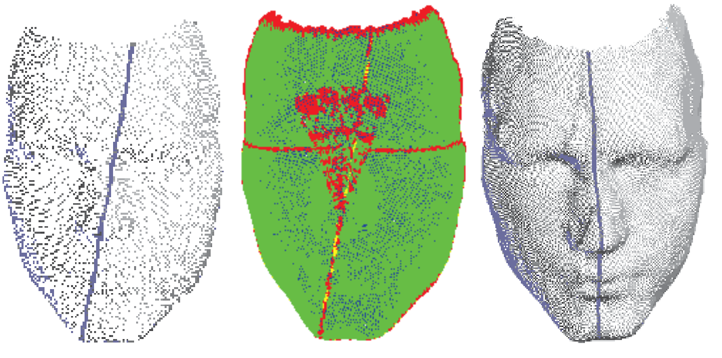
if $d(v_{0p}, v_{1p} > height \text{ and } P_{length})$ **then**

set $height = d(v_{0p}, v_{1p}), length = P_{length}$ and store v_0, v_1 , as possible candidate for symmetry plane points.;

end

rotate the initial symmetry plane by θ.;

end



(a) (b) (c)

Fig. 4. Improving symmetry plane identification. (a) initial symmetry plane, (b) rotating symmetry plane and computing length of the curve. (c) the final symmetry profile

In order to analyze the symmetry profile extracted from the facial data, we fit a spline of the form $P_i = \sum C_i B_i$ where $B - i$ is a cubic polynomial and C_i are the corresponding control points. This process of curve fitting to the extracted discrete symmetry profile data enables us to have a smooth curve passing through the discrete data. Once we have a smooth symmetry profile, we analyze the profile by identifying local extreme points that corresponds to the nose bridge and the lower point of the nose Figure 5.

Based on the symmetry profile, a profile that passes through the nose bridge and through the eyes area can be extracted. Figure 5 shows the relation between the cross-sectional eyes profile which passes through the point NB (nose bridge), and the symmetry profile.

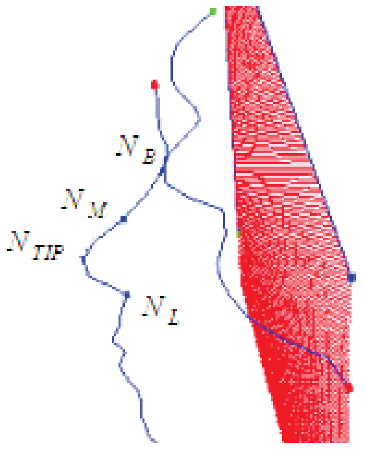


Fig. 5. Symmetry profile analysis

An initial testing of the accuracy of the algorithm for detecting symmetry profile was carried out based on reflective symmetry. Figure 6 illustrates this approach, which is similar to the one used in (Gökberk et al., 2005) to detect a symmetry plane. As shown in Figure 6, it is assumed that $n = (n_x, n_y, n_z)$ is the unit normal vector of the detected symmetry plane. If we define a set of points $P = \{p_i\}$ to be the set of vertices that exist at one side of the symmetry profile, and reflect these points around the symmetry plane, another set of Points Q, will be obtained. Assuming the facial data is perfectly symmetric and the identified symmetry plane is the correct one, then the average mid points of each point and its image $\{p_i, q_i\}$ which is denoted by m_i could be computed using the parametric equation of line $m = p_i + \alpha(q_i - p_i)$ where $\alpha = 0.5$ and the average error value is computed as $Err = \frac{1}{n}\sum_1^n d(\prod, m_i)$, where $d(\prod, m_i)$ is the distance between the i^{th} mid point and the detected symmetry plane $\prod$ in the Euclidean space.

Figure 7 shows some extreme examples of the results of our algorithm on a set of images taken from (Moreno & Sánchez, 2004). In the following section this technique will be further tested by using the resulting features in comparing face images.

4. Profile-based face recognition

Scanned images can be of different poses within the coordinate system. Thus, in order to carry out comparisons between these different scans, the scanned images have to be properly aligned within the Cartesian coordinate. This process is carried out automatically by relying in the proposed algorithm discussed in the previous section. Three feature points namely the nose tip, Nose Bridge, and the lower edge of the nose are used to align the scanned image within the Cartesian product. It is important to stress that the identification of these feature

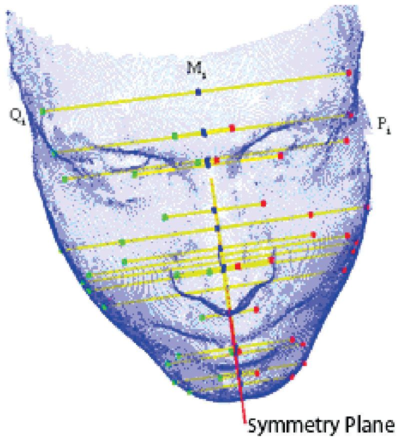
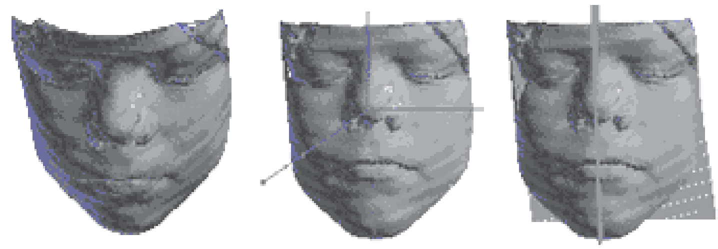


Fig. 6. Symmetry accuracy by calculating the midpoint between a surface point and its image around the symmetry plane

Fig. 7. Visualizing correct identification of symmetry profile in different sample images

points on the symmetry profile is an approximation, in other words the allocated points may not be very precise. However they are good enough for matching and registration purposes as will be discussed and validated in the following section. The alignment of the images is done by carrying out a rigid transformation of the dataset of the 3D points that make the image. The transformation is carried out based on the symmetry profile and the nose tip and is composed of a series of simple translations and rotations to end up with an image aligned within the Cartesian coordinate with the nose tip residing at the origin and facing the positive Z-direction as shown in Figure 8.

For comparisons purposes, it is important to point out that some facial regions are considered more rigid and less sensitive to facial expression variation than others. Nose region for instance, is considered relatively rigid compared with other regions such as the mouth. For profile-based face recognition, the sensitivity of facial regions is even increasing and hence seriously affecting the recognition accuracy, because regions are represented by space profiles. The lower part of the symmetry profile for example is highly sensitive to facial expression variations, while it is more rigid within the area bounded by NL and NB as shown in Figure 9.



(a) (b) (c)

Fig. 8. Processing and registering 3D images (a) loaded face in arbitrary pose and orientation (b) face is automatically processed and aligned with the nose tip residing at the origin (c) the symmetry plane of the face

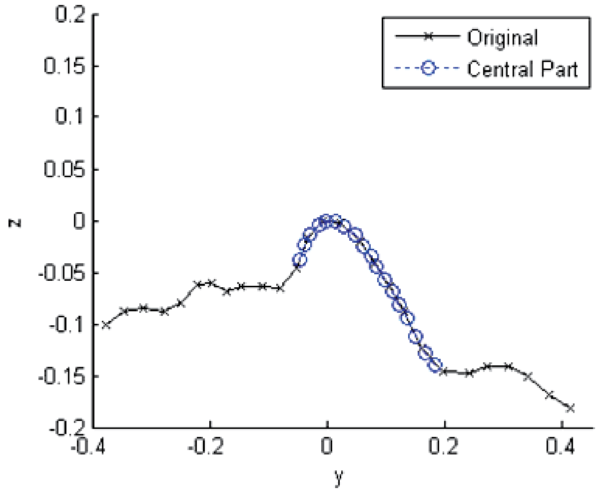


Fig. 9. Central part of the Symmetry profile

Similarly, cross-sectional profles that pass through the eyes area are highly sensitive to facial variations. Thus, it is not reliable to use it for recognition purposes. So for our recognition algorithm we use the central part of the symmetry profile which lies on the nose region (Figure 9) and central part of the cheeks profile. The cheeks profile is simply the profile that crosses the nose area at the mid distance between the points NB and NL. In order to minimize the input data, we compute the Fourier coefficients of the designated profiles and store it in a database, other than storing the actual points of the profile. Thus, having a database of images representing different individuals where each person is represented by two profiles stored by means of their Fourier's. In real time the database file would be loaded into memory and the profiles would be reconstructed according to the general form of Fourier series expansion Equation 5.

$$f(t) = \frac{1}{2}a_0 + \sum_{n=1}^{M} a_n \cos(nt) + b_n \sin(nt) \tag{5}$$

M is chosen to be relatively small, such that the number of coefficients required to reconstruct the curve is relatively much smaller than the number of 3D points that represent the profile. Matching faces against each other is carried out by a profile-by-profile comparisons with closest match selected. The comparison of the profiles is done point-by-point in the 3D space similar to (Zhang et al., 2006). If we let Lp and Lg be two profiles representing the central part of the symmetry profile of a probe and an image in the gallery respectively as shown in Figure 10.

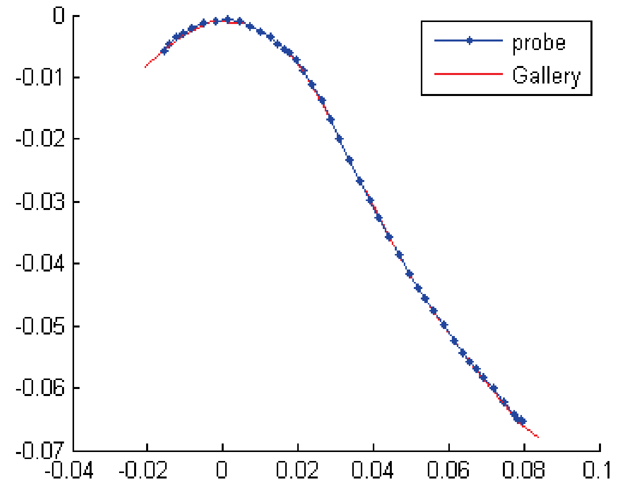


Fig. 10. Profile comparisons

Clearly the distance between the two polylines is directional, in other words the mean distance between the two profiles L_p to L_g is not necessarily the same as the distance from L_g to L_p. These distances are defined as

$$d_{pg} = \frac{1}{n} \sum_{p1 \in L_p} \min_{p2 \in L_g} d(p_1, p_2) \tag{6}$$

$$d_{pg} = \frac{1}{m} \sum_{p2 \in L_p} \min_{p1 \in L_p} d(p_2, p_1) \tag{7}$$

where n, and m represent the number of positions points on the profiles L_p and L_g , and $d(p_1, p_2)$ is the Euclidean distance between p_1, p_2. Thus, the similarity measure between the two profiles can be formulated as

$$E = \frac{1}{2}(d_{pg} + d_{gp}) \tag{8}$$

Based on Equation 8 the measure between two images is computed as

$$E_{total} = E_{cs} + E_{cc} \tag{9}$$

where E_{cs} represents the similarity measure between the central part of the symmetry profiles of the two images, and E_{cc} represents the similarity between the central part of the cheeks profiles of the two images, and both measures are computed as in Equation 8.

5. Experiments and results

For experimental purposes a 3D platform has been developed using Microsoft Foundation Classes (MFC), c++ and openGL. The platform is used to load 3D images, carry out the features extraction and conduct the matching algorithm to search for the best match in Database (Figure 11). In testing our processing and matching algorithm two experiments were carried out using two different databases.

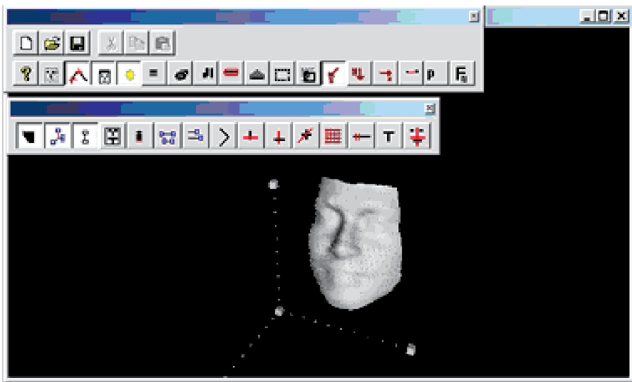


Fig. 11. Software for loading, processing and recognizing 3D Face images

5.1 Experiment 1

In the first experiment a database representing 22 different individuals was used. Each individual in the database is represented by 5 images, each represent a different pose (Figure 12). Only one profile was used to for comparing images, namely the central part of the symmetry profile.

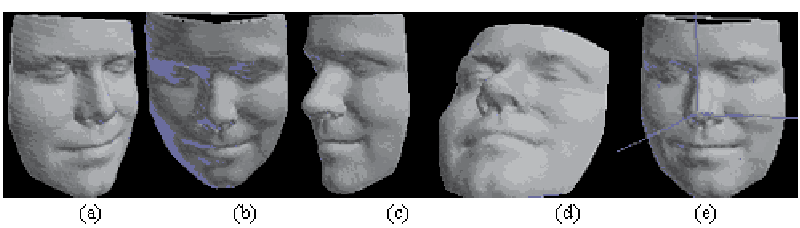


Fig. 12. Facial scans with different poses, (a) pose rotated by degree around the y-axis, (b) rotated by degree around the z-axis, (c) rotated by degree around the y-axis, (d) rotated by degree around the x-axis, and (e) aligned with the Cartesian coordinate.

Figure 13 shows a screen shot for the database file that we used in our experimentation. In this experiment the first line in the file represents the number of images in the database. Individual images are numbered consecutively and each number is followed by the Fourier coefficients representing the central profiles of that person. Hence, in a recognition transaction, an image is loaded into our system, its features are extracted, the pose is aligned within the Cartesian coordinate and finally the database file is loaded into memory and profiles are constructed and compared against the image. In this particular experiment a 100% recognition rate was achieved. This result was expected as the main face variation is due to the pose not to the facial expression variation.

```
22
1.obj
-0.000822  0.022472 -0.043260
-0.001687  0.047064 -0.019793
-0.000342  0.000924 -0.000100
-0.000207  0.004315 -0.000827
-0.000152  0.000727 -0.000179
 0.000127  0.000470 -0.000558
-0.000080  0.000459 -0.000011
-0.000062  0.001190 -0.000574
-0.000033 -0.000401  0.000443
 0.000114  0.000532 -0.000341
```

Fig. 13. Typical database file for 3D images used in the experiments

5.2 Experiment 2

In the second experiment we used gavabDB (Moreno & Sánchez, 2004) which is a public 3D database of human faces. The database covers enough systematic variation in terms of facial poses and facial expressions. Total number of individuals represented in the database is 60, out of these, 45 are male with the remainder being female. Each individual in the database is represented by 9 different images. In our experiment we only consider 7 images per person and discarded two images/person from the database as only part of the face is available in the image such as the left side or right side or the face.

Both profiles (central parts of cheeks and symmetry) are used in this experiment as shown in Figure 14. In total 365 images were tested using our algorithm and were correctly identified, which corresponds to an accuracy recognition rate equal to 86.90%. Inaccurate results were due to the failure of the feature extractions algorithm to standardize the pose and hence extracting the required profiles for comparing the images. In other words 55 different images that were incorrectly identified were actually falsely rejected by the matching algorithm. This raises the False Rejection Rate (FRR) of this experiment to 13.0% which is due to the inaccurately identified features which result in relatively large error value between the profiles of the compared images and result in rejecting the image.

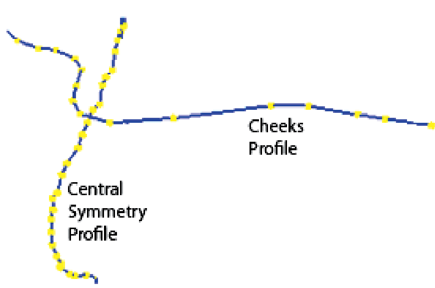


Fig. 14. Two profiles used to compute similarity measure between two face images

6. Conclusions and future work

In this chapter we introduced a new technique for processing 3D images of human faces and extract certain features to be used for recognition purposes. In addition, we have successfully demonstrated that utilizing rigid regions of a human face is very useful in terms of improving recognition rates and minimizing the search space. The average processing time for recognition was 10 seconds. This time includes, loading an image, processing it, extract facial features, standardize the pose, load the database file and conduct the profile comparisons.

Possible improvements to the current recognition system would include improving the features extraction algorithm so that more features points are extracted automatically. In addition, the algorithm should be improved to deal with low quality images. In our experiments the algorithm failed when the images contains holes or spikes, simply because this would lead to false identification of the tip of the nose and would essentially lead to false identification of the rest of the features required.

7. References

Abate, A. F., Nappi, M., Riccio, D. & Sabatino, G. (2007). 2d and 3d face recognition: A survey, *Pattern Recognition Letters* 28(14): 1885 – 1906. Image: Information and Control.

Achermann, B. & Bunke, H. (2000). Classifying range images of human faces with hausdorff distance, *Pattern Recognition, International Conference on* 2: 809–813.

Asteriadis, S., Nikolaidis, N. & Pitas, I. (2009). Facial feature detection using distance vector fields, *Pattern Recognition* 42(7): 1388 – 1398.

BenAbdelkader, C. & Griffin, P. A. (2005). Comparing and combining depth and texture cues for face recognition, *Image and Vision Computing* 23(3): 339 – 352.

Bowyer, K. W., Chang, K. & Flynn, P. (2004). A survey of approaches to three-dimensional face recognition, *17th International Conference on Pattern Recognition*, pp. 358–361.

Bowyer, K. W., Chang, K. & Flynn, P. (2006). A survey of approaches and challenges in 3d and multi-modal 3d + 2d face recognition, *Comput. Vis. Image Underst.* 101(1): 1–15.

Chenghua, X., Yunhong, W., Tieniu, T. & Q., L. Q. A. L. (2004). Depth vs. intensity: which is more important for face recognition?, *17th International Conference on Pattern Recognition*, pp. 342–345.

Colbry, D. & Stockman, G. (2007). Canonical face depth map: A robust 3d representation for face verification, *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1 –7.

Farin, G. & Hansford, D. (1999). Discrete coons patches, *Comput. Aided Geom. Des.* 16: 691–700.

Gökberk, B., Irfanoglu, M. O. & Akarun, L. (2006). 3d shape-based face representation and feature extraction for face recognition, *Image and Vision Computing* 24(8): 857 – 869.

Gökberk, B., Salah, A. A. & Akarun, L. (2005). Rank-based decision fusion for 3d shape-based face recognition, *LNCS 3546: International Conference on Audio and Video-based Biometric Person Authentication (AVBPA 2005*, pp. 1019–1028.

Gordon, G. (1992). Face recognition based on depth and curvature features, *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pp. 808 –810.

Huq, S., Abidi, B., Kong, S. G. & Abidi, M. (2007). A survey on 3d modeling of human faces for face recognition in computational imaging and vision series, *3D Imaging for Safety and Security* 35: 25–67.

Lanitis, A., Taylor, C. & Cootes, T. (2002). Toward automatic simulation of aging effects on face images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24: 442–455.

Lee, J. C. M. E. (1990). Matching range images of human faces, *Third International Conference on Computer Vision*, pp. 722–726.

Lee, Y., Song, H., Yang, U., Shin, H. & Sohn, K. (2005). Local feature based 3d face recoginition, *Lecture notes in Computer Science* 3546: 909–918.

Mahoor, M. H. & Abdel-Mottaleb, M. (2009). Face recognition based on 3d ridge images obtained from range data, *Pattern Recognition* 42(3): 445 – 451.

Moreno, A. B. & Sánchez, A. (2004). GavabDB: a 3D Face Database, *Workshop on Biometrics on the Internet*, Vigo, pp. 77–85.

Nagamine, T., Uemura, T. & Masuda, I. (1992). 3d facial image analysis for human identification, *Pattern Recognition, 1992. Vol.I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on*, pp. 324 –327.

O'Toole, A., Phillips, P., Jiang, F., Ayyad, J., Penard, N. & Abdi, H. (2007). Face recognition algorithms surpass humans matching faces over changes in illumination, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29(9): 1642 –1646.

Pan, G., Wang, Y., Qi, Y. & Wu, Z. (2006). Finding symmetry plane of 3d face shape, *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, Vol. 3, pp. 1143–1146.

Phillips, P., Grother, P., Micheals, R., Blackburn, D., Tabassi, E. & Bone, M. (2003). Face recognition vendor test 2002, *Analysis and Modeling of Faces and Gestures, 2003. AMFG 2003. IEEE International Workshop on*, p. 44.

Phillips, P. J., Flynn, P. J., Scruggs, T., Bowyer, K. W. & Worek, W. (2006). Preliminary face recognition grand challenge results, *in Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pp. 15–24.

Sinha, P., Balas, B., Ostrovsky, Y. & Russell, R. (2006). Face recognition by humans: Nineteen results all computer vision researchers should know about, *Proceedings of the IEEE* 94(11): 1948 –1962.

Su, Y., Shan, S., Chen, X. & Gao, W. (2009). Hierarchical ensemble of global and local classifiers for face recognition, *Image Processing, IEEE Transactions on* 18(8): 1885 –1896.

Sun, C. & Sherrah, J. (1997). 3d symmetry detection using the extended gaussian image, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19(2): 164 –168.

Turk, M. & Pentland, A. (1991). Eigenfaces for recognition, *Journal of cognitive Neuroscience* 3(1): 71–86.

Y., C. J., Lapreste, J. T. & M., R. (1989). Face authentification or recognition by profile extraction from range images, *IEEE Computer Society Workshop on Interpretation of 3D Scenes*, pp. 194–199.

Zhang, L., Razdan, A., Farin, G., Femiani, J., Bae, M. & Lockwood, C. (2006). 3d face authentication and recognition based on bilateral symmetry analysis, *The Visual Computer* 22(1): 43–55.

Zhang, X. & Gao, Y. (2009). Face recognition across pose: A review, *Pattern Recognition* 42(11): 2876 – 2896.

Zhao, W., Chellappa, R., Phillips, P. J. & Rosenfeld, A. (2003). Face recognition: A literature survey, *ACM Comput. Surv.* 35(4): 399–458.

Zhiliang, W., Yaofeng, L. & Xiao, J. (2008). The research of the humanoid robot with facial expressions for emotional interaction, *Intelligent Networks and Intelligent Systems, 2008. ICINIS '08. First International Conference on*, pp. 416 –420.

Part 3

Video and Real-Time Techniques

Real-Time Video Face Recognition for Embedded Devices

Gabriel Costache, Sathish Mangapuram, Alexandru Drimbarean, Petronel Bigioi and Peter Corcoran
Tessera, Galway, Ireland

1. Introduction

This chapter will address the challenges of real-time video face recognition systems implemented in embedded devices. Topics to be covered include: the importance and challenges of video face recognition in real life scenarios, describing a general architecture of a generic video face recognition system and a working solution suitable for recognizing faces in real-time using low complexity devices. Each component of the system will be described together with the system's performance on a database of video samples that resembles real life conditions.

2. Video face recognition

Face recognition remains a very active topic in computer vision and receives attention from a large community of researchers in that discipline. Many reasons feed this interest; the main being the wide range of commercial, law enforcement and security applications that require authentication. The progress made in recent years on the methods and algorithms for data processing as well as the availability of new technologies makes it easier to study these algorithms and turn them into commercially viable product. Biometric based security systems are becoming more popular due to their non-invasive nature and their increasing reliability. Surveillance applications based on face recognition are gaining increasing attention after the United States' 9/11 events and with the ongoing security threats. The Face Recognition Vendor Test (FRVT) (Phillips et al., 2003) includes video face recognition testing starting with the 2002 series of tests.

Recently, face recognition technology was deployed in consumer applications such as organizing a collection of images using the faces present in the images (Picassa; Corcoran & Costache, 2005), prioritizing family members for best capturing conditions when taking pictures, or directly annotating the images as they are captured (Costache et al., 2006).

Video face recognition, compared with more traditional still face recognition, has the main advantage of using multiple instances of the same individual in sequential frames for recognition to occur. In still recognition case, the system has only one input image to make the decision if the person is or is not in the database. If the image is not suitable for recognition (due to face orientation, expression, quality or facial occlusions) the recognition result will most likely be incorrect. In the video image there are multiple frames which can be analyzed in order to have greater recognition accuracy. Even if some frames are not suitable for recognition there is a high probability that some of them will work and the decision made will have a high degree of confidence. Once a face is recognized, it remains recognized in the scene by tracking techniques.

The disadvantage in the video imaging technique is in most cases the quality and size of the input frames are inferior compared to the still images.

2.1 General architecture of a VFR system

Most face recognition systems for still and video image technology follow the same classical workflow:

1. The faces have to be detected in the images.
2. The faces are normalized to the same size and usually same in-plane orientation.
   a. Before or after (2), a pre-processing step tries to minimize the effect of illumination over the face.
3. Features are extracted from the facial region.
4. Test faces are compared with a database of people.

The first difference between the video and still image technology is that video scenarios can use a tracking algorithm together with a detection algorithm in order to keep track of all the faces in the video sequence. Using face tracking combined with face detection has three main advantages:

1. It allows the system to follow the faces across a wide range of variations in pose and lighting where tracking can be done easier than detection.
2. The time and memory requirements of a face tracking algorithm are lower than those of a face detection algorithm. Freed resources can be accessed once a face is detected in a frame. Tracking from that moment forward is a very important aspect when achieving real-time functionality.
3. Once a face in a particular frame is recognized with a high degree of confidence, that particular face does not need to be processed for the next frames. Only track the face and keep the association between the recognized person and the tracked face.

In the classification stage of the video imagery, a history of the recognized face offers greater accuracy than that of a still image.

Figure 1 shows a typical architecture of a video face recognition system.

Below are brief descriptions of each component together with the requirements that need to be satisfied in order to have a robust real-time face recognition system which can be integrated into an embedded device.

2.1.1 Face detection & tracking

The face detection and tracking component is very important in designing the recognition system. The properties of the detection algorithm (detection rate, robustness to variations, speed and memory requirements, etc.) will directly affect the properties of the overall recognition system. It is clear that undetected faces will not be recognized. Also considering the goal of real-time functionality on embedded devices where limited resources are available, spending most of that early on will reduce the application of the other blocks in the diagram in real time.

The main challenges associated with the detection and tracking algorithm are determined by the following factors:

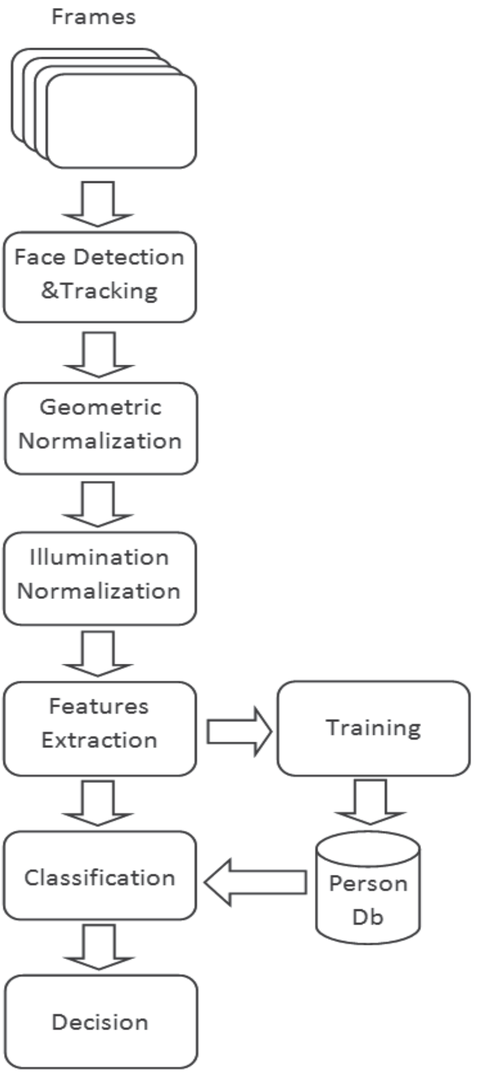


Fig. 1. Architecture of a VFR system

- Face orientation (pose). The appearance of the face may differ in many ways when the orientation of the face changes from frontal to profile or extreme view angles, where face components like the eyes, the nose or the ear may be occluded. It is difficult to detect a face at these extreme angles although face tracking is achievable.
- Changes in facial appearance. Examples include beards, moustaches or glasses. Women may use make-up which can significantly alter the face color and texture. These factors together with the potential for variability in shape, size or color of the face makes face detection challenging.
- Facial expression. The appearance of the face is directly affected by the person's facial expression. Tracking has to be robust to these variations as it is likely to be encountered in normal consumer videos.

- Occlusions. Different components from the face may be occluded in the image by other objects or faces. These have to be addressed by the tracking algorithm.
- Capture conditions. Factors that are involved in capturing the image such as lighting conditions, camera characteristics or quality of the captured image may have a big influence in the detection process.
- Face size or distance to subject. For video face detection and tracking consider the capture resolution and the distance from the capture equipment to the subject. For normal working resolution (qVGA, VGA) the faces can be very small even for relatively short distances.

The detection algorithm should have a high detection rate and robustness to variations such as changes in appearance, capture conditions and face size. The tracking algorithm should improve the robustness to face orientation, expressions and occlusions.

All the above requirements are difficult to fulfil especially for real-time scenarios in embedded devices. In the last few years there has been much progress in this area and now face detection and tracking is a common feature in most consumer cameras and mobile phones.

Tessera's OptiML™ Face Tools Face Tracking and Face Recognition technologies represent a perfect example of state-of-the-art technology in this area.

Some of the relevant parameters of Tessera's Face Tools technology that affect the performance of the overall recognition system include:

- Face tracking for up to 10 faces per frame, with less than 0.1 seconds lock time
- Minimum face size: 14x14 pixels
- Real-time face tracking up to 30 frames per second
- Faces detected in a wide range of orientations including rotation-in-plane and out-of-plane

Together with the detection rate, another metric used to describe the performance of a face detection algorithm is the false positive rate which represents the number of regions falsely reported as faces by the algorithm.

The false positive rate is not as important as the detection rate because the recognition algorithm should be able to differentiate between faces and non-faces when trying to classify the false positive candidates.

2.1.2 Geometric normalization

It is very important to detect and track the faces in all conditions and variations. When comparing local regions between faces, an image registration step must be performed so corresponding facial features are synchronised.

Simple geometric normalization usually involves bringing the faces to a standard size and rotating them in-plane in order to bring the eyes on the same horizontal line. Figure 2 shows some face samples before and after applying the geometric normalization.

More complex normalization scenarios (Corcoran et al., 2006b) can use 3D face models to rotate the face in the out-of-plane space to have identical orientation (i.e only frontal faces). This will have a higher computational requirement and could only be used when there is enough processing power. Figure 3 shows an example of the output of this complex normalization which can help recognition for large pose variations.

All other processing steps applied after geometric normalization should have the same affect on each face.

Fig. 2. Geometric normalization (Before (top), After (bottom))

Fig. 3. Complex geometric normalization

2.1.3 Illumination normalization

If we can control the image capturing environment and impose strict requirements regarding lighting conditions (i.e. control access), recognition accuracy can be improved. In most scenarios where video face recognition is employed, the variations in lighting conditions when the faces are captured can range between dark and bright extremes. The profile face samples for each person to be recognized are captured in very different conditions with still images than those used in video imagery. A pre-processing algorithm should be used to minimize the effect of the lighting conditions when capturing the video images.

Depending on the resources available and capturing conditions, the illumination normalization algorithm can vary from simple algorithms such as: histogram equalization (HE), contrast limited adaptive histogram equalization (CLAHE) (Pizer et al., 1987; Corcoran et al., 2006a), logarithm transformed combined with suppressing DCT coefficients (LogDCT) or retinex (Land, 1986) based approaches, to more complex algorithms that can model the effects of lighting over facial regions (Lee et al., 2001; Smith & Hancock, 2005).

For embedded devices the simple normalization is a good compromise between execution speed and robustness to lighting variations.

Figure 4 displays the output of simple normalization techniques for two images affected by extreme side illumination.

An important issue to be considered when designing an illumination normalization algorithm is the balance between minimizing the effect of illumination and the inherent loss of information; information which is useful for classification. For instance, a face may appear dark because of dark lighting conditions or because the person has dark skin. Usually after normalization this information may not be recovered.

Fig. 4. Simple illumination normalization examples

The validation sets for most of the algorithms that try to minimize the effect of lighting (i.e. Yale database (Georghiades et al.,2001)) consists of faces captured in very different lighting conditions where the normalization algorithms have better results compared with using the original faces without illumination normalization. In real life conditions, the faces can be compared to similar lighting conditions where applying the illumination normalization should not have a negative impact over the recognition results.
It is very important to have a validation set that has a variation distribution close to those most likely to be encountered in the scenarios that the recognition system is designed for.

2.1.4 Feature extraction

Together with the useful information that can be used to differentiate between individuals, the face images described by the pixel values contain redundant information and information that can be ignored in the classification stage. By extracting only the useful information in this step we improve the accuracy of the recognition and also lower the storage requirement for each face.
Below are the main requirements for the feature selection algorithm:

- Good discriminative property. The features need to be able to differentiate between people. This translates into large variations between the value distributions for each person.
- Consistency. Features should not be modified between different images of the same person. This allows for recognition accuracy across large variations. Quantitatively this translates into small variation in the feature distribution for multiple faces of the same person.
- Small size. These features need to be stored and compared. Small size will allow fast comparison and low storage requirement.
- Fast computation. In order to achieve real-time recognition in video images, the faces need to be processed quickly.

The first two requirements will improve the accuracy of the recognition system and the last two requirements will ensure real-time recognition in embedded devices.

Classical approaches for still image recognition were also applied to the video image scenario with good results. These include: Principal Component Analysis (PCA) (Turk & Pentland, 1991), Linear Discriminate Analysis (LDA) (Belhumeur et al., 1996) and Discrete Cosine Transform (DCT) (Podilchuk & Zhang, 1996).
The DCT approach is of particular interest because of the speed of DCT transformation. Most of the capturing devices have DCT already implemented as part of JPEG compression module for storing captured images. More recent approaches like Local Binary Patterns (LBP) (Ojala et al., 2001) and Histogram of Oriented Gradients (HOG) (Lowe, 2004) have been used for face recognition.

2.1.5 Classification

In the case of still image recognition, the system makes a decision if the test face belongs to one of the people in the database and if so, which one (based on comparing the features computed in the previous step for test faces and a database of people).
In the case of video image recognition, the system compares the series of test faces with those in the sample database. Most commonly this is implemented as a series of still images derived comparisons and at each frame the confidence of our decision is modified based on the history of previous comparisons.
Simple classification algorithms like distance between feature vectors are preferred because of their simplicity and speed. More complex learning algorithms can be used if there are enough computation resources.
The classification algorithm is divided into two stages:

1. Training. Prototypes are constructed for each person in the database. The prototypes can be built from single or multiple face samples. Using multiple samples improves the quality of the prototype. The prototype can be represented by a series of feature vectors (such as distance-based classification) or can be represented by statistical models trained with multiple samples (such as learning-based classification algorithms).
2. Testing. Test samples are compared with each person prototype and similarity scores are computed. A decision is made using these similarity scores and the history of previous scores.

If, for a specific scenario, there is a fixed database that does not modify or update on the same platform where recognition is executed, a more complex algorithm for training can be used (i.e. training a learning algorithm) and performed offline. The result of this training algorithm is used during the recognition phase. When the database needs to be updated on-line at any time, the training algorithm needs to be less complex to be run on the embedded device. The result of the training algorithm, either the feature vectors or the person model, needs to be stored in the training database. This will influence the storage requirement of the recognition system.
An interesting algorithm that can be used for video face recognition is to model not only the appearance of the person at each frame but also the transition from frame to frame. A multi-dimensional Hidden Markov Model (HMM) (Nefian & Hayes, 1998) is used in order to model this type of transition. At the moment the complexity of HMMs makes it less favourable for embedded implementation.

2.2 Performance testing

Comparing two face recognition systems is a difficult task because there are many parameters that can describe the performance of a particular recognition system. Usually one system or the other is superior using different sets of parameters.

Depending on the specific application where the recognition system is deployed, some specific performance metrics are more important than others. For example, a security system based on face recognition will have, as a main priority, a very low false acceptance rate, whereas a photo sorting application implemented on an embedded platform based on face recognition, will have its priorities of high recognition rates and low complexity.

The performance of a recognition system can be described by two types of parameters: accuracy parameters that describe how accurate the system is in recognizing faces and technical parameters that represent characteristics such as how fast the system will process a face, etc.

Some of the accuracy parameters that can be used to describe a video face recognition system include:

- Recognition rate. This is the main measurement to describe the accuracy of a recognition system. It represents how many faces are correctly recognized from the total number of faces. For video recognition this is a little more complex as it can be computed as the total number of frames where the faces are recognized.
- False positive rate. For specific applications this parameter can be more important than the recognition rate. This is usually computed as the number of mistakes made by the system. It can be further classified as a false acceptance rate in verification applications where an unknown individual is classified as one person from the database and as a false rejection rate where a person from the database is classified as unknown.
- Receiver Operating Characteristic (ROC) curve. In most cases there is a trade-off between the recognition rate and a false positive rate. For a high recognition rate, tune the recognition system to increase the recognition rate. This will inevitably increase the false positive rate as well and the other way around. The ROC curve represents the recognition rate for each possible false positive rate and only by displaying the ROC curves can a comparison of the two recognition algorithms be made.
- Minimum face size to be detected and recognized. When working with normal video resolution the faces can be very small, even at short distances from the capture equipment. Imposing a high minimum size for the face in order to be recognized can lead to a high rate of faces that are ignored or not recognized in the video images.
- Range of pose variations to recognize a face. Depending on the application to apply the recognition, a higher or lower range of pose variations is needed to recognize the faces.

There are many technical parameters that can be used to describe a video face recognition system including:

- Processing time. This represents the time required to detect, process and classify all faces in a frame. This parameter depends on the platform where the recognition is implemented and will dictate if real-time functionality is available or not. For video frames, the time available for real-time recognition is the time between consecutive frames.
- Memory requirements. This represents the storage requirement for the system and includes the size of the feature vectors, person prototypes and other constants used in the algorithm.
- Number of faces recognized in each frame. The time required for detecting all faces in a frame is constant. This parameter will be influenced by the time required to process and recognize one face after it is detected.

The accuracy parameters depend on the database used for testing and the technical parameters depend on the specific platform where the recognition system is implemented.

Without using the same database and same platform, two recognition systems cannot be compared only by the performance parameters.

3. Proposed recognition system

The goal is to build a video face recognition system running in real-time on low computational power embedded platforms capable of recognizing multiple faces in video sequences. The main use case scenario intended for this system is tagging faces in consumer images as they are captured by digital cameras and mobile phones. The main requirements for the recognition system are high recognition rate, high robustness to variation types and low computational complexity for the algorithms used in the system. For this scenario, the input video stream can vary in size from small (qVGA) to high (full HD). Faces can also vary in size from tens of pixels in width to hundreds of pixels. Large variations in face pose, expression and illumination are also likely to be present.

Tessera's Face Tracking and Detecting technology is used in this experiment (Tessera, 2010). For geometric normalization, a computationally attractive approach is used, which involves: gray-scale transformation of the image, rotation of the face image to align the eyes on horizontal direction and resizing the face image to a small fixed size (i.e. 32x32 pixels). This size will allow for recognition of faces at a range of distances from the camera.

To minimize the effect of lighting variations, use a variant of the retinex (Land, 1986) illumination normalization algorithm. This is done by using a fixed variance matrix computed offline from a large database of images. This approach is very fast to apply and insures that the features computed in the next stage are more robust to large variations in illumination.

The features used for classification, in this chapter, are a variant of the Local Binary Pattern (LBPs) (Ojala et al., 2001) features which have been recently employed, with good results, for face recognition (Ahonen et al., 2006).

The classic approach of using LBP features in face recognition involves computing these features for each pixel in the face image, dividing the face image into small regions (separated or overlapped), and for each region computing the distribution of the LBP values. Often, only a small subset of all the features is used (uniform LBPs) in order to compute the region distribution. The classification involves comparing these distributions between corresponding regions from the test faces and the face samples used to build the prototypes in the training stage.

One approach is completely different. It is based on selecting from all possible features, those features that maximize the two properties defined in Section 2.1.4, namely: consistency and discriminancy. The training stage is split into two stages: 1) Off-line training, using a very large database of faces, in order to determine and select the most consistent features. 2) On-line training, using the face samples in the database that need to be recognized.

The weights for each selected feature are computed in the off-line stage, each weight representing how discriminating the respective features are for the people in the database. Look for the best features that are globally consistent for a very large database (off-line) of people and weight them according to how well they can discriminate for a given database (on-line). Both training stages are presented in the next sections.

For classification a similarity measure between two faces is computed by looking for identical corresponding features in the off-line training stage. For each identical feature value add the similarity between the energy of the features multiplied by the weights computed in the on-line training stage.

3.1 Feature extraction – LBP features

The Local Binary Patterns (LBP) (Ojala et al., 2001) features have been used in the system. These features are computed based on comparing the central pixel with its neighbours, concatenating the binary comparison results and computing the decimal number from the binary string.
Figure 5 shows how the feature is computed from an image.

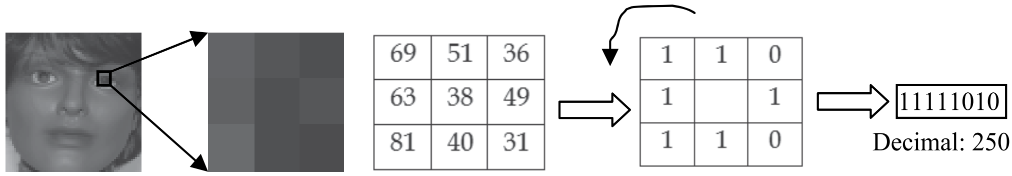


Fig. 5. LBP feature.

The LBP features extract local information from the face region and due to the binary comparisons are robust to changes in illumination.

3.1.1 Multi-scale LBPs and extended LBPs

In order to capture more information from the face region, features are computed at different face resolutions beginning with the standard size used for geometrical normalization, down to smaller scales by downsampling the face image with different factors (i.e. 2, 4 etc). The LBP feature are extracted at each resolution using their 8 closest neighbours together with their extended variants using 8 more distant neighbours. Figure 6 illustrates an example of the normal LBP feature together with its first order extended LBP feature.

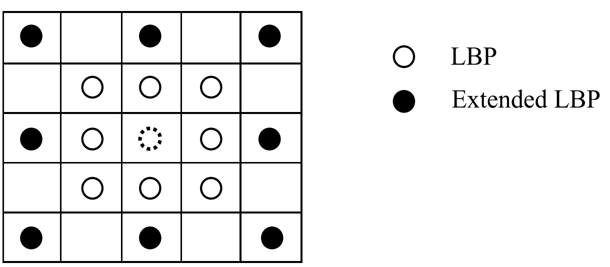


Fig. 6. LBP and Extended LBP

For each pixel in the normalized face image, compute multiple feature values. Do the same for the other scales.

3.1.2 LBP energy

The binary comparisons used for computing the features make them very robust to illumination variations. They also cause loss of information about the similarity of the local regions. For example, a very strong feature will have same value as a very faded feature. For this, calculate the normalized energy of the feature that will be used when comparing identical features for similarity between faces. The energy is computed using the formula:

$$e = \sum I_i^2 / max(I_i^2) \tag{1}$$

where I_i represents the value of the neighbour pixel i used when computing the feature.
The energy is computed for both normal LBP and extended LBP.
Because small size face images are used, the features are not grouped in the face image by dividing the face into regions, but the corresponding features are compared for classification. The features are corresponding if they are computed at the same location, same scale and if they are normal or extended.
The feature vector after this analysis consists of the normal and extended LBPs computed at each location and at each scale together with their energies.

3.2 Off-line training – consistency analysis

Good features are those that do not change between images of the same person. This increases the accuracy of the recognition for different variations of the facial image.
This algorithm ranks a set of features given a large database of facial images. The order of the features is given considering their intra-class variation from low to high. The first features will be the most consistent between faces of same individuals. For recognition these features are more robust to variations.
Assume there is a collection of m people ($P_1, P_2, ...P_m$), each with multiple facial images. For each face compute all N possible features described in the previous section (normal and extended LBPs at all resolutions). Note the feature vector for person P_i image j as:

$$F_{ij} = (F_{ij1}, F_{ij2}... F_{ijN}) \tag{2}$$

The features in the system are the LBP features. This algorithm can be extended to any other type of feature or combinations between features. Below is the general form of the algorithm.
For each feature, define a measure of intra-class consistency $S=(S_1,S_2,\ldots,S_N)$
The steps of the algorithm are:

- Reset all scores.

$$S = (0) \tag{3}$$

- Update scores. For every feature k, for every person i, for every m image of person i:
  - Compare the feature F_{imk} with the same feature k of the remaining j images of person i (F_{ijk} with $j=1:N_i$) where N_i is the number of images for person i.
  - If $|F_{imk} - F_{ijk}| < thr$ then increment S_k

$$S_k = S_k+1 \tag{4}$$

At the end of this process order all features according to their score. Depending on the constraints, either keep a fixed number of features for classification in the latter stage or impose a threshold over the consistency measure.
The $|F_{imk} - F_{ijk}|$ term represents the distance between the feature values. In this case, search for identical features so $thr = 0$. For other types of features, a distance measure needs to be defined and a suitable thr needs to be chosen.
For best results, meaning best globally consistent features, the input database should be very large with all types of variation. Because it is executed off-line it does not affect the speed performance of the overall system.

3.3 On-line training – discriminative analysis

Together with consistency, the features also need to be able to discriminate between the people in the database. The same value for one feature across a database means both perfect consistency and no discriminative power. This algorithm assigns weights to each previously selected feature. The weights determine how well the feature can discriminate between the person in the database and which can be used in the classification stage.

Apply this algorithm to any type of feature using a suitable distance measure. The algorithm description is generic.

Assume m people in the database (P_1, P_2, ...P_m), each with at least one representative facial image. For each face, N representative features selected by the off-line training procedure and their corresponding discriminative scores. For example, for person P_i we have:

- feature vector $F_i=(F_{i1},F_{i2},\ldots,F_{iN})$
- score vector $S_i=(S_{i1},S_{i2},\ldots,S_{iN})$

The steps of the algorithm are:

- Reset all scores.

$$S_i = (0),\ i = 0...m \tag{5}$$

- Update scores. For every feature k, for every person i,
  - Compare the feature F_{ik} with the same feature k of the remaining people (F_{jk} with $j=1:N$)
  - If $|F_{ik} - F_{jk}| >$ thr then increment S_{ki}

$$S_{ki} = S_{ki}+1 \text{ or or } S_{ki} = S_{ki} + |F_{ik} - F_{jk}| \tag{6}$$

In order to make these scores independent of the number of people and faces in the database, normalize them using the maximum sum of scores for a person using the next equation where N represents the number of features and m the number of people in the database:

$$S_{ki} = S_{ki}/max(sum(S_{ln})_{n=1:N})_{l=1:M} \tag{7}$$

The same observations from the previous section for the terms: $|F_{ik} - F_{jk}|$ and thr are valid. In the case of searching for identical LBPs, the parameter $thr = 0$.

3.4 Classification

Having computed the features described in Section 3.1 and the discriminative scores of all features selected in Section 3.2, using the algorithm described in the previous section, compute a similarity (S_{ij}) between two faces (trained face f_i and test face f_j) by counting how many identical corresponding features there are between the two faces using this formula:

$$S_{ij} = \sum w_k * |e_{ik} - e_{jk}| * g_{ij},\ k=1:N \tag{8}$$

where:

- g_{ijk} is equal to 1 if feature k is identical between faces i and j, and 0 otherwise.
- e_{ik}, e_{jk} represent the energy of the feature k from image i and j respectively computed using eq. (1).
- w_k represents the discriminative score for features k of trained face i.

Comparing the test face with all face samples from the trained database, return the most similar person with the test face. By imposing a decision threshold, control the recognition rate versus the false positive rate, depending on the application mentioned in Section 2.2. Once the similarity measure between the test face and the most similar person from the database is higher than the decision threshold, decide that the face is recognized and continue or not the recognition process over the next frames.

4. Results

In order to assess the performance of the recognition system a large database of videos with systematic variations was used, including: pose, illumination, face size/distance to subject, and facial expressions.

In the training stage, a single image was used to train each person. The training face is frontal, good size, normal illumination and good quality. Tests were run for different numbers of people in the database from low (3) to high (100).

For each test, the recognition rate (RR) was measured, as the number of correct classifications, false positive rate (FP2) as wrong classifications and undecided rate (MD) as number of test faces which were not classified.

Figure 7 shows the recognition and error rates as a function of head yaw angle. As specified above, training was conducted at head yaw angles of zero degrees and testing was done with 0˚, 10˚, 20˚ and 30˚ yaw angles.

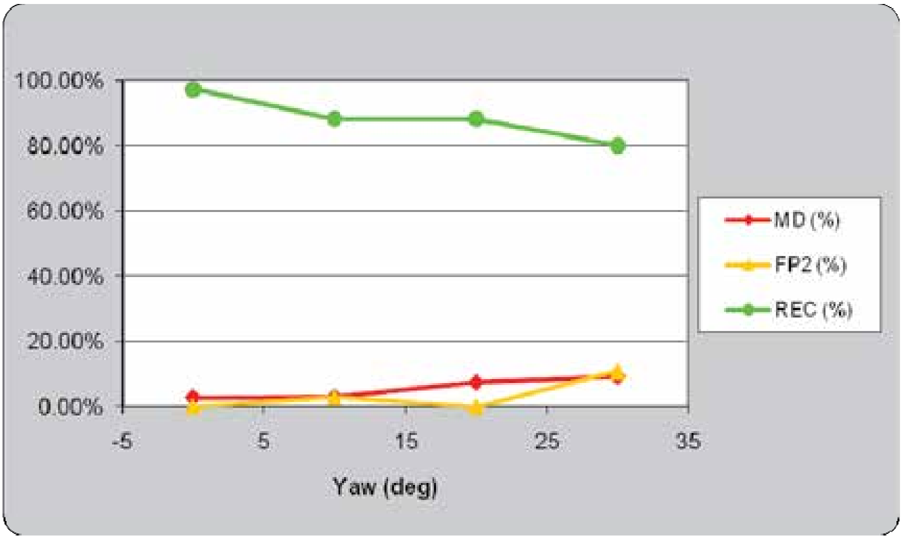


Fig. 7. Recognition performance for different yaw angles

Figure 8 shows the recognition and error rates as a function of head pitch angle. Training was conducted at head pitch angles of zero degrees.

Figure 9 shows the recognition and error rates as a function of different facial expressions. The training faces had no facial expression.

Figure 10 shows the recognition and error rates as a function of different illumination conditions. The approximate EV values for the given conditions are: LowLight (2.4EV) and StrongLight (9EV), which can be considered extreme lighting conditions. Training was conducted using normal indoors ambient lighting.

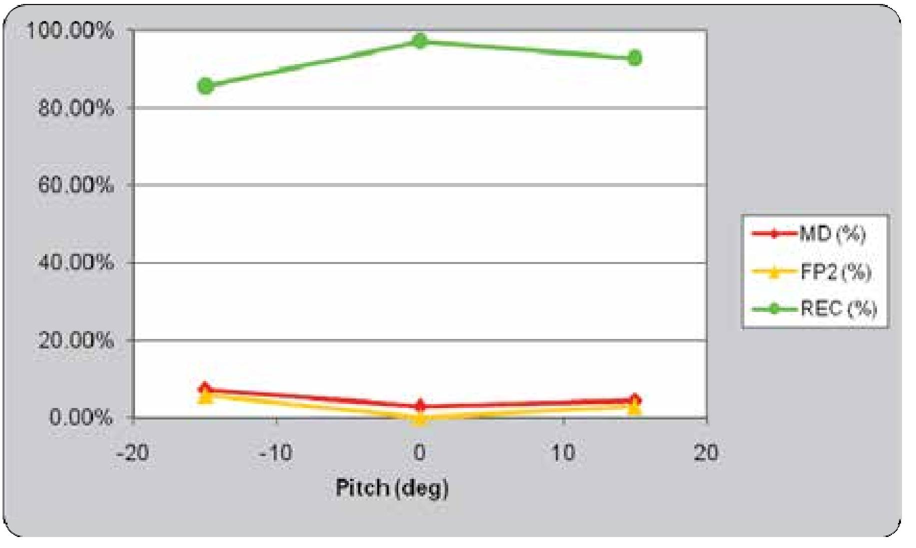


Fig. 8. Recognition performance for different pitch angles

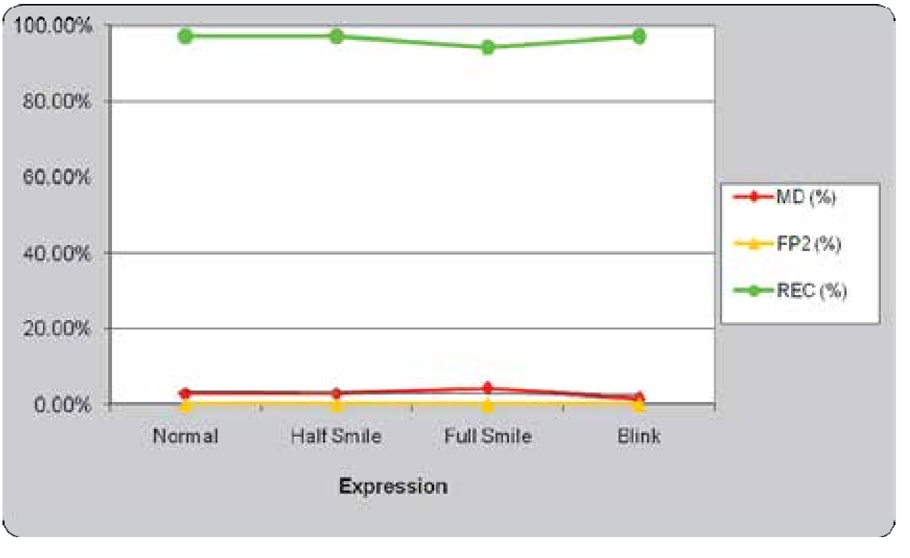


Fig. 9. Recognition performance for different face expressions

The main technical parameters for the system were implemented on an ARM9 platform (266 MHz CPU), the processing time for a face depending on the size of the input frame varied between 8 and 15 milliseconds for qVGA and VGA input frame size which is well within real-time requirements. The size of the features vector for each analyzed face is about 2Kb which is very small.

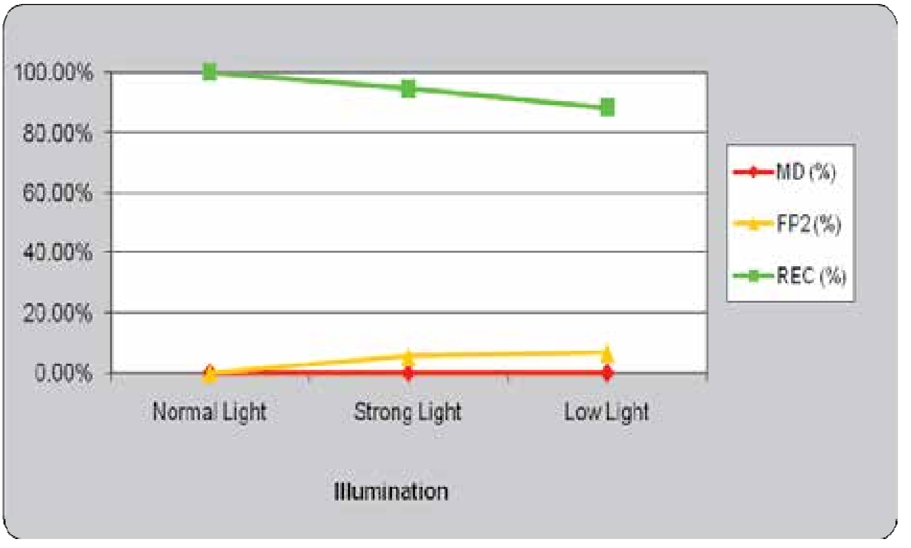


Fig. 10. Recognition performance for illumination conditions

5. Conclusion

This chapter presented the challenges of implementing a real-time video face recognition system on an embedded platform. The first section presented the main issues that need to be addressed when designing such a system and possible solutions. The second part described a working solution based on using LBP features which are fast to compute, robust to variations and able to extract useful information from the face region. In order to obtain a robust recognition system, only the features which have the same value across multiple variations of the same person were extracted. In order to increase the accuracy of the system, weights were associated to the selected features based on their discriminative power between the people from the database.
Results for this system were tested and implemented on an embedded platform, which shows good accuracy across large variations of the input data and technical parameters which satisfy the condition for real-time processing.

6. References

Ahonen, T.; Hadid, A. & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI*, 28, pp. 2037-2041, December 2006.

Belhumeur, P.N.; Hespanha, J.P. & Kriegman, D.J. (1996). Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection, *Proc. of the 4th European Conference on Computer Vision, ECCV'96*, 15-18 April 1996, Cambridge, UK, pp. 45-58

Corcoran, P. & Costache, G. (2005). Automated sorting of consumer image collections using face and peripheral region image classifiers, *Consumer Electronics, IEEE Transactions on* Vol. 51, Issue 3, Aug. 2005, pp. 747 – 754.

Corcoran, P.; Iancu, C. & Costache, G. (2006a). Improved hmm based face recognition system. *International Conference on Optimization of Electrical and Electronic Equipment*, Brasov, Romania, May 2006.

Corcoran, P.; Ionita, M. & Costache, G. (2006b) Pose-invariant face recognition using AAMs, *International Conference on Optimization of Electrical and Electronic Equipment*, Brasov, Romania, May 2006.

Costache, G.; Mulryan, R.; Steinberg, E. & Corcoran, P. (2006). In-camera person- indexing of digital images, *Consumer Electronics ICCE '06 Digest of Technical Papers. International Conference on,* 7-11 Jan. 2006.

Georghiades, A.S.; Belhumeur, P.N. & Kriegman, D.J., (2001). From few to many: illumination cone models for face recognition under variable lighting and pose, *IEEE Trans. Pattern Anal. Mach. Intelligence*, Vol. 23, No. 6, pp 643-660.

Google Picassa (n.d.), *http://picasa.google.com/*

Land, E(1986). An alternative technique for the computation of the designator in the retinex theory of color vision. *Proc. Nat. Acad. Sci.*, vol. 83, 3078-3080

Lee, K.; Ho, J.; & Kriegman, D. (2001). Nine points of light: Acquiring subspaces for face recognition under variable lighting. *In Proc. of CVPR*, pp 519-526.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *226 International Journal of Computer Vision* 60 (2), 91.

Nefian, A.V. & Hayes III, M.H. (1998). Hidden Markov Models for Face Recognition, *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'98*, Vol. 5, 12-15 May 1998, Seattle, Washington, USA, pp. 2721-2724

Ojala, T.; Pietikäinen, M. & Mäenpää, T. (2001). A generalized Local Binary Pattern operator for multiresolution gray scale and rotation invariant texture classification, *Advances in Pattern Recognition, ICAPR 2001 Proceedings*, Springer, 397 – 406

Phillips P. J.; Grother, P.; Micheals, R.; Blackburn, D.M.; Tabassi, E. & Bone, M. (2003). Face Recognition Vendor Test 2002, *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, p.44, October 17, 2003

Pizer, S.; Amburn, E.; Austin, J.; Cromartie, R.; Geselowitz, A.; Greer, T.; Romeny, B.; Zimmerman, J. & Zuiderveld. K. (1987). Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, Vol. 39 pp. 355-368.

Podilchuk, C. & Zhang, X. (1996). Face recognition using DCT-based feature vectors, *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'96)*, vol. 4, pp. 2144-2147, May 1996.

Smith, W.A.P.; & Hancock, E.R. (2005). Single image estimation of facial albedo maps. *BVAI,* pp. 517-526.

Tessera OptiML FaceTools (2010). http://tessera.com/technologies/imagingandoptics/Documents/OptiML_faceTools.pdf

Turk, M.A. & Pentland, A.P. (1991). Face Recognition using Eigenfaces, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition,* pp. 586-591, June 1991.

OptiML is a trademark of Tessera Inc. In the United States and other countries.

7

Video Based Face Recognition Using Convolutional Neural Network

Shefa A. Dawwd and Basil Sh. Mahmood
Computer Engineering Department, University of Mosul, Iraq

1. Introduction

This chapter addresses an improved approach to video face recognition (VFR). Techniques to recognize faces in video streams have been described in the literature for more than 20 years (Wang et al., 2009). Early methods were based on the still-to-still techniques which aimed at selecting good frame and did some relative processing. Recently researchers began to truly solve such problems by spatio-temporal representations. Most of the existing systems address video-based face recognition problems as follows: First, detect face and track it over time. Sometimes selecting good frames which contain frontal faces or valued cues is necessary. Next, when a frame satisfying certain criteria (size, pose, illumination and etc.) is acquired, recognition is performed, sometimes, by using still-to-still recognition technique. Figure 1 shows the whole process. In addition, some methods also utilize combination cues, such as audio, gait and so on, to make a comprehensive analysis and take decision (Yang & Waibel, 1996).

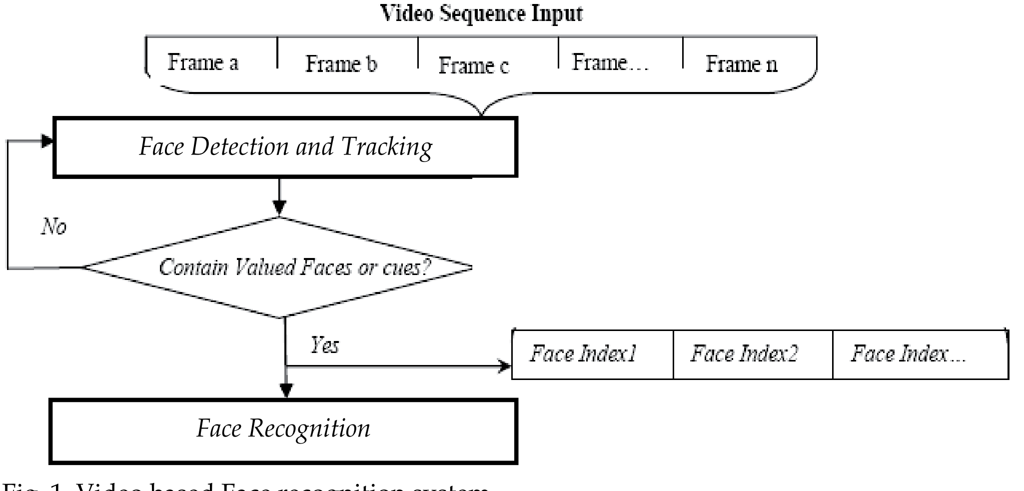


Fig. 1. Video based Face recognition system

The system proposed in this chapter employs the neural network techniques for both face detection and recognition. The face detector uses the frame color information while a grayscale frame is required for the recognition process. The two essential processes are implemented in real time using FPGA to achieve the requirement of video processing. A short description of the proposed system is described in the following sections. Section 2 includes a short introduction on face detection and tracking. After that, a concentration on the recognition process which is based on using the Convolutional Neural Network (CNN) is presented in the rest of this chapter.

2. Face detection and tracking

Real-time face detection is important in the video-based face detection. A real-time face detection methods can uses color information to detect and validate human face (Dawwd et al., 2008; Dawwd, 2009). A hybrid adaptive face detection system which combined the advantages of knowledge based and neural methods is presented in face detection process. This system is a special-purpose object detector that can segment arbitrary objects in real images with a complex distribution in the feature space in real time. This is achieved after training with one or several previously labeled image(s). The adaptive segmentation system uses local color information to estimate the membership probability in the object, respectively, background class. The system can be applied to detect and to localize the human face in colored images in real time. To increase the detection speed, the system is implemented primarily in hardware using FPGA techniques (see Fig. 2).

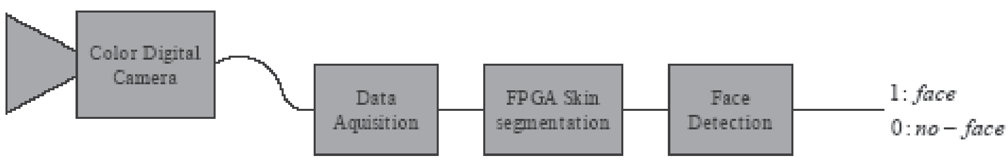
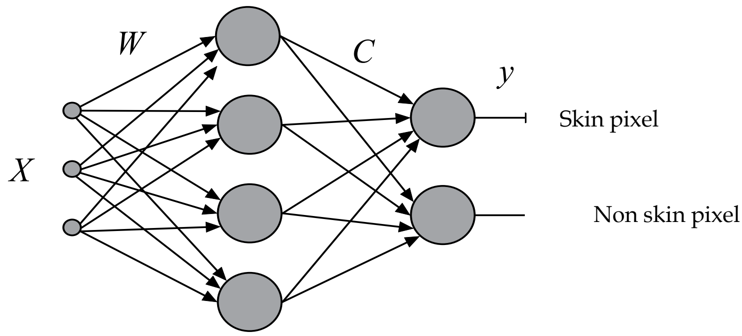


Fig. 2. Face Detector

Here, a method to detect skin color is presented. The skin detector uses a multi layered perceptron (MLP) with three inputs, one hidden layer and two output neurons (see Fig. 3). Each pixel is represented by either RGB (red, green and blue) or Yuv color components. These three color components are used as inputs by the neural network. The network output is given by:

$$y = \sum_{j=1}^{Q} c_j \varphi_j(X) + \beta \tag{1}$$

where $\varphi_j(X)$ is the output of the j-th hidden neuron, and c_j is the synaptic weight of the output neuron. To estimate the neural network parameters (i.e. synaptic weights and biases), a training set containing thousands skin and non-skin pixels was extracted from set of images. The network was trained using backpropagation algorithm. The generalization ability of the trained network is tested using a set containing several thousands of skin and nonskin pixels. The training and test sets were extracted from images containing skin colors of people from different races and under different lighting conditions. The final block of Fig.2 focus on the geometric face shapes of the segmented skin regions to distinguish them from the regions other than faces. Once the face is detected, then it can be traced afterward. Color and shape are important cues for tracking, based on which many methods are proposed, in (Yang & Waibel, 1996) a review of different robust face detectors and trackers are introduced.



where X = {R, G, B} or {Y, u, v}

Fig. 3. Neural Network structure for pixel skin detection

3. Convolutional neural network

Convolutional neural networks (CNN) with local weight sharing topology gained considerable interest both in the field of speech and image analysis. Their topology is more similar to biological networks based on receptive fields and improves tolerance to local distortions. Additionally, the model complexity and the number of the weights are efficiently reduced by weight sharing. This is an advantage when images with high-dimensional input vectors are to be presented directly to the network instead of explicit feature extraction that results in reduction which is usually applied before classification. Weight sharing can also be considered as alternative to weight elimination in order to reduce the number of the weights. Moreover, networks with local topology can more effectively be migrated to a locally connected parallel computer than fully connected feedforwared network (Neubauer, 1998).

The term CNN is used to describe an architecture for applying neural networks to two-dimensional arrays (usually images), based on spatially localized neural input. This architecture has also been described as the technique of shared weights or local receptive fields (Browne & Ghidary, 2003).The concept of sharing weights is that, a set of neurons in one layer using the same incoming weight. The use of shared weights leads to all these neurons detecting the same feature, though at different positions in the input image (receptive fields); i.e. the image is convolved with a kernel defined by the weights. The weight sharing technique has the interesting side effect of reducing the number of free parameters, thereby reducing the capacity" of the machine. Weight sharing also reduces the gap between test error and training error. This advantage is significant in the field of image processing, since without the use of appropriate constraints, the high dimensionality of the input data generally leads to ill-posed problems. Processing units with identical weight vectors and local receptive fields are arranged in a spatial array, creating architecture parallels to models of biological vision systems (Fukushima & Miyake,1982).

4. Neocognitron neural network

Fukushima (Fukushima & Miyake, 1982) were amongst the first to experiment with convolutional neural networks and obtained good results for character recognition by applying convolutional neural networks within an image pyramid architecture: processing layers alternate between convolution and sub-sampling. This architecture is called Neocognitron. This multi-scale architecture has been now widely adopted and appears to provide a robust representation in many object recognition problems. A practical architecture of the Neocognitron is shown in Fig. 4.

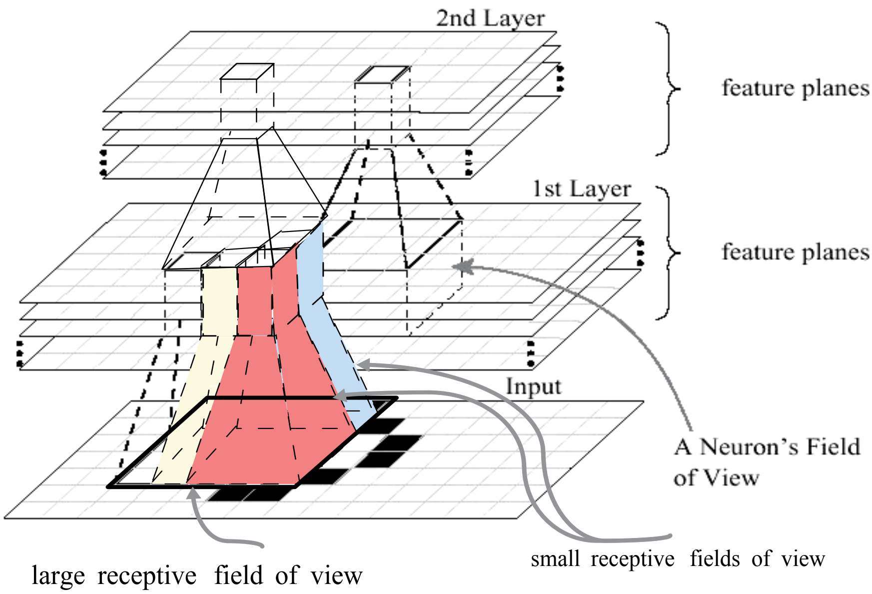


Fig. 4. Neocognitron main layers and receptive fields

Each layer extracts certain shape-features, as for example edge orientation, from a localized region of the preceding layer and projects the extracted information to the next higher layer. The complexity and abstractness of the detected features grow with the layer height, until complicated objects can be recognized. A layer consists of a number of feature planes, each of which is assigned to recognize one specific image feature.

Neurons belonging to the same plane are identical in the sense that they share the same synaptic weights. This architecture, showing a high degree of self-similarity, seems particularly dedicated to be implemented on a parallel hardware platform.

For simplicity, another illustration of the Neocognitron when the feature planes are arranged serially and the receptive fields are represented as circles is shown in Fig.5. Note that the modified Neocognitron (MNEO) is proposed and implemented in this chapter.

The Neocognitron consists of a cascade connection of a number of modular structures preceded by an input layer *Uo*. Each of the modular structures is composed of two sub-layers of cells, namely a sub-layer *Us* consisting of S-cells, and a sub-layer *Uc* consisting of C-cell (S-cells and C-cells are named after simple cells and complex cells in physiological terms, respectively). Regard to CNN cells and layers names, S-cells refer to cells in convolution layers whereas C-cells refer to cells in down-sampling layers. In the Neocognitron, only the input interconnections to S-cells are variable and modifiable and in contrast to the down-sampling layers in CNN, the input interconnections to C-cells are fixed and unmodifiable.

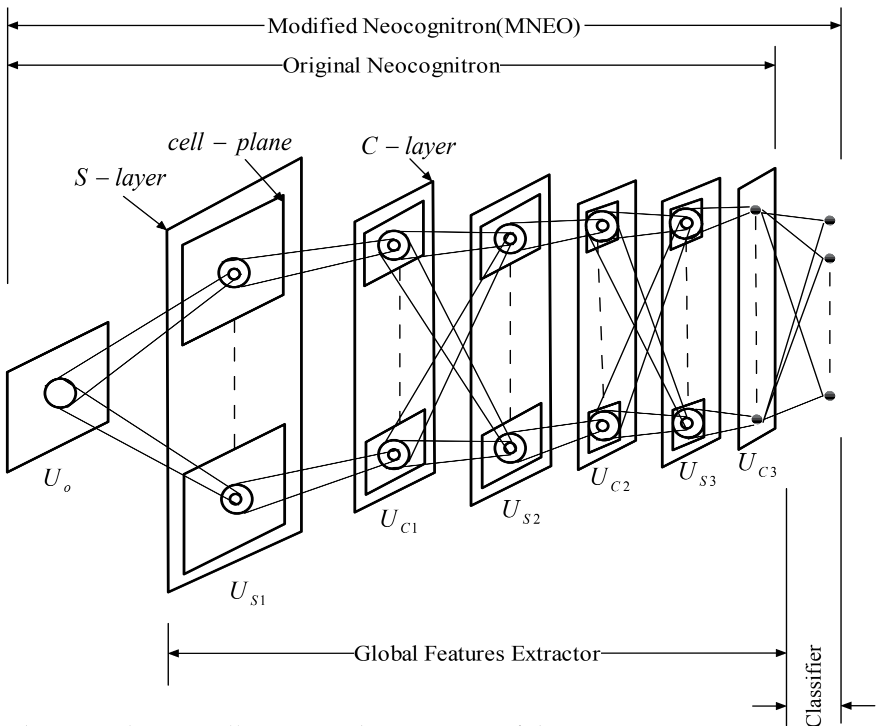


Fig. 5. Schematic diagram illustrating the structure of the Neocognitron

4.1 Cells employed in the neocognitron

All the cells employed in the Neocognitron are of analogue type: i.e., the input and output signals of the cells have non-negative analogue values. Each cell has characteristics analogous to a biological neuron. In the Neocognitron, four different types of cell are used, i.e., S-cells, C-cells, Vs-cells and Vc-cells.

An S-cell has a lot of input terminals, either excitatory or inhibitory. If the cell receives signals from excitatory input terminals, the output of the cell will increase. On the other hand, a signal from an inhibitory input terminal will suppress the output. Each input terminal has its own interconnecting coefficient whose value is positive. Although the cell has only one output terminal, it can send signals to a number of input terminals of other cells.

An S-cell has an inhibitory input which causes a shunting effect. Let *u(1), u(2),…….u(N)* be the excitatory inputs and *v* be the inhibitory input. The output *w* of this S-cell is defined by (Fukushima & Miyake, 1982):

$$w = \varphi\left[\frac{1+e}{1+h} - 1\right] = \varphi\left[\frac{e-h}{1+h}\right] \tag{2}$$

where:

$$e = \sum_{\mathrm{v}=1}^{N} a(\mathrm{v}).u(\mathrm{v})$$

$$h = b.v$$

$$\varphi[x] = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

where *a(v)* and *b* represent the excitatory and inhibitory interconnecting coefficients, respectively

The cells other than S-cells also have characteristics similar to those of S-cells. The input-to-output characteristics of a C-cell are obtained from the last equation if we replace *φ[ ]* by *ψ[ ]*, where *ψ[ ]* is a saturation function defined by:

$$\psi[x] = \begin{cases} \dfrac{x}{\alpha + x} & x > 0 \\ \\ 0 & x \leq 0 \end{cases} \tag{3}$$

The parameter α is a positive constant which determines the degree of saturation of the output. In the computer simulation and in hardware implementation, the parameter α is chosen to be equal to zero. S-cells and C-cells are excitatory cells, i.e., the output terminals of these cells are connected only to excitatory input terminals of other cells. On the other hand, Vs-cells and Vc-cells are inhibitory cells, whose output terminals are connected only to inhibitory input terminals of the other cells. A Vs-cell has only excitatory input terminals and the output of the cell is proportional to the sum of all the inputs weighted with the interconnecting coefficients. That is a Vs-cell yields an output proportional to the (weighted) arithmetic mean of its inputs. A Vc-cell also has only excitatory input terminals, but its output is proportional to the (weighted) root-mean-square of its input. Let *u(1), u(2),…….u(N)* be the inputs to a Vc-cell and *c(1),c(2),…….c(N)* be the interconnecting coefficients of its input terminals. The output *w* of this Vc-cell is defined by:

$$w = \sqrt{\sum_{v=1}^{N} c(v).u^2(v)} \tag{4}$$

4.2 Formulae governing the network

S-cells have inhibitory inputs with a shunting mechanism. The output of an S-cell of the k_l-*th* S-plane in the *l*-th module is given by (Fukushima & Miyake, 1982)

$$u_{sl}(k_l, n) = r_{ol}.\varphi\left[\frac{1 + \sum_{k_{l-1}=1}^{K_{l-1}} \sum_{v \in S_l} a_l(k_{l-1}, v, k_l).u_{cl-1}(k_{l-1}, n+v)}{1 + \dfrac{r_{ol}}{1 + r_{ol}}.b_l(k_l).v_{cl\text{-}1}(n)} - 1\right] \tag{5}$$

where:

$$v_{cl-1}(n) = \sqrt{\sum_{k_{l-1}=1}^{K_{l-1}} \sum_{v \in S_l} c_{l-1}(v).u_{cl-1}^2(k_{l-1}, n+v)}$$

φ[ ]:	a function defined by equation 2
$v_{cl-1}(n)$:	Vc-cell,layer *l-1*, position *n*
$a_l(\,),b_l(\,)$:	modifiable weights
$c_{l-1}(\,)$:	positive fixed weights
r_{ol}:	selectivity parameter
S_l:	receptive field

The selectivity parameter r_{ol} in the above equation controls the intensity of the inhibition. The larger the value of r_{ol} is, the more selective becomes the cell's response to its specific feature. r_{ol} is believed that it is a key factor to control the ability of the Neocognitron to recognize deformed patterns. If the selectivity is too high, the Neocognitron loses the ability to generalize and cannot recognize deformed patterns robustly. If the selectivity is too low, the Neocognitron loses the ability to differentiate between similar patterns of different categories. The values of fixed interconnections $c_{l-1}(\,)$ are determined so as to decrease monotonically with respect to $|v|$. The size of the connecting area S_l of these cells is set to be small in the first module and to increase with respect to depth *l*.

The interconnections from S-cell to C-cell are fixed and unmodifiable as mentioned. Each C-cell has input interconnection leading from a group of S-cells in the S-plane preceding it (i.e., in the S-plane with the same k_l–number as that of the C-cell). This means that all of the S-cells in the C-cell's connecting area extract the same stimulus features but from slightly different positions on the input layer. The values of the interconnections are determined in such a way that the C-cell will be activated whenever at least one of these S-cells is active, Hence, even if a stimulus pattern which has elicited a large response from the C-cell is shifted a little in position, the C-cell will still keep responding as before, because another neighboring S-cell in its connecting area will become active instead of the first. In other words, a C-cell responds to the same stimulus feature as the S-cell preceding it, but is less sensitive to a shift in position of the stimulus feature

Quantitatively, the output of a C-cell of the k_l-th C-plane in the *l*-th module is given by:

$$u_{cl}(k_l,n)=\psi\left[\frac{1+\sum_{\mathrm{v}\in D_l} d_l(\mathrm{v}).u_{sl}(k_l,n+\mathrm{v})}{1+\nu_{sl}(n)}-1\right] \tag{6}$$

where:

$$\nu_{sl}(n)=\frac{1}{K_l}\sum_{k_l=1}^{K_l}\sum_{\mathrm{v}\in D_l} d_l(\mathrm{v}).u_{sl}(k_l,n+\mathrm{v})$$

ψ[]:	a function defined by equation 3
$v_{sl}(n)$:	Vs-cell,layer *l*, position *n*
$d_l(v)$:	fixed interconnection which determined so as to decrease monotonically with respect to $|v|$
D_l:	receptive field, the size of D_l is set to be small in the first module and to increase with the depth of *l*

4.3 Learning rules

The Neocognitron is trained layer by layer starting from the first hidden layer. After training of the first hidden layer with images containing only simple features, the training set for the next layer, containing more complex patterns, is propagated through the first layer is reached. This procedure is repeated until the output layer is reached. Thus higher layers represent features of increasing complexity. One advantage of this approach is that the first hidden layer does not have to be retrained for each classification problem since, for typical visual recognition tasks, edge usually have to be extracted at the first level. The reinforcement learning rule proposed by Fukushima is used here both for supervised and unsupervised training of the Neocognitron (Fukushima & Miyake, 1982):

$$\begin{aligned} &\Delta a_l(k_{l-1}, v, \hat{k}_l) = q_l . c_{l-1}(v) . u_{cl-1}(k_{l-1}, \hat{n} + v) \\ &\Delta \mathrm{b}_l(\hat{k}_l) = q_l . v_{cl-1}(\hat{n}), \end{aligned} \tag{7}$$

where q_l is a positive constant which determines the speed of increment.

4.4 Recognition by the neocognitron

In order to help with the understanding of the principles by which the Neocognitron performs pattern recognition, Fig. 6 shows a rough sketch of the working of the network in the state after completion of self-organization.

The network is assumed to be self-organized on repeated presentations of a set of stimulus patterns such as "A", "B", "C" and so on. In the state when self-organization has been completed, various feature-extracting cells are formed in the network, as shown in Fig. 6.

If pattern "A" is presented to the input layer *Uo*, the cells in the network yield outputs as shown in Fig 4. For instance, the S-plane with *k1*=1 in sub-layer U_{s1} consists of a two-dimensional array of S-cells which extract ∧ -shaped features. Since the stimulus pattern "A" contains a ∧-shaped feature at the top of this S-plane, it yields a large output as shown in the enlarged illustration in the lower part of Fig 6.

A C-cell in the succeeding C-plane (i.e., the C-plane in sub-layer U_{c1} with *k1*=1) has interconnections from a group of S-cells in this S-plane. For example, the C-cell shown in Fig 6 has interconnections from the S-cells whenever at least one of these S-cells yields a large output. Hence, the C-cell responds to a ∧-shaped feature situated in a certain area in the input layer and its response is less affected by the shift in position of the stimulus pattern than that of the preceding S-cells. Since this C-plane consists of an array of such C-cells, several C-cells which are situated near the top of this C-plane respond to the - shaped feature of the stimulus pattern "A". In sub-layer U_{c1}, besides this C-plane, other C-planes extract features with shapes like , and so on.

In the next module, each S-cell receives signals from all the C-planes of sub-layers U_{c1} . For example, the S-cell of sub-layer U_{s2} shown in Fig.6 receives signals from C-cells within the thin-lined circles in sub-layer U_{c1} Its input interconnections have been reinforced in such a way that this S-cell responds only when ∧ -shaped, -shaped, and -shaped features are presented in its receptive field with configuration like

Hence, pattern "A" elicits a large response from this S-cell, which is situated a little above the center of this S-plane. Even if the positional relation of these three features is changed a little, this cell will still keep responding, because the preceding C-cells are not so sensitive to the positional error of these features. However, if the positional relation of these three features is changed beyond some allowance, this S-cell stops responding.

Same approach can be followed if face features is required to be detected. For example, in Fig. 7, the eye features (F1 to F4) are composed to form the eye pattern.

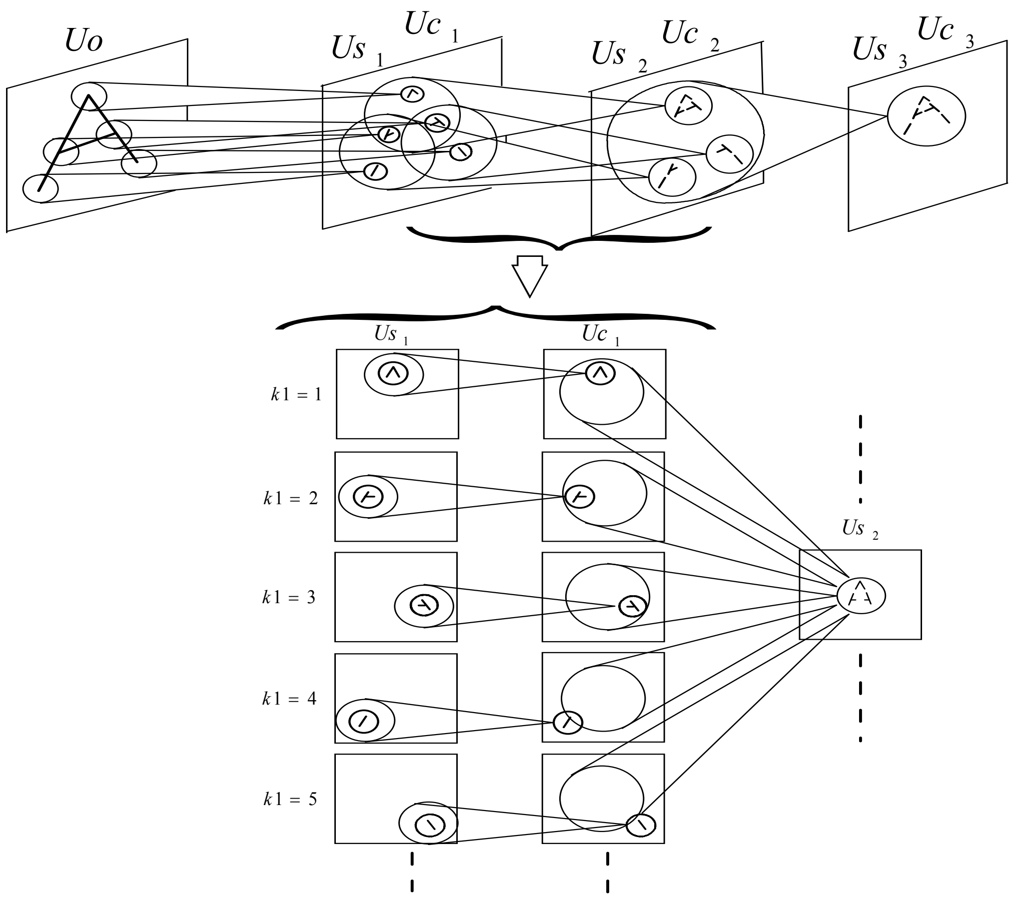


Fig. 6. (Fukushima & Miyake, 1982): An example of the interconnections between cells and the response of the cells after completion of the self-organization

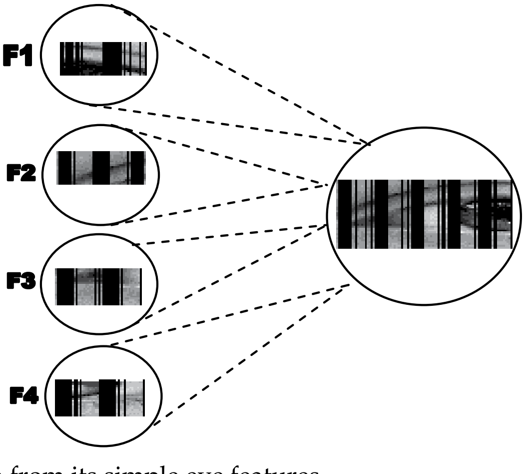


Fig. 7. Eye composition from its simple eye features

4.5 The proposed image recognition neural network system

The image recognition system described in this chapter consists of a hierarchy of several layers of artificial neurons, arranged in planes to form layers. The system consists of two parts: feature extractor and classifier. The feature extractor operates on an input image, which are then processed by the classifier (see Fig. 5). The Neocognitron is used as a feature extractor. An image is divided by the feature extractor into sub-images. The extraction of local features is based on the similarity among sub-images. The feature extractor is usually trained by unsupervised training algorithm. The training is achieved sequentially layer by layer, and the output of each layer will be considered as the input of the next layer.

The main role of the classifier is to relate the global features generated by the feature extractor (Neocognitron) to the desired recognition code. The classifier is usually feedforward and fully connected. The classifier is usually trained by supervised training algorithm. If two images belonging to the same category of a training set have different global features that result from the output of highest U_{cl} sub-layer of the Neocognitron, then, the classifier will associate these two different global features to the same recognition code. This is considered as the advantage of the classifier. It can be said that the Convolutional Image recognition system used in this chapter is based on the original Neocognitron but with some modifications and additional parts. This new structure is called MNEO to differentiate it from the original Neocognitron (see Fig. 5).

4.6 Modification of the Neocognitron(MNEO)

Since the layers of the Neocognitron in this work are independently trained, therefore, there are several possibilities for combining different kinds of neurons and learning rules. One method is proposed in his work which uses Mc Culloch-Pitts neurons in S-sublayers(Neubauer, 1998) instead of using complicated neurons based on the original Neocognitron. Also kohonen's topology preserving mapping algorithm is used for parameter adaptation (Dawwd, 2000). In order to reduce the training time, only one representation map is trained and then copy its representations to create the layer's planes. While the classifier discussed above is considered as an additional part for the original Neocognitron.

4.6.1 Simple model of neurons

In contrast to the Neocognitron, the MNEO uses S-neuron based on the Mc Culloch-Pitts model. Inhibitory cells are not used and consequently S-sublayers can be easily trained by any training algorithm. Therefore, the output of an S-cell of the k_l-*th* S-plane in the l-th module will be

$$u_{sl}(n,k_l) = \phi\left[\sum_{k_{l-1}}^{K_{l-1}} \sum_{v \in S_l} a_l(k_{l-1},v,k_l)\,u_{cl-1}(k_{l-1},n+v)\right] \tag{8}$$

where:

$$\phi(x) = 1/(1+\exp(-x))$$

4.6.2 Learning algorithm of the MNEO

As mentioned earlier, since the Neocognitron is used for feature extraction, the unsupervised self-organizing learning algorithm (SOM) (Dawwd, 2000) can be used to develop the representations in the S-sublayer(s). The SOM algorithm requires initializing the map size before starting of the training. Learning occurs only in S-sublayers. Essentially the algorithm modifies an unsupervised learning rule to cope with competition in a weight shared layer as follows:

First after an input has been presented to the network, the most active node (i.e. the winning neuron) is determined, second, the S-neuron connections are updated by using kohonen's rule. After learning has been completed, weight sharing is performed along the spatial to create the S-planes that represent the S-sublayer.

After learning has been completed and S-planes have been created, the input image is projected through S-sublayer. For each overlapped spatial window (each sub-image), the input vector is projected to each neuron in each S-plane at the same spatial coordinate, then the most active neuron among these planes is selected .The later operation determines which feature is included in that sub-image. Then the other neurons are set to zeros.

4.6.3 Complex model of neurons

For the designed network we do not care about the values and type of connections of C-cells to the input vector represented by the receptive field of the corresponding S-plane. As mentioned earlier to simplify the implementation of U_{cl} sub-layers, α is chosen to be zero. Since the inhibitory cells in the complex sublayer of the modified S-layer are not of use, therefore, the output of a C-cell of the k_l-th C-plane in the l-th module will be:

$$u_{cl}(k_l,n)=\psi\left[\sum_{v\in D_l} d_l(v).u_{Sl}(k_l,n+v)\right] \tag{9}$$

and $\psi[\ ]$ is defined as:

$$\psi[x]=\begin{cases}1 & \mathrm{x}>0\\ 0 & \mathrm{x}\le 0\end{cases} \tag{10}$$

5. Mapping neural networks on parallel computers

How can we map a specific neural network on a parallel computer to achieve maximum performance? (Schoenauer et al., 1998). The key concepts of an efficient mapping are load balancing over an array of processing elements, minimizing inter processing element communication and minimizing synchronization between the processing elements (PEs). Hence that each PE performs the computations required for a single neuron of the network. Furthermore, the mapping should be scalable both for different network sizes, and for different number of processing elements. In Fig. 8, the weight matrix presentation of a simple neural network (four neurons with four synapses each) is shown in the middle, while the left side shows the conventional presentation of the same network. The rectangle N in the mid part of Fig.8 denotes the activation function of the neuron. The circle wij represents the computation of the synapse: yi = wij*xj +yi-1 where yi-1 is the result from the proceeding synapse.

Direct implementation for non-linear sigmoid activation functions is very expensive. There are two practical approaches to approximate sigmoid functions with simple FPGA designs (Zhu & Sutton, 2003). *Piecewise linear approximation* describes a combination of lines in the

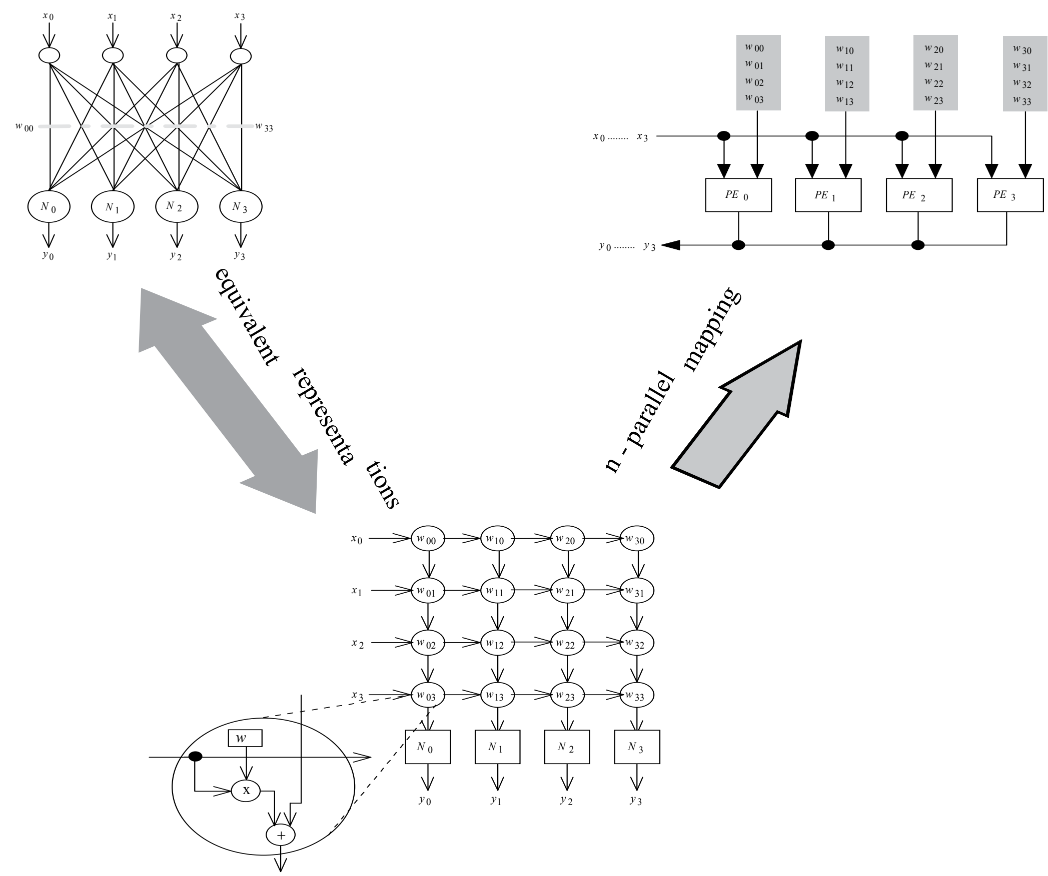


Fig. 8. Presentation of a neural network: conventional (left), weight matrix (middle), mapped N-parallel (right)

form of $y=ax + b$ which is used to approximate the sigmoid function. Note that if the coefficients for the lines are chosen to be powers of two, the sigmoid functions can be realized by a series of shift and add operations. Many implementations of neuron activation functions use such piecewise linear approximations one of them is. The second method is *lookup tables*, in which uniform samples taken from the center of sigmoid function can be stored in a table for look up. The regions outside the center of the sigmoid function are still approximated in a piece-wise linear fashion.

6. MNEO design and implementation

As mentioned previosly, the MNEO convolutional neural network model is to be implemented. This model is formed by an even number of layers, the odd ones which are simple layers have adaptable input connections and the even ones which are complex layers have fixed input connections. Each pair of layers has an equal number of planes. The number of planes is increased in a consequent manner from input to output layer in order to detect more specific features of higher complexity while the spatial resolution is decreasing. According to these properties, a special strategy is to be followed to implement the architecture of the MNEO in hardware. In this chapter, a parallel digital hardware implementation of the MNEO in FPGA platform is presented in details.

6.1 Network parameters

The size and parameters of the MNEO neural network is dependent on the application. Therefore, a specified application should be determined beforehand. As they contain a great deal of information and many complex features which can vary significantly depending on the acquisition conditions, faces are considered one of the more challenging problems in image analysis and object recognition.

6.1.1 Still-to-Still image database

A resolution of 32x32 pixels can be considered for the task of face recognition since a face is primarily characterized by existence of eyes, nose and mouth together with their geometrical relationship all of which can be recognized at low spatial resolution (Neubauer, 1998). The Oracle Research Labs database (see Fig. 9, available in http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html), which has 10 different images of 40 distinct subjects is used in this work. The images are grayscaled with a resolution of 92x112, but the resolution is reduced to 32x32. The influence of the resolution reduction in hardware field is to reduce the probability of FPGA over-fitting, and to lower the information content that has to be learned by the networks and consequently reduce the required hardware resources. The designed network can classify 12 out of the 40 subjects. The experiments were performed on five training images and five testing images per person. A total of 60 training images and 60 testing images are used to adjust the parameters which presented in the next section. The training is wholly implemented in software.

6.1.2 Parameters setting of the MNEO

Parameters setting of the MNEO such as the choice of the number of layers, neurons, cell planes, and so on, is a complex process. In fact, this process requires a lot of 'fine-tuning' effort and can be obtained by multiple simulation run of networks with different parameters. Then the most precise network is selected and its parameters are adapted. This selection is based on the evaluation of the network with respect to the recognition rate. The parameters selection strategy used in MNEO can simulate the selection of good selectivity parameter in the original Neocognitron. The network which to be implemented in this chapter is depicted in Fig. 10. The network structure consists of five layers, first hidden layer is a simple layer of four convolutional planes. The second hidden layer is a complex layer of also four convolutional planes. The third and fourth hidden layers are simple and complex layers respectively, each is of 16 convolutional planes. There are 12 output neurons in the last layer (fully connected feedforward layer), according to 12 different subjects (faces). Receptive fields sizes are chosen as 5x5, with 4 overlapped pixels(each field is overlapped over another by four pixels in both horizontal and vertical directions). The features in the hidden layers are organized as (4x4)x4, with 2 overlapped pixels,(4x4)x4, with 3 overlapped pixels,(4x4)x16, with 2 overlapped pixels, and (4x4)x16.

6.1.2.1 Hardware implementation of the S-cell

In return to equation(8), the response of S-cell is simply a function of input vector $\hat{x}$ (receptive field) and weight vector $\hat{w}$ which can be written as (Cios & Shin, 1995):

Fig. 9. Three subjects of the ORL face database (There are 10 images for each subject).

$$u = \phi\left(\sum_{i=0}^{N-1} x_i . w_i\right) = \phi(\hat{x} \bullet \hat{w}) = \phi\left(\frac{|\hat{x}|^2 + |\hat{w}|^2 - d^2}{2}\right) \tag{11}$$

where

$d^2 = |\hat{x} - \hat{w}|^2 = |\hat{x}|^2 + |\hat{w}|^2 - 2\hat{x} \bullet \hat{w}.$

It can be seen from the above equation that the neuron response depends on the distance between $\hat{x}$ and $\hat{w}$ .Thus the smaller the distance between them, the greater the response of the neuron. Now considering that of one specific feature each receptive field has to be

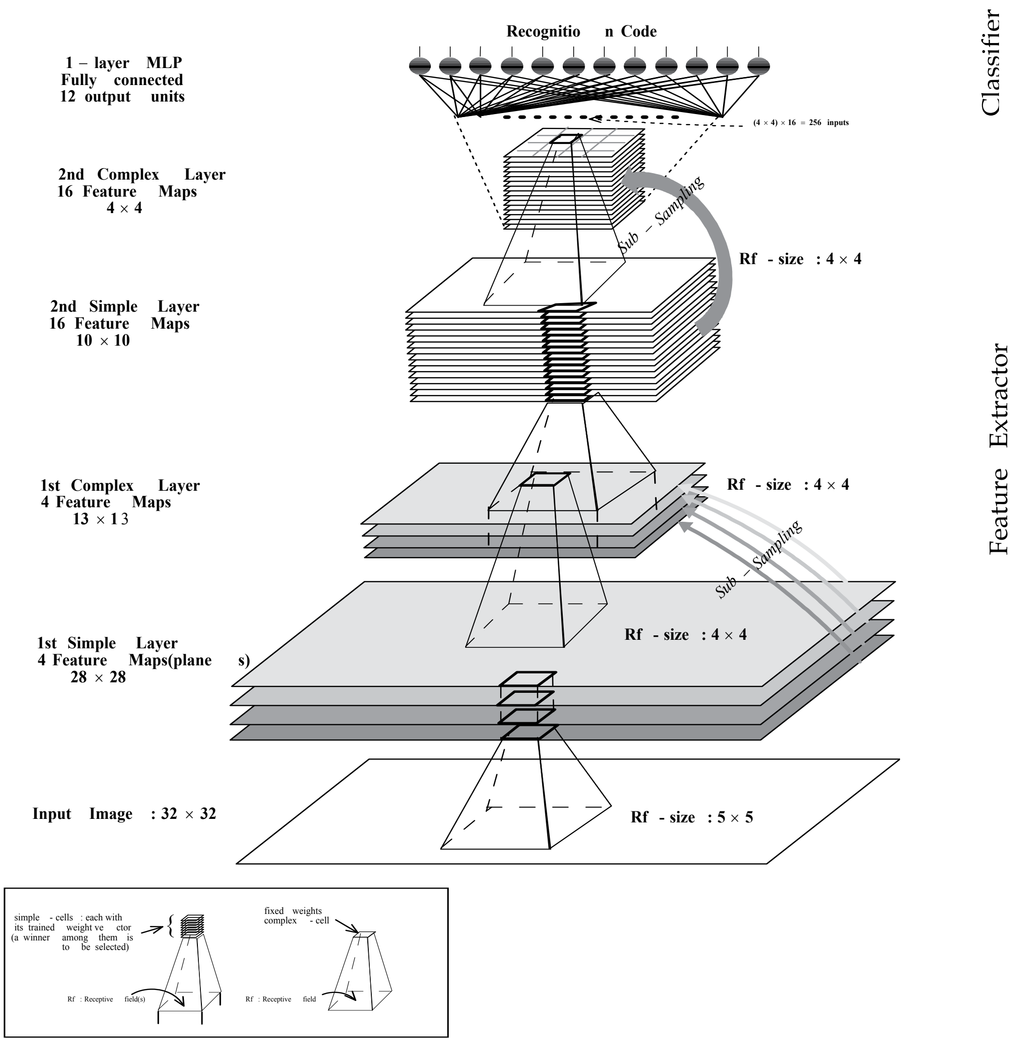


Fig. 10. The modified Neocognitron (MNEO) structure with its layer's parameters used for face recognition

detected by a number of S-cells distributed over different plans. This means, the sigmoid activation functions defined in equation (8) for those S-cells (spatially localized cells) can be replaced by a competitive function. This function sets the cell that has a minimum distance for that receptive field and resets the other cells.

The above calculation for the S-cell responses is implemented primarily in software during the MNEO training phase. For the MNEO propagation phase, the same approach is used but achieved in hardware. This ensures that the cell itself that is stimulated during the learning phase will also be stimulated during the propagation phase when the same input is applied.

In this approach, since the value of the output cell is either '0' or '1', then only one storage element is required which simplifies the successive operations and their hardware. This is because there is no need to deal with real numbers that are usually produced from the

sigmoid function. This simplification also plays a positive role when implementing the down-sampling operation done by complex layers.
The selection of the similarity measure is another factor that influences on the hardware implementation of the S-cell. Manhattan distance is used as a measure of similarity between $\hat{x}$ and $\hat{w}$ (Dawwd Sh., Mahmood B., 2009). It measures the features that are detected by the S-cell either in training or in propagation phase. In particular, the Manhattan distance is used to avoid multiplications that are required in the calculation of Euclidean distance (the most critical operation in hardware). Also dot product between $\hat{x}$ and $\hat{w}$ is avoided when using Manhattan distance. Manhattan distance is defined as:

$$d = |\hat{x} - \hat{w}| = \sum_{i=o}^{N-1} |x_i - w_i| \tag{12}$$

It can be seen from the above equation that the implementation of Manhattan distance in hardware requires computational units for integer subtraction, accumulation and the determination of absolute unsigned values.

6.1.2.2 Hardware implementation of the C-cell

As it is done in simplifying the hardware implementation of the S-cell by representing its output by only one bit, the same is done with the C-cell. From equation(3) it can be seen that if the parameter a is chosen equal to 0, the output of the C-cell will be either one or zero as shown in (9). Here, also reduction of storage elements and simplification of the hardware connected to the output of the C-cell are achieved. Depending on the representation of the S-cell and the C-cell activation functions, the C-cell can be built by only using OR function. The benefits from these representations not only influence on the implementation of simple and complex layers, but also they play an important and essential role for implementing the final layer of the MNEO(the fully connected feedforwared layer).

6.1.2.3 Hardware implementation of the Feedforward-cell

In feedforward layer, the dot product calculations among the weight vectors and the receptive field vectors of the last complex layer in the MNEO require multiplications. But the mentioned modification of the MNEO network does not need this multiplication operation. Feedforward layer only needs accumulation operation and the number of required accumulators equals to the number of the feedforward cells. Each accumulator accumulates the scalar weight of a cell if its input is '1'. The equation for feedforward cell is:

$$P = \theta\left(\sum_{i=0}^{N-1} x_i . w_i\right) \tag{13}$$

θ is the sigmoid transfer function, P is the cell output , x_i and w_i are the input and weight vectors respectively. For example assume the feedforward cell receives 3 weighted inputs as $\hat{x}$ =(1,0,1) and its trained weights are $\hat{w}$ =(0.98,0.13,0.22), then the accumulator produces (0.98+0.22) which equal to 1.2. Along with the accumulator, conditioning circuitry (mainly AND gates) is used to select which value of $\hat{w}$ vector is to be accumulated. To produce cell's output, activation function (θ) (sigmoid usually used in feedforward) should be implemented and as will be shown in section 6.2, its implementation will require only one multiplication operation.

6.2 Implementation of the processing flow

Some transfer functions like sigmoid function need some modifications to simplify the hardware of the function. In this case, the sigmoid function has been substituted by a piecewise linear function like *satlin* function (Chapman, 1997). The substitution is based on the selection of a linear *satlin* equation that has a minimum least square error with the original sigmoid function. Only one multiplier and one adder are required to implement the approximated linear transfer function. Using one multiplier and one adder for each feedforward node may be also perceived as a critical problem if the number of output nodes exceeds the number of embedded multipliers in the FPGA chip. To solve this problem, only one *satlin* computational unit is built and made common to all feedforward nodes, such that the output nodes use this unit sequentially in a pipelined manner.

As it is not worthwhile to use pipelining for the successive layers of the convolutional network, then the processing units required and used for one layer can be used for the other layers. Thus, hardware is minimized and fully time utilized. Layers hardware implementation cycle is shown in Fig. 11. All receptive fields of view are processed in parallel. The reconfigurable processing elements are those responsible to generate the node's outputs for each layer. The PEs are built with adders for Manhatan distance calculations, buffers, comparators accumulators and activation function emulators which contain multipliers. All these components are fully pipelined. Each layer begins its calculation according to a control mechanism (Dawwd Sh., Mahmood B., 2009).

7. Video face recognition

Real time frame processing of video image should be implemented in the period of 1/30 second (30 frame/sec) or less than this period. To achieve this goal, detection and recognition should be achieved in fastest possible time. Therefore the original Neocognitron is modified and the MNEO is presented in this work to deal with this challenge.

Face recognition is challenging visual classification task. Reasonable deviations from a three-dimensional face shape have to be detected, also it is necessary to normalize the face with respect to the size and position and orientation. Using the Neocognitron simplifies this task very much, because the Neocognitron can recognize stimulus pattern correctly without being affected by shifts in position or by affine scaling and rotation. In this work, the MNEO is trained by using images that extracted from different positions, scales, rotations and orientations. If a trained image is applied to the system in recognition cycle, then the system should recognize it. If an image for the same class is applied but in variant pose, the system can also recognize it according to the mentioned properties of Neocognitron(or MNEO). If more training samples are used, then more generalization is achieved.

The recognition is performed in the region of interest (the segmented area). The MNEO consider the problem of face recognition under pose variations. Once the segmentation process of the colored frame is complete in the detection stage, the recognition process begins. The face detection block in Fig. 2 can be removed in our proposed system. This is achieved by increasing the training samples with variant poses for each face class as mentioned in the last paragraph. If different consequent face index frames (see Fig. 1) are recognized according to predefined statistic criteria, then the recognized face can be considered as a valid class. If the threshold value is less than the acceptable, then the

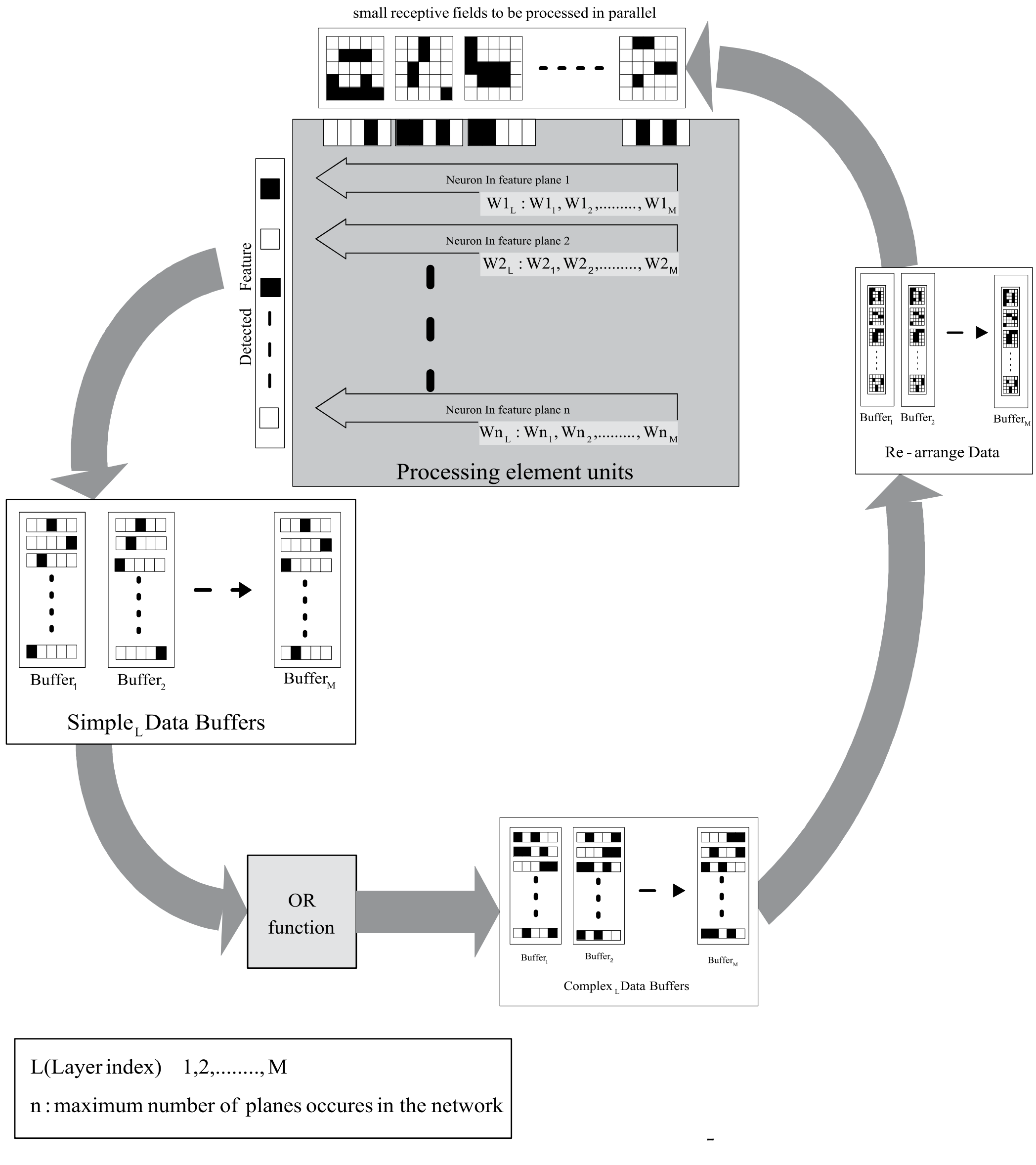


Fig. 11. MNEO layers hardware implementation cycle. All *n* feature neurons are executed in parallel, each tuned to a specific feature. Between adjacent layers, many memories act as data buffers. A special stage is necessary to re-arrange the layer output into field of views serving as input for the next layer. After implementation of all simple and complex layers, feedforward layer also uses the same processing element units to generate the final recognition code.

recognized face is not a valid class. The latter may be happened when the region of interest (the segmented region) belongs to parts other than face regions (may be hand or other color close to the color of the human skin).

8. Result

In this chapter, Xilinx Spartan-3E FPGAs are used for implementations because these devices come with large internal SRAM blocks. Each block can be used for internal weight storage and for buffering the data vectors of the segmented image.

8.1 CNN performance evaluation

In the CNN_SIMD_FPGA system, since the system does not support learning, the Connection Per Second (CPS) and Connection-bytes-per-second (CBS) are considered to evaluate the system speed performance. The most common performance rating is Connection Per Second (CPS) (Lindsey & Lindblad, 1994) which is defined as the number of multiply and accumulate operations per second during the recall or execution phase. The speed performance of the FPGA based system among other FPGA systems depends on the operation frequency of the FPGA model used. The operation frequency of the FPGA model is 50 MHz, the number of input vectors that processed in parallel are five, in each clock cycle, 5 input connections of (9bits≈1byte) are evaluated by 4 weight connections of the same bit precision, then the maximum CPS and CBS(= *bytes(weight). bytes(input). CPS)* achieved from the designed system are:

$$CPS=5\times4\times50\times10^{6}=1GCPS$$

$$CBS=1\times1\times CPS \quad =1GCBS$$

The above performance seems reasonable and comparable with the available neural network hardware (Dias et al., 2003).

8.2 CNN system and face recognition

The goal of the CNN system is to identify particular person in real-time or to allow access to a group of people and deny access to all. Multiple images per person are often available for training and real-time recognition is required.

The CNN hardware system can recognize face's image with the same recognition accuracy that achieved when using the software version. This is due to the use of efficient model, its parameters setting, functions approximations and the hardware implementation such as convolution node that is based on the realization of a competition unit. The system is trained to recognize 12 different classes. The recognition rate achieved from both software and hardware versions were equal to 93% when 60(12x5) training image and 60(12x5) testing images were used. Further recognition rate improvements can be obtained by performing more fine tuning to the CNN parameters during learning which is implemented in software. Some techniques for fine tuning improvements can be found in (Chapman ,1997).

8.3 Speedup achieved in H/W CNN

From Fig. 12, one can see that the overall time required for processing one complete image on Xilinx Spartan-3 200,000 / Spartan 3E 500,000-gates Platform FPGAs of 50MHz is equal

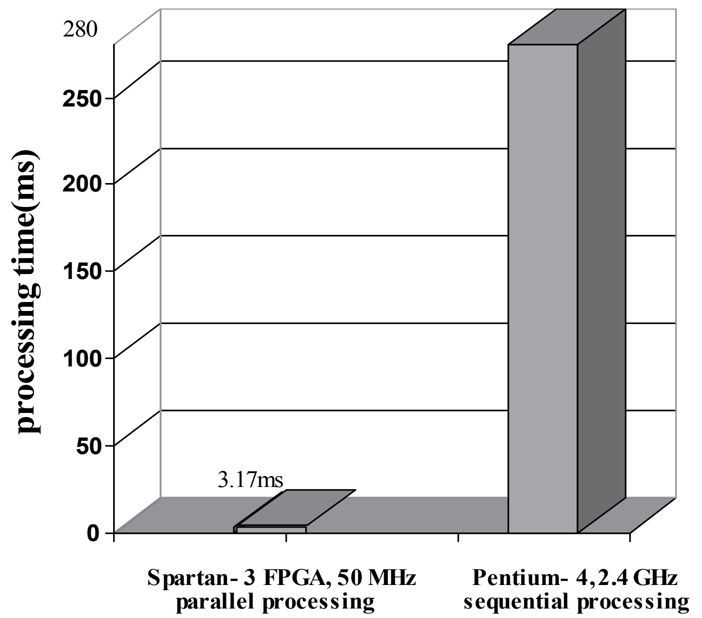


Fig. 12. An image processing time of parallel and sequential processors

to (3.17)ms, while the same model needs (280) ms when implemented primarily in software, resulting a speedup of (88).

8.4 Area consumption in H/W CNN

When the word length of the calculation unit (the input and weight value) is set to 9 bits and the word length of the accumulator is set to 13 bits, the CNN hardware required 1488 slices. This number utilizes 77% of the total number of slices in the Spartan-3 200,000 FPGA and 30% of the total number of slices in the Spartan-3E 500,000 FPGA. This means that the CNN system can be synthesized in a cheap FPGA chip. If larger system of large input image is required, the CNN system can be also synthesized in a chip of a reasonable cost.

9. Conclusion

In this work we have succeeded in mapping one of the most complex neural networks (the Neocognitron) on an FPGA SIMD architecture. The modifications of the Neocognitron to reduce its complexity give it the possibility of realizing and processing in real-time, then a high speed frame processing is achieved. Using the binary representation of cells outputs in all the network layers highly reduced the hardware resources required to implement the network. Consequently a relatively small FPGA model of 200,000 gates can implement the complex design of the MNEO system.

In the MNEO architecture, the 'sharing' of weights over processing units reduces the number of free variables, increasing the generalization performance of the network. Sharing the weights over the spatial array, leading to intrinsic insensitivity to translations, rotation and scaling of the input which is considered as attractive feature for video processing where more number of valid faces are detected, afterward, speeding up the video recognition process.

10. References

Browne M., GhidaryS. (2003). Convolutional Neural Networks for Image Processing: An Application in Robot Vision, Lecture Notes in Computer Science, Publisher: Springer Berlin / Heidelberg, Volume 2903, pages 641 – 652.

Chapman R. (1997). Using A Neuroprocessor Model for Describing Artificial Neural Nets, Project Report for EE563, Neuroprocessor Modelin.

Cios K. and Shin I. (1995). Image recognition neural network, Neurocomputing, vol. 7, pages 159-185.

Dawwd Sh. (2000). Face Recognition using Neocognitron Neural Network, M.Sc thesis, Univ of Mosul, Mosul, Iraq.

Dawwd Sh., Mahmood B., Majeed R.(2008). Real time Image segmentation for face detection based on Fuzzy Logic, Nahrain University College of Engineering Journal, IRAQ, vol. 11,No. 2, pages 278-287.

Dawwd Sh. (2009). High Performance Colored Image Segmentation System Based Neural Network, Al-Rafidain Engineering Journal, IRAQ, vol.17, No. 2, 2009, pages 1-10.

Dawwd Sh., Mahmood B.(2009). A reconfigurable interconnectedfilter for face recognition based on convolution neural network, 4th IEEE international Workshop for Design and Test, Riyadh, Saudi Arabia, ISBN: 978-1-4244-5748-9.

Dias F. M., Antunes A. andMota A.(2003). Commercial Hardware for Artificial Neural Networks: a Survey, SICICA - 5th IFAC International Symposium on Inteligent Components and Instruments for Control Applications, Aveiro.

Fukushima K. and Miyake S. (1982). Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position , pattern recognition, vol. 15, no. 6, pages 455-469.

Lindsey, C. and Lindblad T. (1994).Review of Hardware Neural Networks: A User's Perspective, Proceeding of 3rd Workshop on Neural Networks: From Biology to High Energ Physics, Isola d'Elba, Italy, Sept. 26-30.

NeubauerC. (1998). Evaluation of Convolutional Neural Networks for Visual Recognition, IEEE Transactions on Neural Networks. vol. 9, no. 4, pages 685-696.

Schoenauer T., Jahnke A., Roth U., Klar H. (1998). Digital Neurohardware: Principles and Perspectives, Neuronal Networks in Applications - NN'98 – Magdeburg.

Wang H., Wang Y., Cao Y. (2009). Video-based Face Recognition: A Survey, World Academy of Science, Engineering and Technology, December 2009, Issue 60. 1307-6892.

Yang J. and Waibel A. (1996). A real-time face tracker, n Proceedings of the Third IEEE Workshop on Applications of Computer Vision, pages 142-147, Sarasota, FL.

Zhu J. and Sutton P. (2003). FPGA Implementations of Neural Networks – a Survey of a Decade of Progress, International Conference on Field Programmable Logic and Applications (FPL'03),LNCS 2778, Springer Verlag, Berlin, Heidelberg, pages 1062-1066.

Adaptive Fitness Approach - an Application for Video-Based Face Recognition

Alaa Eleyan[1], Hüseyin Özkaramanli[2] and Hasan Demirel[2]
[1]Mevlana University
[2]Eastern Mediterranean University
[1]Turkey
[2]Northern Cyprus

1. Introduction

In the last two decades face recognition has emerged as an important research area with many potential applications that surely ease and help safeguard our everyday lives in many aspects (Zhao et al., 2003; Kirby & Sirovich, 1990; Turk & Pentland, 1991; Martinez & Kak, 2001; Belhumeur et al., 1997; Philipps et al., 2000; Eleyan & Demirel, 2007, 2011; Brunelli & Poggi, 1993; Wiskott et al., 1997). The face recognition problem from still images has been extensively studied (Sinha et al., 2006; Eleyan et al., 2008). Face recognition from video has recently attracted the attention of many researchers (Zhou et al., 2003; Li & Chellappa, 2002; Wechsler et al., 1997; Steffens et al., 1998; Eleyan et al., 2009). Video is inherently richer in information content when compared with still images. It has important properties that are absent in still images. Some of these important properties are the temporal continuity, dynamics and the possibility of constructing 3D models from faces. On the other hand, it should also be noted that video acquired facial data are normally of very low quality and low resolution, which make recognition algorithms very inefficient. The temporal continuity and dynamics of a person captured by a video makes it easier for humans to recognize people. Humans are usually able to recognize faces in very low resolution images. This is not the case for the computer based techniques which have been shown to be quite capable in recognizing faces from still images. Utilization of these properties for more efficient and high performance face recognition algorithms requires approaches that are different than the traditional approaches.

There are many reasons why humans are so successful in recognition of faces in video while computers are not. Some of these are: 1) Humans use a collection (flow) of data over time rather than an individual video image during both training and testing. 2) Humans are superbly capable of tracking objects. And in so doing can make excellent use of flow of data

In the training stage, when a new person is to be "memorized" many features such as appearance, gestures, gait etc. are encoded. Each person in the human memory (gallery) is encoded differently and there are quite a number of people memorized by humans. In the testing (recognition) stage human beings compare these features and make a decision on the identity of a person. This process however is not a "one shot" comparison, but it is continually made based on the flow of data. When the person is far a way for example, it is difficult to discern the facial features. However from the gait and gestures the human brain is able to extract important information to identify an approaching person. Based on this information the human brain automatically deems some of the people in the memory as unlikely candidates to match the approaching person and thus those candidates are not considered in further comparisons/associations. As the person approaches closer, the human brain restricts the comparison to reduced set of likely candidates in the memory.

Inspired by this biological process of making comparisons and making decisions based on a reduced set of candidates at testing stage, we propose in this chapter to design an analogues structure for computer based face recognition from video whereby the gallery is continually updated as the frames of the probe video is processed. In order to demonstrate the effectiveness of the proposed approach we employ features derived from PCA or LBP. After every probe frame, the feature vector is compared with the feature vectors of the gallery images, the unlikely images in the gallery are discarded based on the accumulated fitness of the gallery images. An update set of features are derived using remaining image in the gallery. The update set of features are used to test the next frame in the probe video. The results obtained using the idea of updated galley set indicates that significant improvement in recognition performance can be achieved. The adaptive fitness approach (AFA) is also tested without updating the gallery set. Again, this scheme with fixed gallery set gives comparable performance results as the scheme with updated gallery set.

The rest of the chapter is organized as follows. Section 2 briefly reviews feature extraction. Section 3 presents the face video database. Section 4 introduces the adaptive fitness update approach. Section 5 reports our experimental results and discussions, and Section 6 concludes this chapter.

2. Feature extraction

Feature extraction is a very crucial stage of data preparation for later on future processing such as detection, estimation and recognition. It is one of the main reasons for determining the robustness and the performance of the system that will utilize those features. It's important to choose the feature extractors carefully depending on the desired application. As the pattern often contains redundant information, mapping it to a feature vector can get rid of this redundancy and preserve most of the intrinsic information content of the pattern. The extracted features have great role in distinguishing input patterns.

In this work, instead of using more biologically oriented features, for the reasons of simplicity we employ features derived from principal component analysis (PCA) (Kirby & Sirovich, 1990; Turk & Pentland, 1991) and local binary patterns (LBP) (Ahonen et al., 2004; Ojala et al., 2002). However the recognition framework does allow the incorporation of other features. In PCA case, one needs to prepare a projection space using the training set and use it to preparing the feature vectors of both training and tested sets. In LBP every image is processed independently to form its feature vectors. So if the size of the training set changed as it does in AFA, new space has to be prepared if PCA is used to form the feature vectors while feature vectors will stay same if LBP is used.

3. Video face database

In this study we used the BANCA database (Popovici et al., 2003), which is a multimodal database designed with various acquisition devices (2 cameras and 2 microphones), and under several scenarios (controlled, degraded and adverse). The videos were collected for 52 individual (26 male and 26 female) on 12 different occasions (4 recordings for each scenario). In our work we will be using the video sequences for the 52 individual with the three different scenarios. In the degraded scenario a web cam was used, while higher quality camera was used in the controlled and adverse scenarios. Figure 1 shows samples from the database for the three scenarios.

Fig. 1. Samples of the BANCA database images *Left*: Controlled, *Middle*: Degraded, *Right*: Adverse scenarios.

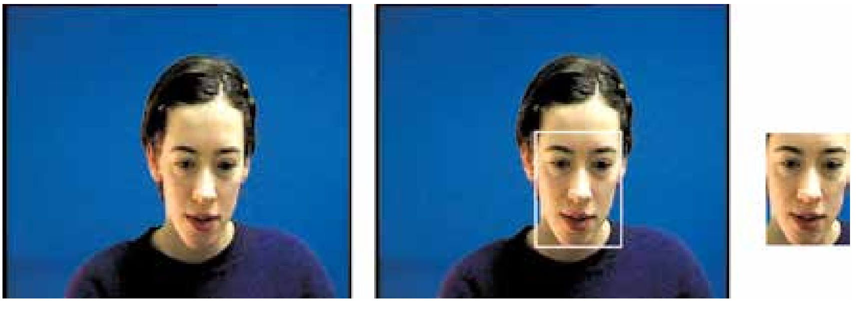


Fig. 2. Example of using face detection algorithm to crop the face region from the whole frame.

As it was computationally expensive to use all the frames in each individual's video sequence, we selected 60 frames which correspond to every other frame in the video sequence. The face images from the first n frames (n={1,2,…,10}) of each video sequence were used to form the gallery set to train the system, while the rest were used for testing.
It was essential to run face detection in the pre-processing stage on the extracted frames in order to prepare them for the face recognition process. For this reason, the local Successive Mean Quantization Transform (SMQT) (Nilsson et al., 2007) has been adopted for face detection and cropping due to its robustness to illumination changes. Cropped faces were converted to grayscale and histogram equalized to minimize the illumination problems.

Bicubic interpolation was used to resize the resulting face images to the same size of the reference resolution (size of gallery images 128 ×128). Figure 2 shows an example of the face detection cropping and resizing preprocess for one of the image in BANCA database.

4. Adaptive fitness based updating

4.1 Adaptive Fitness Approach (AFA)

The features of each subject in the gallery are derived from the first n frames (n={1,2,…,10}) of each subject's video sequences using PCA and LBP. Each frame in the test/probe video is treated as a single still image. Feature vectors of each test frame are formed using PCA or LBP techniques. Each feature vector encodes the similarity of the test frame to each of the gallery images. It is natural to expect that some of the gallery images will have high similarity with the frame under the test while others will have low similarity. One can thus establish with some confidence that those gallery images with very low similarity measures will very likely not be the match for the probe frame. Thus when processing the next frame in the test video, one can reduce the size of the gallery by discarding those unlikely candidates. This enables one to make the gallery set smaller after each tested frame. It is a well known fact that the discriminating power of algorithms such as PCA improves when the gallery set that is reduced. The algorithm of discarding images from the gallery set enables in a way the mechanism employed by human brain in recognizing an approaching person. When the person is far away (low resolution) the human brain uses global features to identify for example the approaching person and it does so by eliminating people in its gallery who are unlikely to form a match. When the person gets closer there is an automatic update of the gallery and the approaching person is compared against smaller number of people in gallery. Eventually when the person is very close, the gallery images are reduced to just a few.

Inspired by this biological process which is employed by human brain in recognition tasks, we propose a simple approach to adaptively shrink the size of the gallery set after each frame of the test video is processed. A fitness measure $\Phi_{i,k}$ is defined using the Euclidean distance as

$$\Phi_{i,k} = \begin{cases} \dfrac{\bar{\delta}_{i,k} - \delta_{i,k}}{\delta_{i,k}} & , \quad k = 1 \\ \Phi_{i,k-1} + \dfrac{\bar{\delta}_{i,k} - \delta_{i,k}}{\delta_{i,k}} & , \quad 2 \le k \le , N \end{cases} \tag{1}$$

where $\delta_{i,k}$ is a vector denoting the Euclidean distance between the i^{th} gallery image and the k^{th} test frame. $\bar{\delta}_{i,k}$ is the mean distance value of the vector $\delta_{i,k}$, and $\Phi_{i,k}$ denotes the accumulated fitness between the i^{th} gallery image and the k^{th} test frame. At the first frame of the test video the fitness is set to be just the normalized distance as the first line in equation (1) indicates. The normalization is achieved by subtracting $\delta_{i,k}$ from the mean distance $\bar{\delta}_{i,k}$ and dividing by the corresponding element $\delta_{i,k}$. in order to reduce the effect of outliers.

The accumulated fitness measure forms the basis for shrinking the size of the gallery by discarding the candidates in the gallery that are unlikely to form a match with the probe video frame. After eliminating unlikely candidates from the gallery, a new set of features is formed from the remaining more fit candidates. For example, if PCA is used for feature extraction, after eliminating images from the gallery the existing eigenspace is updated and new feature vectors is formed for the remaining images. On the other hand if LBP is used, throwing out an image accounts to throwing out the corresponding feature vector; thus there is no need to recalculate a new set of feature vectors. Eventually the continuous updating of the gallery promises to leave behind few candidates that are very likely to form a match with the person under test.

This approach has several advantages. Its resemblance to the recognition by human beings is the first to note. Second, it promises to speed up the recognition process due to the discarding of the unfit images from the gallery. However it should be pointed out that due to the discarding of images from the galley this approach may lead to, even though very unlikely, throwing out some of the correct images in the gallery.

The number of discarded images from the gallery set at each processed frame depends on the standard deviation of the accumulated fitness values at that particular frame. The standard deviation of this distribution is used to establish a fitness threshold θ_c for discarding gallery images. The critical fitness value θ_c is picked conservatively to ensure with almost 100% confidence that the correct gallery images are not eliminated. This forces one to process almost all the frames in order to come up with a decision since with a low θ_c one discards few images from the gallery. This also leads to higher computational burden. This undesirable situation can be avoided by picking a higher threshold θ_c.

The adaptive fitness approach can also be used without updating the gallery. In this scheme one simply process all the frames in the probe video and accumulates the fitness measure with the originally prepared feature vectors. This approach where the gallery is fixed and no updating is required is computationally more efficient compared with the scheme where one updates the gallery and the feature vectors. However, this advantage is not significant since the updating of feature vectors after the gallery is reduced in size can be done incrementally without much computational burden. Furthermore, in the scheme with gallery updating one does not need to process all the video frames to come up with a decision. Figures 3 and 4 give step by step the algorithms of these two schemes.

Initialize *gallery set*
For *frame= 1, 2, … N*
    Compute *feature vectors*
    Project *probe image*
    Compute *fitness measure*
    Accumulate *fitness measure*
    Discard *gallery images with lowest fitness values*
    Update *gallery set*
End
Identify *using highest accumulated fitness value*

Fig. 3. Pseudo code for Adaptive Fitness Approach (AFA) with updated gallery set, N = 50.

Initialize *gallery set*
Compute *feature vectors*
For *frame= 1, 2, … N*
    Project *probe image*
    Compute *fitness measure*
    Accumulate *fitness measure*
End
Identify *using highest accumulated fitness value*

Fig. 4. Pseudo code for Adaptive Fitness Approach (AFA) with fixed gallery set, N= 50.

An example of how the accumulated fitness measure is employed in the video recognition process with updating of the gallery is depicted in Figures 5 and 6. The feature vectors in Figure 5 are derived from PCA where in Figure 6 feature vectors come from LBP. In this example the probe video belongs to person # 1. The accumulated fitness measure in both figures show clearly that the accumulated fitness corresponding to person # 1 increases while for all other people it is insignificant. Number of training images for each person in this example was n=1 using the controlled scenario (see first row in table 1).

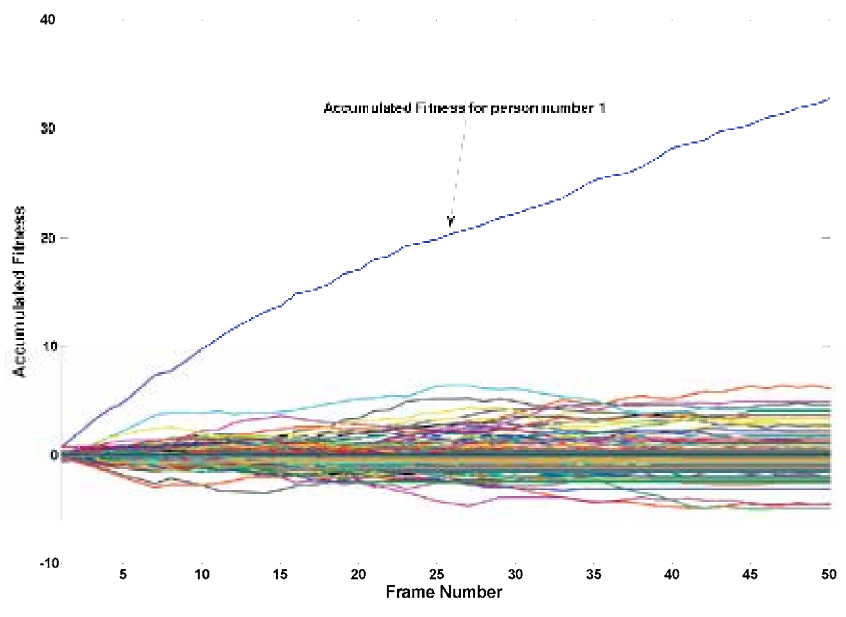


Fig. 5. Example of fitness accumulation through the frames for 1st video sequence of person number 1 using AFA with PCA.

4.2 Adaptive Fitness Fusion Approach (AFFA)

To recognize an individual human beings use more than one feature such gait, face, body shape and even wearing. A simple fusing technique is employed. The individual fitness

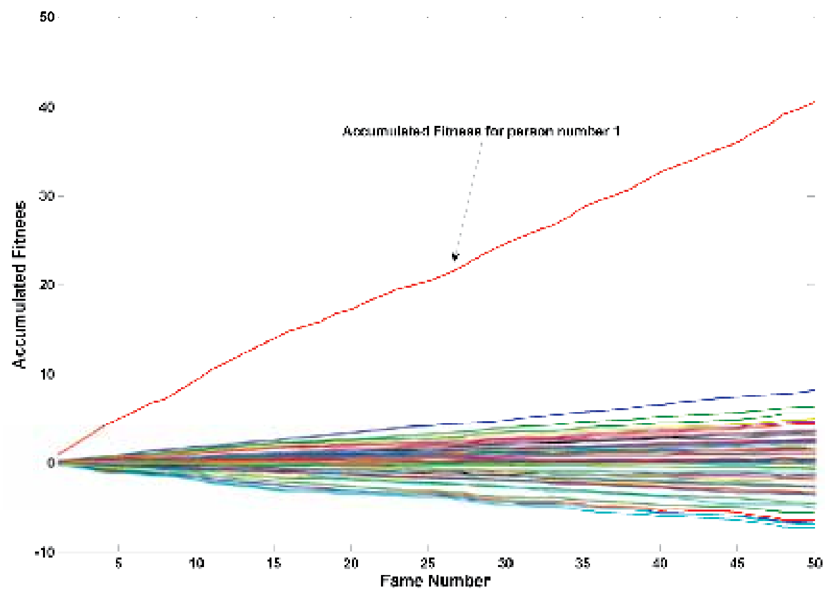


Fig. 6. Example of fitness accumulation through the frames for 1st video sequence of person number 1 using AFA with LBP.

measures coming from PCA and LBA are simply added. The recognition system based on feature vector fusion is the same as before. In the same manner, at the end of processing all the frames the individual with the highest fitness value is declared to be the correct subject. Figure 7 and 8 show the pseudo codes for the proposed fitness fusion idea with fixed and updated gallery set respectively.

Initialize *gallery set*
Compute *feature vectors using PCA*
Compute *feature vectors using LBP*
For *frame= 1,2,…N*
  Project *probe image using PCA*
  Project *probe image using LBP*
  Compute *fitness measure* Φ_{PCA}
  Compute *fitness measure* Φ_{LBP}
  Sum $\Phi_{total} = \Phi_{PCA} + \Phi_{LBP}$
  Accumulate *fitness measure* Φ_{total}
End
Identify *using highest accumulated total fitness value*

Fig. 7. Pseudo code for AFFA approach with fixed gallery set, *N*= 50.

Initialize *gallery set*
For *frame= 1,2,…N*
    Compute *feature vectors using PCA*
    Compute *feature vectors using LBP*
    Project *probe image using PCA*
    Project *probe image using LBP*
    Compute *fitness measure* Φ_{PCA}
    Compute *fitness measure* Φ_{LBP}
    Sum $\Phi_{total} = \Phi_{PCA} + \Phi_{LBP}$
    Accumulate *fitness measure* Φ_{total}
    Discard *gallery images with lowest fitness values*
    Update *gallery set*
End
Identify *using highest accumulated total fitness value*

Fig. 8. Pseudo code for AFFA Approach with updated gallery set, N= 50.

5. Simulation results and discussions

Figures 9 to 14 show the performance of the proposed AFA with updated and fixed gallery. AFA used both LBP and PCA for feature extraction and the results were compared against single frame based PCA and LBP methods, respectively. The three scenarios were shown in these figures with 1 and 5 training images in the gallery set. Both updated and fixed galleries show high competitive results.

The performance of the system is tested using BANCA database under 3 scenarios: controlled, degraded and adverse. For each scenario there are 52 people. For each individual there are 4 videos. The initial gallery is formed from varying number of training images per individual. For this study the numbers ranged from 1 to 10 as 2nd column of table1 depicts.

Usually human beings recognize people by fusing more than one feature. Here we show how the simple approach can be extended to benefit from different feature vectors. This fusion further improves the performance significantly. Again we employ features derived from PCA and LBP for simplicity and convenience.

Due to the fact that the performance of the AFA without fusion was very high (almost 100%) in order to faithfully see the improvement of fusion we increased the video database. As explained in section 3, the Banca database consists of 52 people with 3 scenarios and 4 recordings for each scenario. We treated the 4 recordings of each individual in each scenario as a different individual. This modification accounts to using 208 subjects with 3 different

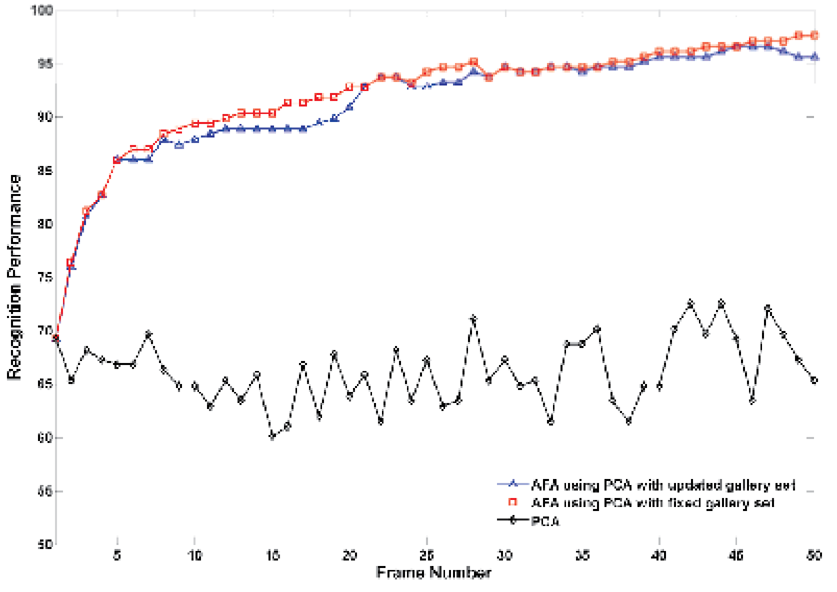


Fig. 9. Performance in controlled scenario with 1 training image per video with updated and fixed gallery set using PCA.

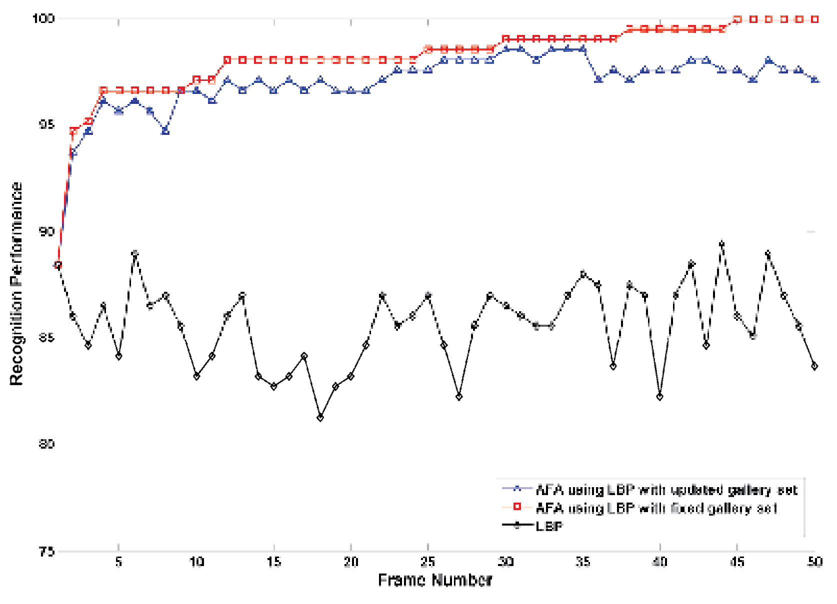


Fig. 10. Performance in controlled scenario with 1 training image per video with updated and fixed gallery set using LBP.

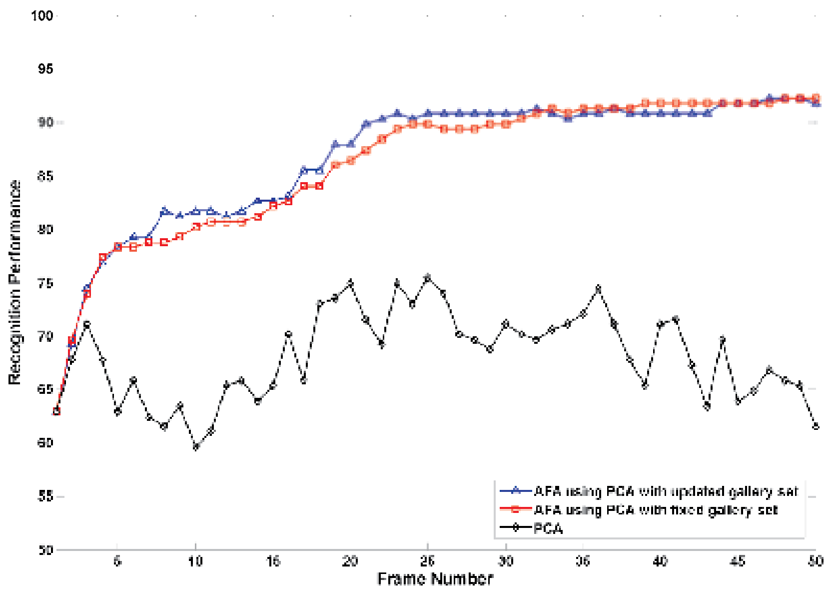


Fig. 11. Performance in adverse scenario with 1 training image per video with updated and fixed gallery set using PCA.

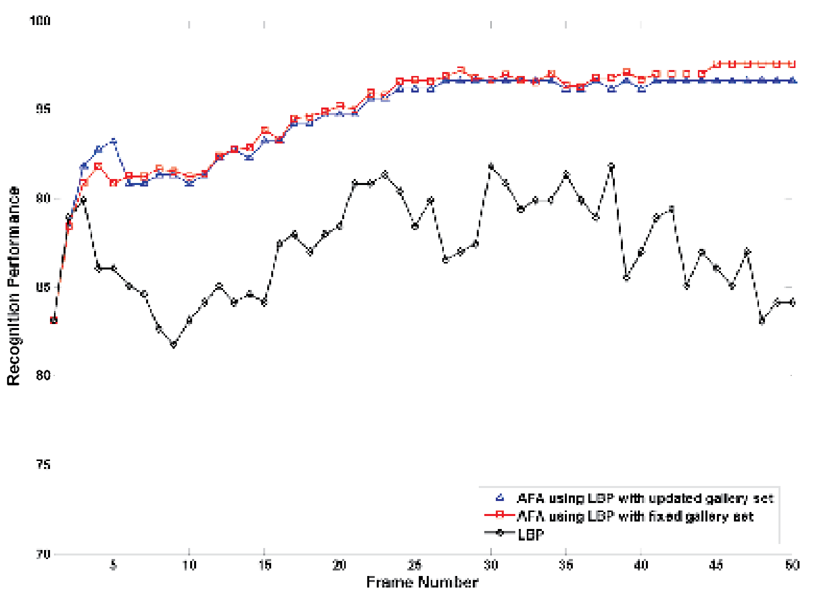


Fig. 12. Performance in adverse scenario with 1 training image per video with updated and fixed gallery set using LBP.

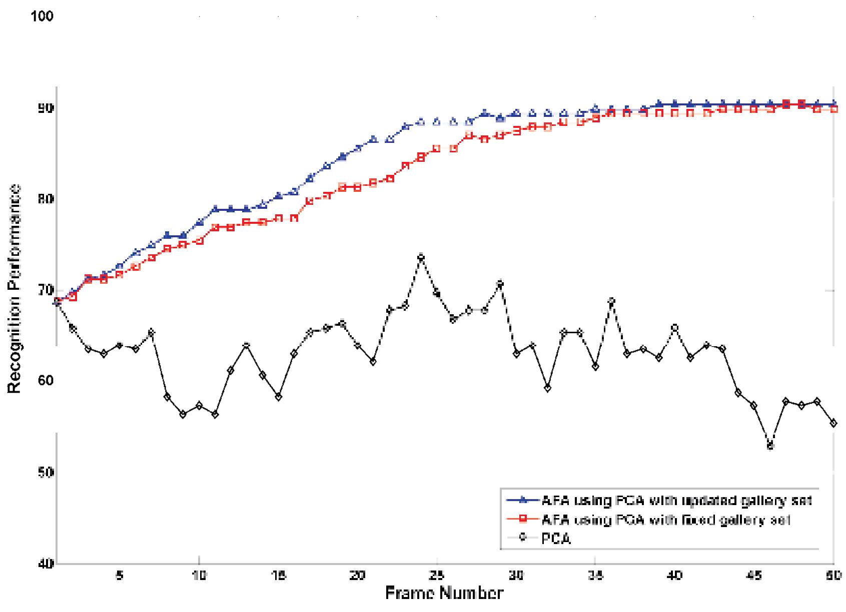


Fig. 13. Performance in degraded scenario with 1 training image per video with updated and fixed gallery set using PCA.

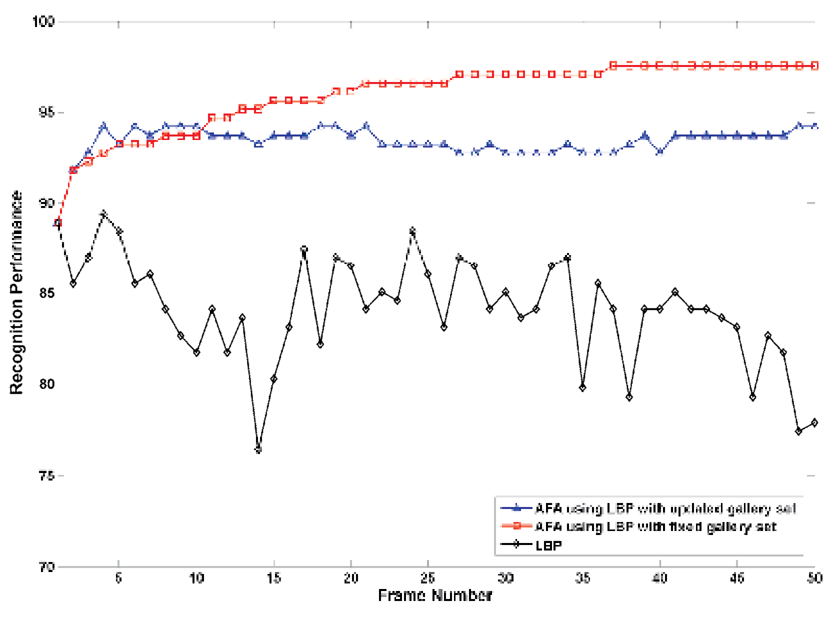


Fig. 14. Performance in degraded scenario with 1 training image per video with updated and fixed gallery set using LBP.

Scenario	# of gallery images per individual (n)	Recognition Performance (%)		
		AFA updated gallery set	AFA fixed gallery set	PCA/LBP
Controlled	1	95.67 / 97.12	97.60 / 100	66.25 / 85.62
	2	97.60 / 98.56	98.56 / 100	77.17 / 90.51
	3	98.08 / 99.04	99.04 / 100	82.13 / 93.28
	5	99.04 / 100	99.52 / 100	89.14 / 96.58
	10	100 / 100	100 / 100	96.16 / 98.38
Degraded	1	90.39 / 94.23	89.90 / 97.60	63.06 / 84.09
	2	95.67 / 95.67	96.15 / 98.08	73.48 / 88.30
	3	96.63 / 98.56	96.63 / 98.56	78.25 / 91.30
	5	98.08 / 99.52	97.12 / 99.04	84.76 / 94.43
	10	100 / 100	100 / 100	97.15 / 97.63
Adverse	1	91.83 / 96.63	92.31 / 97.60	68.17 / 87.24
	2	95.19 / 99.52	96.63 / 98.56	78.65 / 92.76
	3	98.56 / 100	98.08 / 99.56	84.25 / 95.41
	5	99.52 / 100	99.04 / 100	90.35 / 97.39
	10	100 / 100	100 / 100	99.04 / 98.89

Table 1. Performance of the adaptive fitness approach (AFA) using different number of training images from BANCA database with fixed and updated gallery set.

Scenario	# of gallery images per individual (n)	Recognition Performance (%)					
		updated gallery set			fixed gallery set		
		AFA_{PCA}	AFA_{LBP}	AFFA	AFA_{PCA}	AFA_{LBP}	AFFA
Controlled	1	68.27	80.77	83.77	68.75	80.77	84.25
	2	79.33	88.94	89.90	78.85	88.94	89.90
	3	83.66	90.87	92.79	83.17	90.87	92.79
	5	89.90	93.27	95.67	90.38	93.27	96.15
	10	95.67	97.60	100	94.23	97.60	100
Degraded	1	70.67	75.85	78.85	71.15	75.85	79.33
	2	80.29	84.62	87.50	78.85	84.62	87.02
	3	83.17	89.90	90.38	83.65	89.90	90.87
	5	89.90	92.79	94.71	88.46	92.79	94.23
	10	91.83	95.19	100	92.31	95.19	99.52
Adverse	1	68.75	74.04	78.85	69.23	74.04	79.33
	2	73.08	77.88	83.25	73.56	77.88	83.73
	3	80.29	85.10	89.54	79.81	85.10	89.54
	5	86.06	90.87	95.67	84.62	90.87	95.19
	10	93.27	96.63	100	92.79	96.63	100

Table 2. Performance of the adaptive fitness Fusion approach (AFFA) using different number of training images from BANCA database with fixed and updated gallery set.

scenarios. This is far more challenging since the 4 recordings of each individual are quit similar in terms of feature vectors. The results of this modification in the database size together with the adaptive fitness fusion approach (AFFA) results between LBP and PCA with updated and fixed gallery sets are shown in table 2.

The graphs in figure 15 to figure 20 show examples of the performance results of the proposed fitness fusion in the three different database scenarios (controlled, degraded, adverse) with 1 and 5 training images. The results shown in these figures are obtained using the scheme with fixed gallery set. It is clear in all figures that the fusing of separately obtained the fitness values Φ_{PCA} and Φ_{LBP} using the PCA and LBP feature vectors helped to improve the performance of the system. For example, in figure 15 the performance of AFA approach in the degraded scenario with 1 training image was 71.16 % and 75.85% using PCA and LBP, respectively. When fusion technique AFFA was applied the performance increased to 79.33 %. In figure 16, the training images were increased from 1 to 5 images in the same scenario. With AFA the performance was 88.46 % and 92.79% using PCA and LBP, respectively, and with AFFA it reached 94.23%. Same observation can be made for the other two database scenarios with different training images in figures 17 to 20.

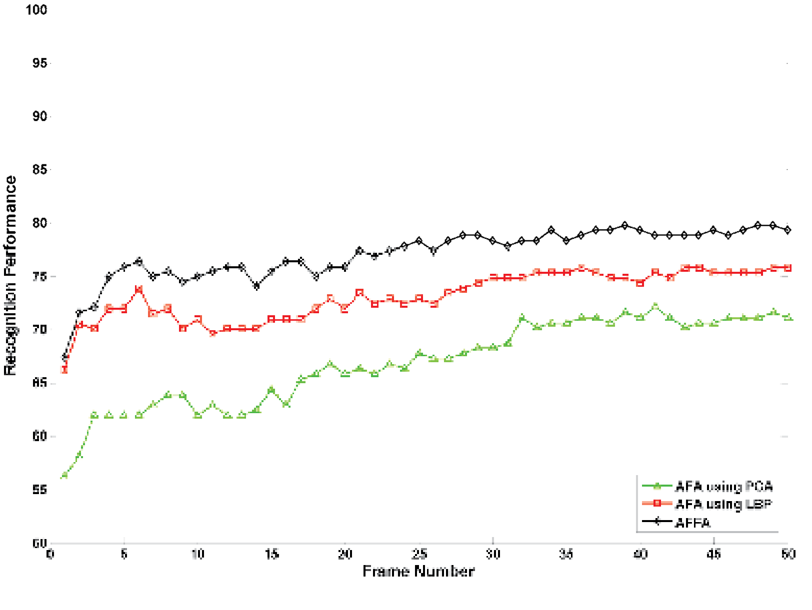


Fig. 15. Performance in degraded scenario with 1 training image per video with fixed gallery set.

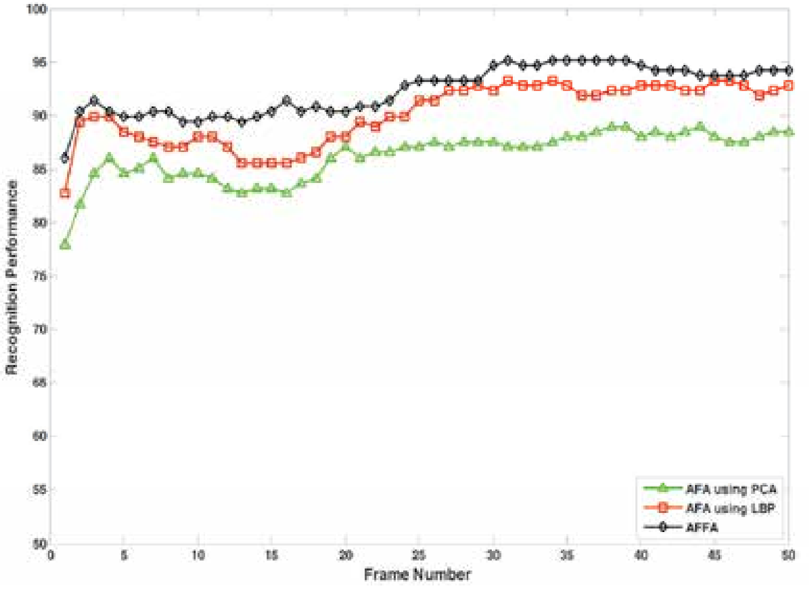


Fig. 16. Performance in degraded scenario with 5 training image per video with fixed gallery set.

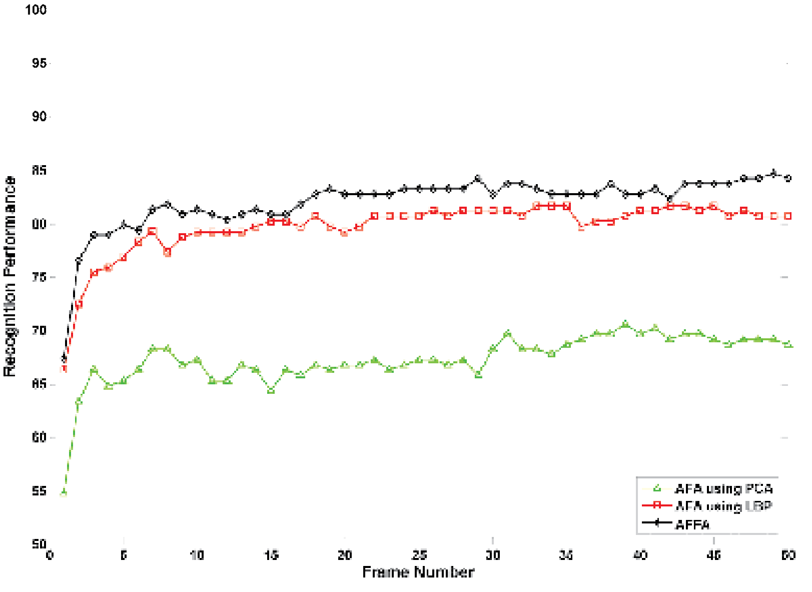


Fig. 17. Performance in controlled scenario with 1 training image per video with fixed gallery set.

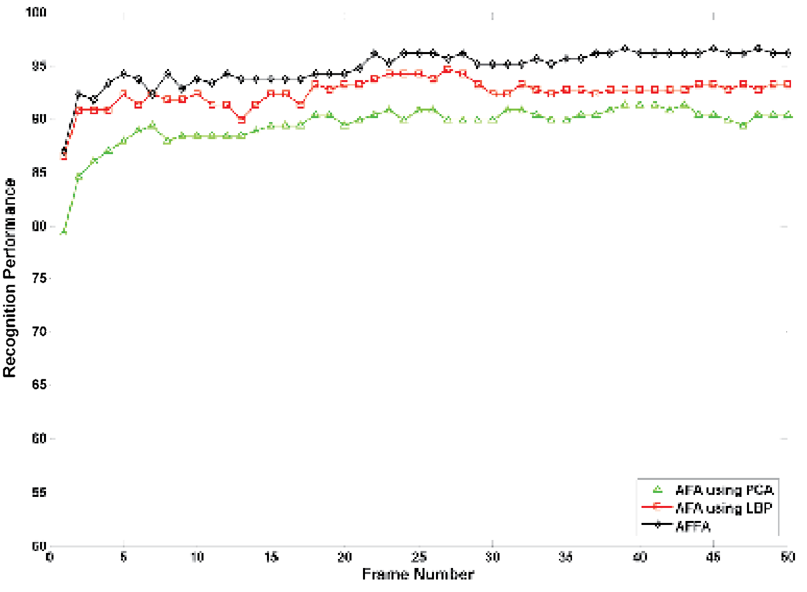


Fig. 18. Performance in controlled scenario with 5 training image per video with fixed gallery set.

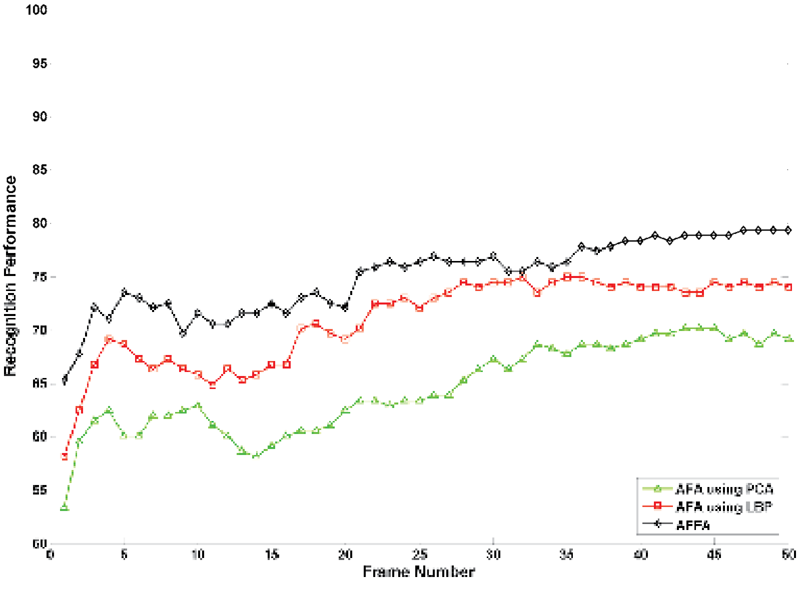


Fig. 19. Performance in adverse scenario with 1 training image per video with fixed gallery set.

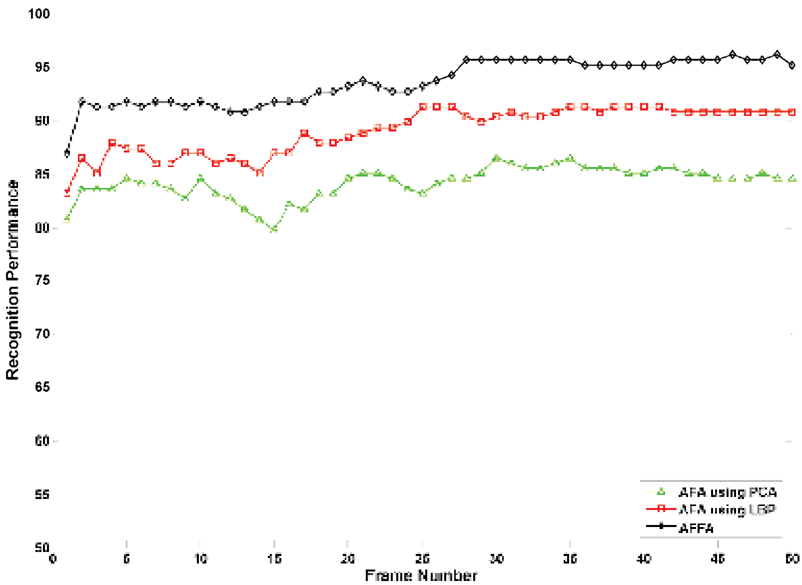


Fig. 20. Performance in adverse scenario with 5 training image per video with fixed gallery set.

6. Conclusion

In this chapter a new biologically inspired approach called Adaptive Fitness Approach (AFA) for identifying faces form video sequences is proposed. The fitness value of each image in the gallery set is calculated and accumulated as the probe video frames are processed. To schemes are used with the AFA approach. First scheme employs discarding of unfit images from gallery followed by an update of the feature vectors. In the second scheme gallery and thus the feature vectors are kept fixed.

In order to demonstrate the proposed AFA approach with updated and fixed gallery schemes, PCA and LBP derived features are employed for convenience. Performance of both schemes is far superior to single frame based PCA or LBP approaches. Even for very small number of training images. The adaptive fitness framework is also shown to conveniently accommodate fusing of different feature vectors with further and significant improvement in recognition performance over the AFA with single feature.

7. References

Ahonen, T.; Hadid, A.; & Pietikainen, M. (2004). Face Recognition with Local Binary Patterns, in: *Processing European Conference on Computer Vision*, pp. 469-481

Belhumeur, P.; Hespanha, J. & Kriegman, D. (1997). Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection, *IEEE Transaction on pattern analysis and machine learning*, Vol. 19, No.7, pp.711-720

Brunelli, R. & Poggio, T. (1993). Face Recognition: Features Versus Templates, *IEEE Transaction on pattern analysis and machine learning*, Vol. 15, No. 10, pp. 1042-1052

Eleyan, A. & Demirel, H. (2007). PCA and LDA Based Neural Networks for Human Face Recognition, In: *Face Recognition*, Kresimir Delac and Mislav Grgic (Ed.), pp. 93-106, ISBN: 978-3-902613-03-5, I-Tech Education and Publishing, Croatia

Eleyan, A.; Ozkaramanli, H. & Demirel, H. (2008). Complex Wavelet transform-based Face Recognition. *EURASIP Journal on Advances in Signal Processing*, Vol. 2008, Article ID 185281, 13 pages

Eleyan, A.; Ozkaramanli, H. & Demirel, H. (2009). Adaptive and Fixed Eigenspace Methods with a Novel Fitness Measure for Video based Face Recognition, *24th International Symposium on Computer and Information Sciences*, pp. 636-640

Eleyan, A. & Demirel, H. (2011). Co-Occurrence Matrix and Its Statistical Features as a New Approach for Face Recognition, *Turkish Journal of Electrical Engineering & Computer Sciences*, Vol. 19, pp. 97-107

Kirby, M. & Sirovich, L. (1990). Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces. *IEEE Transaction on pattern analysis and machine learning*, Vol. 12, pp. 103-108

Li, B. & Chellappa, R. (2002). A generic Approach to Simultaneous Tracking and Verification in Video, *IEEE Transaction on Image Processing*, Vol. 11, pp. 530-544

Martinez, A. M. & Kak, A. C. (2001). PCA versus LDA, *IEEE Transaction on pattern analysis and machine learning*, Vol. 23, pp. 228-233

Nilsson, M.; Nordberg, J. & Claesson, I. (2007). Face Detection using Local SMQT Features and Split up Snow Classifier, *ICASSP 2007* 2 pp: II-589-II-592

Ojala, T.; Pietikainen, M.; Maenpaa, T. (2002). Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns *IEEE Transaction on pattern analysis and machine learning*, Vol. 24, No. 7, pp. 971-987

Philipps, P.J.; Moon, H.; Rivzi, S. & Ross, P. (2000). The Feret Evaluation Methodology for Face-Recognition Algorithms, *IEEE Transaction on pattern analysis and machine learning*, Vol. 22, pp. 1090-1104

Popovici, V; Thiran, J.; Bailly-Bailliere, E.; Bengio, S.; Bimbot, F.; Hamouz, M.; Kittler, J.; Mariethoz, J.; Matas, J.; Messer, K.; Ruiz, B. & Poiree, F. (2003). The BANCA Database and Evaluation Protocol, in: *Proceedings of the International Conference on Audio- and Video-Based Biometric Person Authentication*, pp. 625-638

Steffens, J.; Elagin, E. & Neven, H. (1998). Personspotter—Fast and Robust System for Human Detection, Tracking, and Recognition, in: *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pp. 516-521

Sinha, P.; Balas, B.; Ostrovsky, Y. & Russell, R. (2006). Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About, *Proceedings of the IEEE 94,* Vol. 11, pp. 1948-1962

Turk, M. & Pentland, A. (1991). Eigenfaces for Recognition, *Journal of Cognitive Neuroscience*, vol. 3, pp. 71-86

Wechsler, H.; Kakkad, V.; Huang, J.; Gutta, S. & Chen, V. (1997). Automatic Video-Based Person Authentication using The RBF Network, in: *Proceedings of the International Conference on Audio- and Video-based Biometric Person Authentication*, pp. 85-92

Wiskott, L.; Fellous, J.M.; Kruger, N. & Malsburg, C. (1997). Face Recognition by Elastic Bunch Graph Matching, *IEEE Transaction on pattern analysis and machine learning*, Vol. 19, No. 7, pp. 775-780

Zhao, W.; Chellappa, R.; Rosenfeld, A. & Phillips, P.J. (2003). Face Recognition: A Literature Survey, *ACM Computing Surveys*, pp. 399-458

Zhou, S.; Krueger, V. & Chellappa, R. (2003). Probabilistic Recognition of Human Faces from Video, *Computer Vision and Image Understanding*, Vol. 91, pp. 214-245

9

Real Time Robust Embedded Face Detection Using High Level Description

Khalil Khattab[1], Philippe Brunet[1], Julien Dubois[2] and Johel Miteran[2]
[1]DRIVE – ISAT, University of Burgundy,
[2]Le2i, University of Burgundy,
France

1. Introduction

Face detection is a fundamental prerequisite step in the process of face recognition. It consists of automatically finding all the faces in an image despite the considerable variations of lighting, background, appearance of people, position/orientation of faces, and their sizes. This type of object detection has the distinction of having a very large intra-class, making it a particularly difficult problem to solve, especially when one wishes to achieve real time processing.

A human being has a great ability to analyze images. He can extract the information about it and focus only on areas of interest (the phenomenon of attention). Thereafter he can detect faces in an extremely reliable way. Indeed, a human being is able to easily locate faces in its environment despite difficult conditions such as occlusions of parts of a face and bad lightening. Many studies have been conducted to try to replicate this process, automatically using machines, because face detection is considered as a prerequisite for many computer vision application areas such as security, surveillance, and content based image retrieval.

Over the last two decades multiple robust algorithmic solutions were proposed. However, researches in the field of computer vision and pattern recognition in particular tend to focus on the algorithmic and functional parts. This generally leads to implementations with little constraints of time, computing power and memory. Most of these techniques, even if they achieve good performance in terms of detection, are not suited for real time application systems. Nonetheless, Boosting–based methods, firstly introduced by Viola and Jones in (Viola & Jones, 2001; 2002), has led the state-of-the-art in face detection systems. These methods present the first near real time robust solution and by far the best speed / detection compromise in the state-of-the-art (up to 15 frames/s and 90% detection on 320x240 images). This family of detectors relies upon a cascade of several classification stages of progressive complexity (around 20-40 stages for face detection). Depending on its complexity, each stage contains several classifiers trained by a boosting algorithm (Freund & Schapire, 1995; Lienhart, Kuranov, & Pisarevsky, 2003; Viola & Jones, 2002)

These algorithms help achieving a linear combination of weak classifiers (often a single threshold), capable of real time face detection with high detection rates. Such a technique can be divided into two phases: Training and detection (through the cascade). While the training phase can be done offline and might take several days of processing, the final cascade detector should enable real-time processing. The goal is to run through a given image in order to find all the faces regardless of their scales and locations. Therefore, the image can be seen as a set of sub-windows that have to be evaluated by the detector which selects those containing faces. This approach is optimized for a sequential implementation but this implementation has two major drawbacks: high dependency between the different stages of the detector and irregularity in time processing.

Most of the Boosting–based face detection solutions deployed today are general purpose processors software. But with the development of faster camera sensors which allows higher image resolution at higher frame-rates, these software solutions are not always working in real time. Even more the current technology of multi-core processor cannot be exploited to its full limits because of the dependency between the different stages. Seeking some improvement over the software, several attempts were made trying to implement face detection on multi-FPGA boards and multiprocessor platforms using programmable hardware , however in almost all the cases the resulting implementation are capable to accelerate the detection but degrade the detection accuracy.

The major difficulties in a parallel implementation of the cascade detector (boosted based methods) are the full dependency between the consecutive stages and classifier repartition which is optimized for sequential implementation. Based on this observation and our belief that a useful acceleration of the face detection should not compromise the detection performances, our main contribution is a new structure that exploits intrinsic parallelism of a boosting-based object detection algorithm without compromising its accuracy. At first we present a new stage grouping capable of equally partition the computation complexity of the algorithm. Based on this partitioning, a new parallel model is proposed. This model is capable of exploiting the parallelism and the pipelining in these algorithms, and provides regularity in time processing. It can also be customizable according to the cascade in use.

This chapter also shows that a hardware implementation is possible using high-level SystemC description models. SystemC enables PC simulation that allows simple and fast testing and leaves our structure open to any kind of hardware or software implementation since SystemC is independent from all platforms. The processing blocs are modeled using SystemC. We show that, using a SystemC description model paired with a mainstream automatic synthesis tool, can lead to an efficient embedded implementation. We also display some of the tradeoffs and considerations, for this implementation to be effective.

Finally, using the results of the processing blocks' implantations, we define a new architectural structure of the implementation including the interconnectivity of the memory blocks and the number and the type of the used memories. This final system proves capable of achieving 47 frames per second for 320x240 images as well as keeping the same detection accuracy as the original method. In the end, we show a detailed comparison between our system and the other state-of-the-art embedded implementation for boosting based face detection.

This chapter can be considered as a continuation of previously published work (Khattab, Dubois & Miteran 2009) in which we proposed a new architecture for an embedded real-time face detector based on a fast and robust family of methods, initiated by Viola and Jones. However only parts of the processing blocks were implemented, memories types and interconnection wasn't optimized and the system validation was made in simulation

2. Review of Boosting based object detectors

Object detection is defined as the identification and the localization of all image regions that contain a specific object regardless of the object's position and size, in an uncontrolled background and lightning. It is more difficult than object localization where the number of objects as well as their size are already known. The object can be anything from a vehicle, human face (Figure 1), human hand, pedestrian (Viola, Jones, & Snow, 2003), etc. The majority of the boosting based object detectors work-to-date has primarily focused on developing novel face detection since it is very useful for a large array of applications. Moreover, this task is much trickier than other object detection tasks, due to the typical variations of hair style, facial hair, glasses and other adornments.

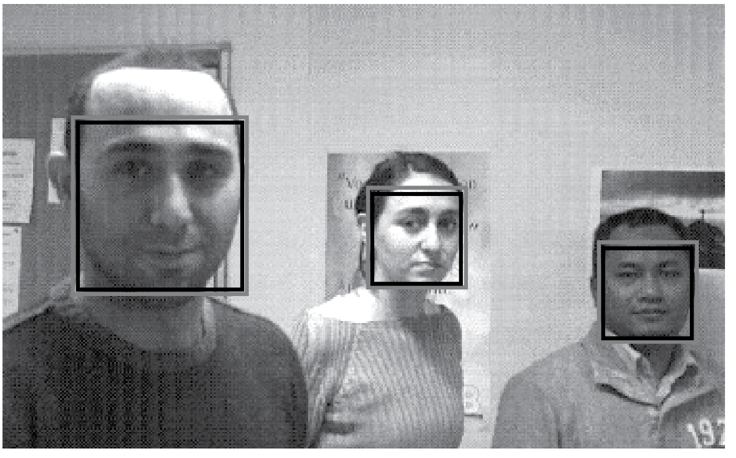


Fig. 1. Example of face detection

2.1 Theory of Boosting Based object detectors

2.1.1 Cascade detection

The structure of the cascade detector (introduced by Viola and Jones) is that of a degenerated decision tree. It is constituted of successively more complex stages of classifiers (Figure 2). The objective is to increase the speed of the detector by focusing on the promising zones of the image. The first stage of the cascade will look over for these promising zones and indicates which sub-windows should be evaluated by the next stage. If a sub-window is labeled at the current classifier as non-face then it will be rejected and the decision upon it is terminated. Otherwise it has to be evaluated by the next classifier. When a sub-window survives all the stages of the cascade, it will be labeled as a face. Therefore the complexity increases dramatically with each stage, but the number of sub-windows to be evaluated will decrease more tremendously. Over the cascade the overall detection rate should remain high while the false positive rate should decrease aggressively.

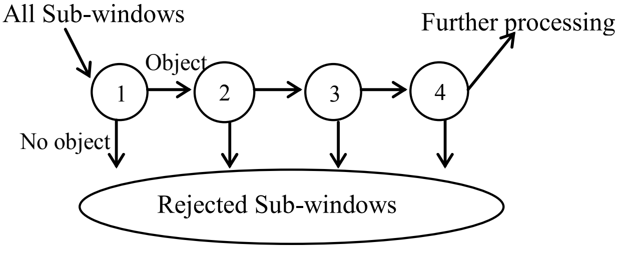


Fig. 2. Cascade detector

2.1.2 Features

To achieve a fast and robust implementation, Boosting based faces detection algorithms use some rectangle Haar-like features (shown in Figure 3) introduced by (Papageorgiou, Oren, & Poggio, 1998): Two-rectangle features (A and B), Three-rectangle features (C) and Four-rectangle features (D). They operate on grayscale images and their decisions depend on the threshold difference between the sum of the luminance of the white region(s) and the sum of the luminance of the gray region(s).

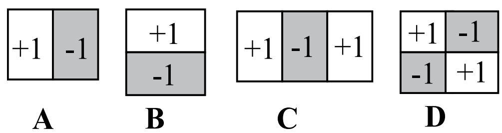


Fig. 3. Rectangle Features

Using a particular representation of the image so-called the Integral Image (II), it is possible to compute very rapidly the features. The II is constructed of the initial image by simply taking the sum of luminance value above and to the left of each pixel in the image:

$$ii(x,y) = \sum_{x'<x,y'<y} i(x',y') \tag{1}$$

Where $ii(x,y)$ is the integral image and $i(x,y)$ is the original image pixel's value. Using the Integral Image, any sum of luminance within a rectangle can be calculated from II using four array references (Figure 4). After the II computation, the evaluation of each feature requires 6, 8 or 9 array references depending on its type.
However, assuming a 24x24 pixels sub-window size, the over-complete feature set of all possible features computed in this window is 45,396: it is clear that a feature selection is necessary in order to keep real-time computation time compatibility. This is one of the roles of the Boosting training step.

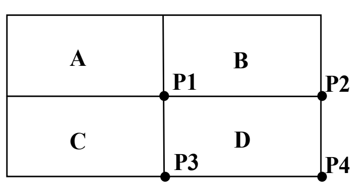


Fig. 4. The sum of pixels within Rectangle D can be calculated by using 4 array references; SD= II [P4] – (II [P3] + II [P2] – II [P1])

2.1.3 Weak classifiers and Boosting training

A weak classifier $h_j(x)$ consists of a feature f_j, a threshold θ_j and a parity p_j indicating the direction of the inequality sign:

$$h_j(x) = \begin{cases} 1 \; if \;\; p_j f_j(x) < p_j \theta_j \\ 0 \; otherwise \end{cases} \tag{2}$$

Boosting algorithms (Adaboost and variants) are able to construct a strong classifier as a linear combination of weak classifiers (here a single threshold) chosen from a given, finite or infinite, set, as shown in Equation 3.

$$h(x) = \begin{cases} 1 \sum_{t=1}^{T} \alpha_t h_t(x) > \theta \\ 0\, otherwise \end{cases} \tag{3}$$

Where θ is the stage threshold, a_t is the weak classifier's weight and T the total number of weak classifiers (features). This linear combination is trained in cascade in order to have better results.

There, a variant of Adaboost is used for learning object detection; it performs two important tasks: feature selection from the features defined above; and constructing classifiers using selected features. The result of the training step is a set of parameters (array references for features, constant coefficients of the linear combination of classifiers, and thresholds values selected by Adaboost). This set of features parameters can be stored easily in a small local memory.

2.2 Previous implementations

The state-of-the-art initial prototype of this method, also known as Viola-Jones algorithm, was a software implementation based on trained classifiers using Adaboost. The first implementation shows some good potential by achieving good results in terms of speed and accuracy; the prototype can achieve 15 frames per second on a desktop computer for 320x240 images. Such an implementation on general purpose processors offers a great deal of flexibility, and it can be optimized with little time and cost, thanks for the wide variety of the well-established design tools for software development. However, such implementation can occupy all CPU computational power for this task alone; nevertheless, face/object detection are considered as prerequisite step for some of the main application such as biometric, content-based image retrieval systems, surveillance, auto-navigation, etc. Therefore, there is more and more interest in exploring an implementation of accurate and efficient object detection on low cost embedded technologies. The most common target technologies are embedded microprocessors such as DSPs, pure hardware systems such as ASIC and configurable hardware such as FPGAs.

Lot of tradeoffs can be mentioned when trying to compare these technologies. For instance, the use of embedded processor can increase the level of parallelism of the application, but it costs high power consumption, all while limiting the solution to run under a dedicated processor.

Using ASIC can result better frequency performance coupled with high level of parallelism and low power consumption. Yet, in addition to the loss of flexibility, using this technology requires a large amount of development, optimization and implementation time, which elevates the cost and risk of the implementation.

FPGAs can have a slightly better performance/cost trade-offs then previous two, since it permits high level of parallelism coupled with some design flexibility. However some restriction in design space, costly rams connections as well as lower frequency comparing to ASIC, can rule-out it use for some memory heavy applications.

For our knowledge, few attempts were made trying to implement Boosting based face detection on embedded platforms. Nevertheless, these proposed architectures were configurable hardware based implementations and most of them couldn't achieve high detection frame rate speed while keeping the detection rate close of that's of the original implementation. For instance, in order to achieve 15 frames per second for 120x120 images, Wei et al. (Wei, Bing, & Chareonsak, 2004) choose to skip the enlargement scale factor from

1.25 to 2. However such a maneuver would lower the detection rate dramatically. Theocharideset al. (Theocharides, Vijaykrishnan, & Irwin, 2006) has proposed a parallel architecture taking advantage of a grid array processor. This array processor is used as memory to store the computation data and as data transfer unit, to aid in accessing the integral image in parallel. This implementation can achieve 52 frames per second at a 500 MHz frequency. However, details about the image resolution were not mentioned. Another complex control scheme to meet hard real-time deadlines is proposed in (M Yang, Wu, Crenshaw, Augustine, and Mareachen 2006). It introduces a new hardware pipeline design for Haar-like feature calculation, and a system design exploiting several levels of parallelism. But it sacrifices the detection rate and it is better fitted for portrait pictures. And more recently, an implementation with NoC (Network-on-Chip) architecture is proposed in (Lai, Marculescu, Savvides, & Chen, 2008) using some of the same element as (Theocharides, Vijaykrishnan, & Irwin, 2006), this implementation achieves 40 frames per second for 320x240 images. However detection rate of 70% was well below the software implementation (82% to 92%), due to the use of only 44 features (instead of about thousands).

3. Global parallelism

In this section we provide a detailed analysis of the boosting based face detection algorithm, in order to extract as much useful information for designing an efficient parallel architecture. For this, we first present an analysis of the sequential implementation. We then analyze the different stages of the cascade, and the computational complexity of each one of them. Finally, we propose a parallel structure to accelerate this algorithm.

3.1 Sequential implementation

The strategy used in software implementation consists of processing each sub-window at a time. The processing on the next sub-window will not trigger until a final decision is taken upon the previous one i.e. going through a set of features as a programmable list of coordinate rectangles. The processing time of an image depends on two factors: the processing time of each sub-window and the number of sub-windows to process.

The processing time of a sub-window can vary dramatically depending on the complexity of its content. For example, an image of uniform color will definitely take less time to process than an image containing several faces. For this reason, the cascade-like detection algorithms are irregular and not predictable. In fact, Viola and Jones have already indicated that the speed of the cascade detector depends on the image content and accordingly the average number of weak classifiers evaluated per sub-window on an image sequence. Moreover, their tests showed that, on average, 10 weak classifiers are evaluated per sub-window. These tests were done using CMU image database (Rowley, Baluja, & Kanade, 1998).

3.1.1 OpenCV implementation

Several variants of Boosting based face detection can be found in today's literature. However the principal of cascade detection remain the same in almost all of these variants. As for the cascade /classifiers, we chose to use the database found on Open Computer Vision Library (OpenCv[1]). OpenCV provides the most used trained cascade/classifiers datasets and face-

[1]OpenCv. (2009). Open source computer vision library. http://sourceforge.net/projects/opencvlibrary/

detection software today. The particular classifiers, used on this library, are those trained with a base detection window of 24x24 pixels, using Adaboost. These classifiers are created and trained, by Lienhart et al (Lienhart & Maydt, 2002) , for the detection of upright front face detection. The detection rate of these classifiers is between 80% and 92%, depending on the images Database. This cascade includes more than 2500 features spread on 25 stages.
Using this sequential implementation, we decided to investigate each stage. For instance, the first stage classifier should be separated from the rest since it requires processing all the possible sub windows in an image, while each of the other relies on the results of previous stage and evaluates only the sub windows that passed through.

3.1.2 Classification stages

As mentioned earlier the first stage of the cascade must run all over the image and rejects the sub-windows that do not fit the criteria (no face in the window). The detector is scanned across locations and scales, and subsequent locations are obtained by shifting the window some number of pixels k. Only positive results trigger in the next classifier.
The addresses of the positive sub-windows are stored in a memory, so that next classifier could evaluate them and only them in the next stage. Figure 5 shows the structure of such classifier. The processing time of this first stage is stable and independent from the image content; the algorithm here is regular.

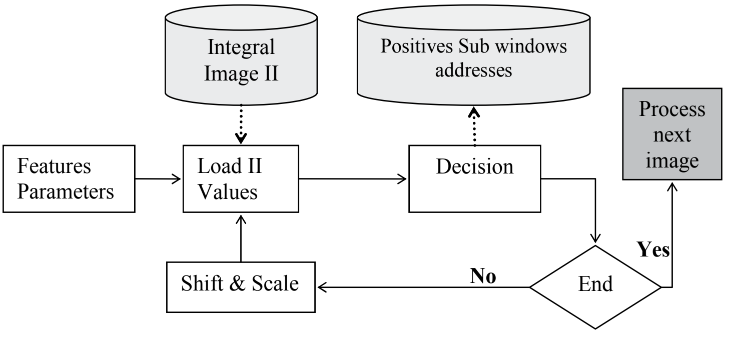


Fig. 5. First cascade stage

The other classification stages, shown in Figure 6, do not need to evaluate the whole image. Each classifier should examine only the positive results, given by the previous stage, by reading their addresses in the memory, and then takes a decision upon each one (reject or pass to the next classifier stage).
Each remaining stage is expected to reject the majority of sub-windows and keep the rest to be evaluated later in the cascade. As a result, the processing time depends largely on the number of positive sub-windows resulted from the previous stage. Moreover the classifier complexity increases with the stage level.

3.1.3 Full sequential implementation analysis

For a 320x240 image, scanned on 11 scales with a scaling factor of 1.25 and a step of 1.5, the number of total sub-windows to be investigated is 105,963. Based on tests done in (Viola & Jones, 2001), an average of 10 features are evaluated per sub-window. As a result, the estimated number of decision made over the cascade, for a 320x240 image, is 1.3 million as

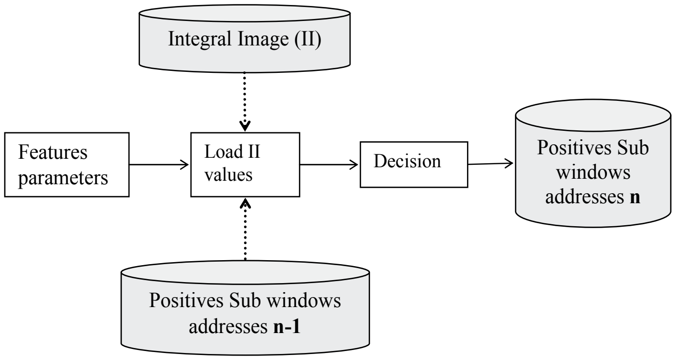


Fig. 6. n_{th} Stage classifier

on average. Thereafter around 10 million memory access (since each decision needs 6, 8 or 9 array references to calculate the feature in play). Note that the computation time of the decision (linear combination of constants) as well as the time needed to build the integral image, are negligible comparing to the overall memory access time.

Considering the speed of the memory is 10 ns per access (100 MHz), the time needed to process a full image is around 100 ms (about 10 images per second). However, this rate can vary with the image's content.

3.2 Possible parallelism

We applied the frontal face detector, "Discrete AdaBoost" of OpenCV, on the CMU image database in order to analyze the number of sub-windows rejected per stage and subsequently the number weak classifiers (features) evaluated per sub-window. Indeed, 75 081 800 sub-windows have triggered a total of 668 659 962 evaluations of weak classifiers. Hence, only 9 weak classifiers are evaluated per sub-window on average.

Even more, these analysis revealed another major characteristic of the cascade implementation: The unbalance in processing loads between the different stages. This is caused by the fact that the boosting based face detection is optimized for sequential implementation. The training phase of the boosting methods is configured to reject as much sub-windows as early as possible.

On average, about 35% of the total memory access (and processing) load takes place in each of the first two stages while less than 32% take place in all of the remaining stages combined

In the rest of this section, we show how to take advantage of such unbalance in memory access in order to propose a feasible parallel model

3.2.1 Pipeline solution

The previous analysis of the OpenCV cascade revealed that more than a third of the memory access take place on each of the first two cascade stages while less than third in all remaining stages. This analysis leads us to suggest a new pipelined solution (shown in Figure 7) of 3 parallel blocks that work simultaneously: In the first two blocks we intend to implement respectively the first and second stage classifier, then a final block assigned to run over all remaining stages sequentially.

Unlike the state-of-the-art software implementation, the proposed structure tends to run each stage as a standalone block. Nevertheless, some intermediate memories between the stages must be added in order to stock the positively-labeled windows addresses.

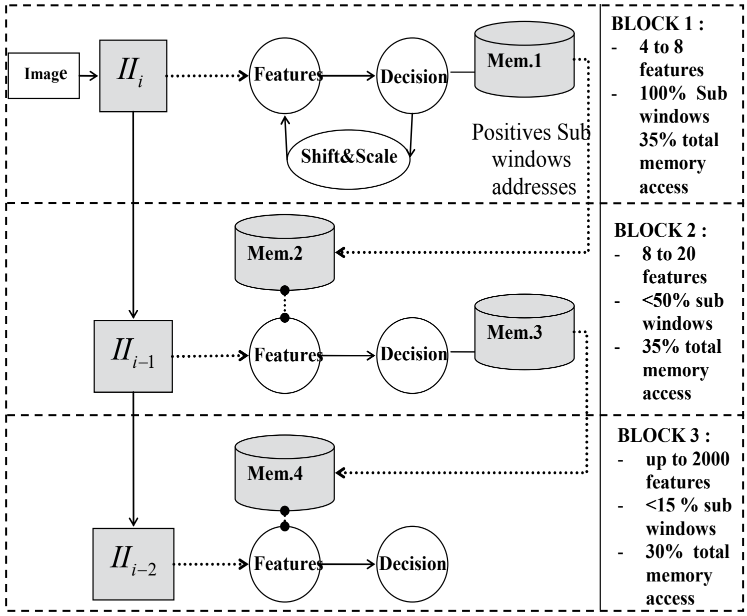


Fig. 7. Parallel structure

The new structure proposed above can upsurge the speed of the detector in one condition: since that the computation complexity is relatively small and the time processing depends heavily on the memory access, an integral image memory should be available for each block in order to gain benefit of three simultaneous memory accesses. Figure 7 shows the proposed parallel structure. At the end of every full image processing cycle, the positive results from Block1 trigger the evaluation of Block2. The positive results from Block2 trigger the evaluation of Block3. And the positive results from Block3 are labeled as faces. It should be noted that blocks cannot process simultaneously on the same image i.e. if at a given moment Block1 is working on the current image I_n, then Block2 should be working on the previous image I_{n-1} and Block3 should be working on the one before I_{n-2}.

This structure requires data dependency between the parallel blocks. The addresses of the sub-windows classified positively by Block 1 shall be transmitted to Block 2. Similarly, the addresses of sub-windows classified positively by Block 1 and 2 respectively, must be transmitted to the Block 3.

The large numbers of sub-windows addresses require the use of intermediate memories, which will manage the communication between the different blocks. At any given time, Block 1 processes on image I_n and stores the addresses of its positively labeled sub-windows in a memory (*mem.1*). At the same time Block 2 processes an image I_{n-1} but only the sub-windows positively labeled by the first and whose addresses are stored in memory *mem.2*. The addresses of sub-windows positively labeled by Block 2 are stored in a memory (*mem.3*). Respectively, Block 3 processes an image I_{n-2} , but only its sub-windows positively labeled by Block 2 and whose addresses are read from a memory mem.4. Block 3 works the same way as in the sequential implementation: the block run back and forth through all remaining stages, to finally give the addresses of the detected faces.

After each image cycle, and the memories *mem.1* and *mem.2* are swapped, same goes for *mem.3* and *mem.4*

This can be translated into the model shown in Figure 8. A copy of the integral image is available to each block, as well as, three pairs of logical memory are working in ping pong to accelerate the processing.

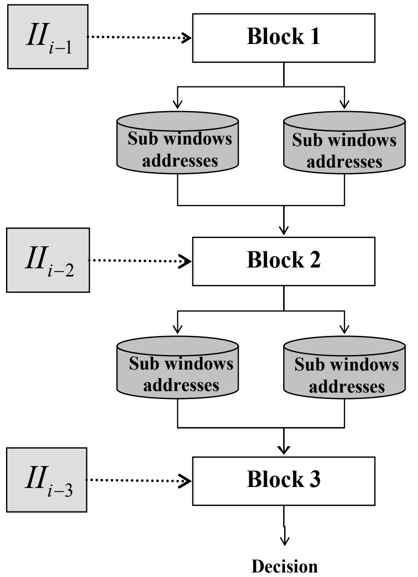


Fig. 8. Data Flow

The given parallel model ought to run at the same speed rate as its slower block. As mentioned earlier, the first stage of the cascade requires more access memory and therefore more time processing than the second stage alone or all the remaining stages together. In the first classifier stage, all 105,963 sub-windows should be inspected using three features with eight array references each. Therefore, it requires about 3.4 million of memory access per image. Using the same type of memory as in section 3.1.4, an image needs roughly 34 ms (29 images per second) of time processing.

3.2.2 Parallel model discussion

Normally the proposed structure should stay the same, even if the cascade structure changes, since most of the boosting cascade structures have the same properties as long as the first two cascade stages.

One of the major issues surrounding boosting based detection algorithms (especially when applied on to face detection in a non-constraint scene) is the inconsistency and the unpredictable processing time e.g. a white image will always takes a little processing time since no sub-window should be capable of passing the first stage of the cascade. As opposite, an image of thumbnails gallery will take much more time.

The proposed structure not only gives a gain in speed; this first stage happens to be the only regular one in the cascade, with fixed time processing per image. This means that we can mask the irregular part of the algorithm by fixing the detector overall time processing.

As a result, the whole system will not work at 3 times the speed of the average sequential implementation; but a little bit less. Further work in section 5 will show that the embedded implementation can benefit from some system teaks (pipelining and parallelism) within the computation that will make the architecture even faster.

Due to the masking phenomena in the parallel implementation, decreasing the number of weak classifiers can accelerate the implementation; but only if the first stage of the cascade is accelerated.
For this structure to be implemented effectively, its constraints must be taken into consideration. The memory, for instance, can be the most greedy and critical part; the model requires multiple memory accesses to be done simultaneously. The definition of an architectural structure of this representation depends on two factors: the nature of memory access and the desired performance of the system. For instance, the memory blocks of the integral images are used in the computation of the Haar-features rectangles. The integral images are stored in a linear fashion; however the reading access depends on the position, the size and the parameters of the weak classifier to be evaluated. It is for this reason that access to these memories are made randomly. On the other hand, the blocks of intermediate memories are used to store and read the addresses of the sub-windows. Both the write and the read of these addresses are done sequentially. To optimize performance of architecture, it is imperative to use memories appropriate to the nature of each of the different access types. Thus, as we shall see in the implantation section 4, we recommend using two different types of memory.
It is obvious that a generic architecture (a processor, a global memory and cache) will not be enough to manage up to seven simultaneous memory accesses on top of the processing, without crashing it performances.

4. Architecture definition and implementation

Flexibility and target architecture are two major criteria for any implementation. First, a decision has been taken upon building our implementation using a high level description model/language. Modelling at a high level of description would lead to quicker simulation, better bandwidth estimation, better functional validation, and more importantly it can help delaying the system orientation and thereafter delaying the hardware target.

4.1 SystemC description

C++ implements Object-Orientation on the C language. Many Hardware Engineers may consider that the principles of Object-Orientation are fairly remote from the creation of Hardware components. Nevertheless, Object-Orientation was created from design techniques used in Hardware designs. Data abstraction is the central aspect of Object-Orientation which can be found in everyday hardware designs with the use of publicly visible "ports" and private "internal signals". Moreover, component instantiation found in hardware designs is almost identical to the principle of "composition" used in C++ for creating hierarchical design. Hardware components can be modelled in C++, and to some extent, the mechanisms used are similar to those used in HDLs. Additionally C++ provides inheritance as a way to complement the composition mechanism and promotes design reuse.
Nonetheless, C++ does not support concurrency which is an essential aspect of systems modelling. Furthermore, timing and propagation delays cannot easily expressed in C++.
SystemC[2] is a relatively new modeling language based on C++ for system level design. It has been developed as standardized modeling language for system containing both hardware and software components.

---

[2]SystemC, Initiative Open. Initiative Open SystemC, (OSCI) http://www.systemc.org.

SystemC class library provides necessary constructs to model system architecture from reactive behaviour, scheduling policy and hardware-like timing. All of which are not available using C/C++ standalone languages.
There is multiple advantages of using SystemC, over a classic hardware description languages, such as VHDL and Verilog; flexibility, simplicity, simulation time velocity, and for most the portability.

4.1.1 SystemC implementation for Functional validation and verification

The SystemC approach consists of a progressive refinement of specifications. Therefore, a first initial implementation was done using an abstract high-level timed functional representation.
In this implementation, we used the proposed parallel structure discussed in section 3.
This modeling consists of high level SystemC modules (TLM) communicating with each other using channels, signals or even memory-blocks modules written in SystemC. Scheduling and timing were used but have not been explored for hardware-like purposes. Data types, used in this modelling, are strictly C++ data types.
Functional validation of our model SystemC is performed using a simulation phase (Figure 9). We simulate the behavior of a SystemC model by executing its processes in a pseudo concurrent way. The simulation stops when there is no eligible process and event notification. To manage the progress of simulation time, the SystemC simulator has a timed notifications schedule to be triggered. This schedule is a list of notifications of timed event that is sorted according to the time of such notification triggers.
Functional validation of the system was performed by comparing the results of the SystemC written structure with the results of OpenCV's software implementation, using 25 random images from the CMU image database.

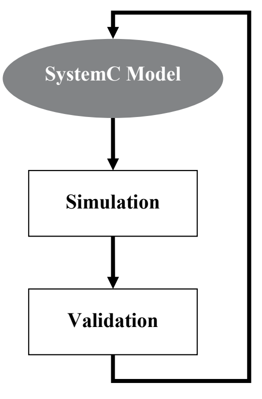


Fig. 9. SystemC functional validation flow

4.1.2 Modelling for Embedded implementation

While the previous SystemC modelling is very useful for functional validation, more optimization should be carried out in order to achieve a hardware implementation. Indeed, SystemC standard is a system-level modelling environment which allows the design of various abstraction levels of systems. The design cycle starts with an abstract high-level untimed or timed functional representation that is refined to a bus-cycle accurate and then an RTL (Register Transfer Level) hardware model. SystemC provides several data types, in

addition to those of C++. However these data types are mostly adapted for hardware specification.
Besides, SystemC hardware model can be synthesizable for various target technologies. Numerous behavioural synthesis tools are available on the market for SystemC (e.g. Synopsys Cocentric compiler, Mentor Catapult, SystemCrafter, and AutoESL). It should be noted, that for all those available tools, it is necessary to refine the initial simulation-like SystemC description in order to synthesize into hardware. The reason behind is the fact that SystemC language is a superset of the C++ designed for simulation. Therefore, a new improved and foremost a more refined "cycle accurate RTL model" version of the design implementation was created.
Our design is split into compilation units, each of which can be compiled separately. Alternatively, it is possible to use several tools for different parts of your design, or even using the partition in order to explore most of the possible parallelism and pipelining for more efficient hardware implementation. Eventually, the main block modules of the design were split into a group of small modules that work in parallel and/or in pipelining. For instance, the module Block1 contains three compilation units (modules): a "Decision" Module which contains the first stage's classifiers. This module is used for computation and decision on each sub-window. The second module is "Shift-and-Scale" used for shifting and scaling the window in order to obtain all subsequent locations. Finally, a "Memory-Ctrl" module manages the intermediate memory access.
As result, a SystemC model composed of 12 modules: three for Block1, two for Block2, three for Block3, one for the Integral image transformation, 3 for the memories.
Other major refinements were done: Divisions were simplified in order to be power of two divisions, dataflow model was further refined to a SystemC/C++ of combined finite state-machines and data paths, loops were exploited and timing/scheduling were taken into consideration. Note that in most cases, parallelism and pipelining were forced manually.
However, this level of description can vary depending on the needs of the high-level synthesis tool used for the hardware implementation. For example tools like Mentor Graphics CatapultC can propose and test different alternatives of parallel and pipeline implementations for high level of description C/SystemC. Other tools like SystemCrafter require manual coding of parallelism and pipeline operations.
On the other hand, not all the modules were heavily refined, for example the three memory modules were used in order simulate physical memories, which will never be synthesized no matter what the target platform is.

4.2 High level synthesis

SystemC hardware model can be synthesizable for various target technologies. However, no synthesizer is capable of producing efficient hardware from a SystemC program written for simulation. Automatic synthesis tool can produce fast and efficient hardware only if the entry code accommodates certain difficult requirements such as using hardware-like development methods. Therefore, the results of the synthesis design implementation depend heavily and the tool itself, and the different level of refinements done on the entry code. Figure 10 shows the two different kinds of refinements needed to achieve a successful fast implementation, using a high level description language. The first type of refinements is the one set by the tool itself. Without it, the tool is not capable of compiling the SystemC code to RTL level. Even so, those refinements don't lead directly to a good proven implementation. Another type of refinements should take place in order to optimize the size, the speed and sometimes (depending on the used tool) power consumption.

For our design, several refinements have been done on different modules depending on their initial speed and usability.
The SystemC scheduler uses the same behavior for software simulation as for hardware simulation. This works to our advantage since it gives the possibility of choosing which of the modules to be synthesized, while the rest works as SystemC test bench for the design.
Our synthesis phase was performed using an automatic tool, named SystemCrafter, which is a SystemC synthesis tool that targets Xilinx FPGAs.

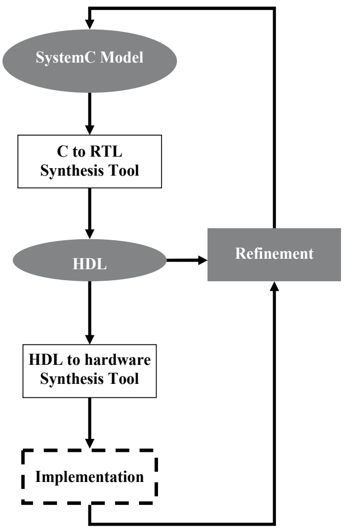


Fig. 10. SystemC to hardware implementation development flow

It should be noted that the used SystemC entry code can be described as VHDL-like synchronous and pipelined C-code (bit accurate): Most parallelism and pipelining within the design were made manually using different processes, threads, and state-machines. SystemC data types were used in order to minimize the implementation size. Loops were exploited, and timing as well as variables lengths were always a big factor.
Using the SystemCrafter, multiple VHDL components are generated and can be easily added or merged into/with other VHDL components (notably the FIFO's modules). As for the testbench set, the description was kept in high level abstraction SystemC for faster prototyping and simulation.
Basically, our implementation brings together four major components: the Integral Image module, the first stage decision module, the second stage decision module and Block 3 which runs sequentially the rest of the cascade stages. Each of these components was implemented separately in order to analyze their performances. In each case, multiple graphic simulations were carried out to verify that the output of both descriptions (SystemC's and VHDL's) are identical.
The reduced number of weak classifiers in the first two stages (three and nine respectively) allows us to store their settings in internal memory (LUT type). In the case of Block3, the number of weak classifiers is about 2500. Knowing that every weak classifier has 13 parameters (addresses rectangles, weight, threshold...) and each of these parameters is defined with an integer data type of size 24 bits. The size of memory needed to store these parameters is therefore: *2500* x *13* x *24* = *780 000* = *780Kbits*. It is therefore clear that the

storage of these parameters in internal LUT type memory can occupy a large number of logical blocks (slices) within the FPGA. We chose to store these parameters in the blocks RAM (BRAM). Indeed the use of SystemCrafter facilitates this task by using the type ram_block. This structure can be used in the same way as type "ram" plus it handles the storage and the control of the FPGA's BRAMS.

4.3 Performances

The Xilinx Virtex-4 XC4VL25 was selected as a target FPGA. The VHDL model was back annotated using the Xilinx ISE.

4.3.1 Non-optimized implementation

The synthesis results of the design implementation for each of the components are given on Table1.

	Logic Utilization	Used	Available	Utilization
Integral Image	Number of occupied Slices:	913	10752	8%
	Number of Slice Flip Flops:	300	21504	1%
	Number of 4 input LUTs:	1761	21504	8%
	Maximum frequency	**129 MHz**		
BLOCK1	Number of occupied Slices:	1281	10752	12%
	Number of Slice Flip Flops:	626	21504	3%
	Number of 4 input LUTs:	2360	21504	11%
	Maximum frequency	**47 MHz**		
BLOCK2	Number of occupied Slices:	3624	10752	34%
	Number of Slice Flip Flops:	801	21504	4%
	Number of 4 input LUTs:	7042	21504	33%
	Maximum frequency	**42 MHz**		
BLOCK3	Number of occupied Slices:	3178	10752	29%
	Number of Slice Flip Flops:	722	21504	3%
	Number of 4 input LUTs:	3014	21504	14%
	Maximum frequency	**43 MHz**		

Table 1. The synthesis results of the components implementations

The clock rate of the design did not exceed the rate of its slowest component. Therefore it is necessary to simulate and estimate the average speed of each block. Another big advantage of SystemC is the possibility of using C++/SystemC testbenches with VHDL models. And using simulation tools such as ModelSim, we can determine exactly the number of cycles needed to process a sub-window and thereafter the speed of each block. For instance Block1 can operate with a maximum frequency of 47 MHz and can process a sub-window in 42 clock cycles. Table 2 shows the average speed of each hardware Block using the CMU image database.

This actual implementation is capable of achieving only up to 11 frames per second on 320x240 images. Accelerating Block 1, Block 2 and Block3 is essential in order to achieve higher detection speed.

	Maximum Frequency (MHz)	Number of Cycle per Sub-windows	Average speed (frame per second)
Block1	47	42	11
Block2	42	112	11,3
Block3	43	192 to 2400	16

Table 2. speed of processing blocks before optimization

4.3.2 Optimized implementation

Analyzing the automatically generated VHDL code shows that despite all the refinement already done, the SystemCrafter synthesis tool still produces a much complex RTL code than essentially needed. Particularly, when using arrays in loops, the tool creates a register for each value, and then wired it into all possible outputs. Things get worse when trying to update all the array elements within one clock cycle. A scenario that occurs regularly in our design e.g. updating classifiers parameters. Simulation tests proved that these last manipulations can widely slowdown the design frequency. Therefore more refinements have been made for the "Decision" SystemC modules. For instance, the arrays updating were split between the clock cycles, in a way that no additional clock cycles are lost while updating a single array element per cycle.

The synthesis results for the new improve and more refined decision modules are shown in Table 3. The refinements made allow faster, lighter, and more efficient implementation for all 3 modules. Even more, the ModelSim simulation of our design shows that the refinements also allow achieving less cycles per decision (sub-windows processing) in all 3 blocks. Table 4 shows the new average speed of each VHDL Block using the CMU image database.

<table>
<tr><th></th><th>Logic Utilization</th><th>Used</th><th>Available</th><th>Utilization</th></tr>
<tr><td rowspan="4">BLOCK1</td><td>Number of occupied Slices:</td><td>713</td><td>10752</td><td>7%</td></tr>
<tr><td>Number of Slice Flip Flops:</td><td>293</td><td>21504</td><td>1%</td></tr>
<tr><td>Number of 4 input LUTs:</td><td>1091</td><td>21504</td><td>5%</td></tr>
<tr><td>Maximum frequency</td><td colspan="3">127 MHz</td></tr>
<tr><td rowspan="4">BLOCK2</td><td>Number of occupied Slices:</td><td>2582</td><td>10752</td><td>24%</td></tr>
<tr><td>Number of Slice Flip Flops:</td><td>411</td><td>21504</td><td>2%</td></tr>
<tr><td>Number of 4 input LUTs:</td><td>5082</td><td>21504</td><td>24%</td></tr>
<tr><td>Maximum frequency</td><td colspan="3">127 MHz</td></tr>
<tr><td rowspan="4">BLOCK3</td><td>Number of occupied Slices:</td><td>1703</td><td>10752</td><td>16%</td></tr>
<tr><td>Number of Slice Flip Flops:</td><td>405</td><td>21504</td><td>2%</td></tr>
<tr><td>Number of 4 input LUTs:</td><td>2616</td><td>21504</td><td>12%</td></tr>
<tr><td>Maximum frequency</td><td colspan="3">127 MHz</td></tr>
</table>

Table 3. The synthesis results for the new improved decision modules

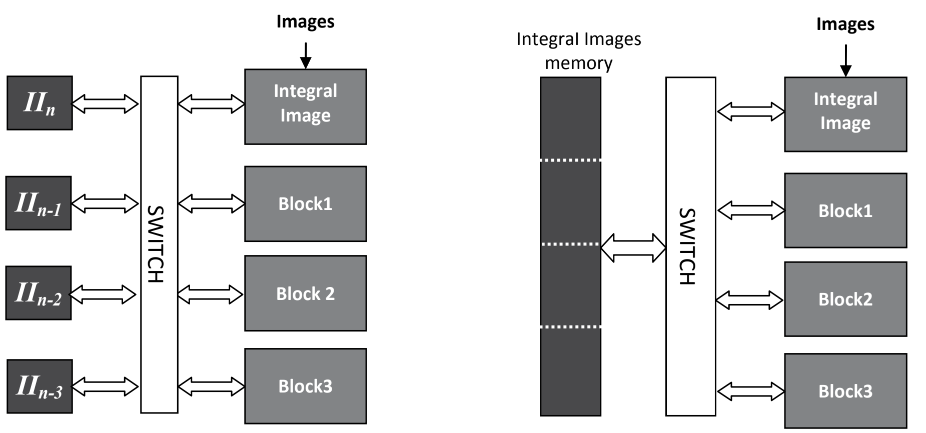
The FPGA can operate with a frequency of 127 MHz. Using the same logic as before, a system is as fast as its slowest blocks, therefore the new design can achieve up to 47 frames per second on 320x240 images.
The design can run on even faster pace, if more refinements and hardware considerations are taken. However, it should be noted that using different SystemC synthesis tools can yield different results. After all, the amount and effectiveness of the refinements depend largely on the tool itself.
Other optimizations can be done by replacing some of the auto-generated VHDL codes from the crafter with manually optimized ones.

	Maximum Frequency (MHz)	Number of Cycle per Sub-windows	Average speed (Image per second)
Block1	127	28	47
Block2	127	76	47,7
Block3	127	132 to 22890	50

Table 4. Speed of processing blocks after optimization

4.4 Architectural structure and memory blocks connectivity

The synthesis results from the previous paragraph helps showing the limitation in processing speed. However the performance of design also depends on the architectural structure of the implementation including the interconnectivity of the memory blocks and the number of physical memory used.
A first possible solution is to use a system with separated memory blocks (Figure 11.a.). At the end of each image cycle, a "switch" module is in charge of swapping memory blocks at a physical level. This solution can maximize processing performances, but it is too costly in terms of physical memory and resultant physical interconnections.
Reducing the number of physical memory can be explored in other solutions that integrate timesharing systems. Figure 11.b. shows a solution with a single physical memory for



a. Separated memory blocks b. Timesharing memory system

Fig. 11. Different architecture solution for memory's interconnection

Integral Images (same can be done for the intermediate memory blocks). The physical memory is logically divided into four memory banks that work in queues. The swapping of the memory access is also done by "Switch". But unlike the previous solution, where the swapping is made at a physical level, they are calculated at a logic level. This solution is optimal when the number of memory accesses is small. However in cases where memory accesses are the limiting factor of the system, this solution is less efficient than the using separated memories.

4.4.1 Intermediate memory

One of the drawbacks of the proposed parallel structure (given in section 4) is the use of additional intermediate memories (unnecessary in the software implementation). Logically, an inter-blocks memory unit is formed out of two memories working in ping-pong.

A stored address should hold the position of a particular sub-window and its scale; there is no need for two-dimensional positioning, since the Integral Image is created as a monodimensional table for a better RAM storage.

For a 320x240 image and a base sub-window size of 24x24, a word of 32 bits would be enough to store the concatenation of the position and the scale of each sub-window.

As for the capacity of the memories, a worst case scenario occurs when half of the possible sub-windows manage to pass through first block. That leads to around 2 x 53,000 (50% of the sub-windows) addresses to store. Using the same logic on the next block, the total number of addresses to store should not exceed the 168 000. Eventually, a combined memory capacity of less than 1 Mbytes is needed. The simulation of our SystemC model shows that, even when facing a case of consecutive positive decisions for a series of sub-windows, access onto those memories will not occur more than once every each 28 cycles (case of mem.1 and mem.2), or once each 76 cycles (case of mem.3 and mem.4). The access on these memories is regular since the writing and the reading are always done sequentially. Due to these facts, we propose a timesharing system (shown in Figure 12) using four memory banks, working as a FIFO block, with only one physical memory. In order to determine the exact characteristics of the needed memory, several testbenchs were created to compute the maximal bandwidth needed as well as the optimal FIFO queue size in worst

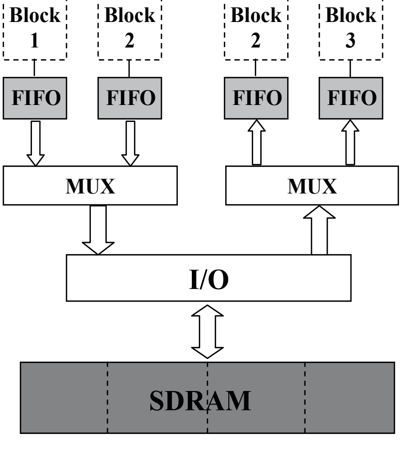


Fig. 12. Intermediate Memories structure

case scenario. The results of these simulation shows that a frequency of 17 MHz and a buffer size (each FIFO block) of 10 are enough for our configuration.
Typical hardware implementation of a 1 Mbytes SDRAM memory, running on a frequency of anything higher than 17 MHz, is enough to replace the four logical memories. Moreover, the required buffer size is very small. They FIFO are easily implemented with a limited number of block RAM (BRAM). We can use two BRAMS in dual-port mode or four BRAMS single port mode.

4.4.2 Integral image memory

The set of processing blocks (Integral Image, Block 1, Block2 and Block3) need to access 4 different Integral Images simultaneously. To achieve a detection of 47fps on 320x240 images, each of block must operate at their maximum frequency. In the worst case scenario the total bandwidth needed to access all Integral images is about 10,7 Gigabits/s.
However unlike the intermediate memory, the access to the integral images is never sequential or regular. The memory usage of SDRAM is not suited. In fact, the non-consecutive data transfer will drop dramatically the SDRAM bandwidth. Typically, for a latency of two cycles, the available bandwidth is divided by 3.The use of SRAMs appears to be more appropriate. For these reasons, we propose a solution with four SRAM memory units (Figure 13). Though, a solution with less memory units can be considered, the use of four minimizes the complexity of the switching module "SWITCH_II". At the end of each image cycle, a circular swapping logic is performed between memory units.

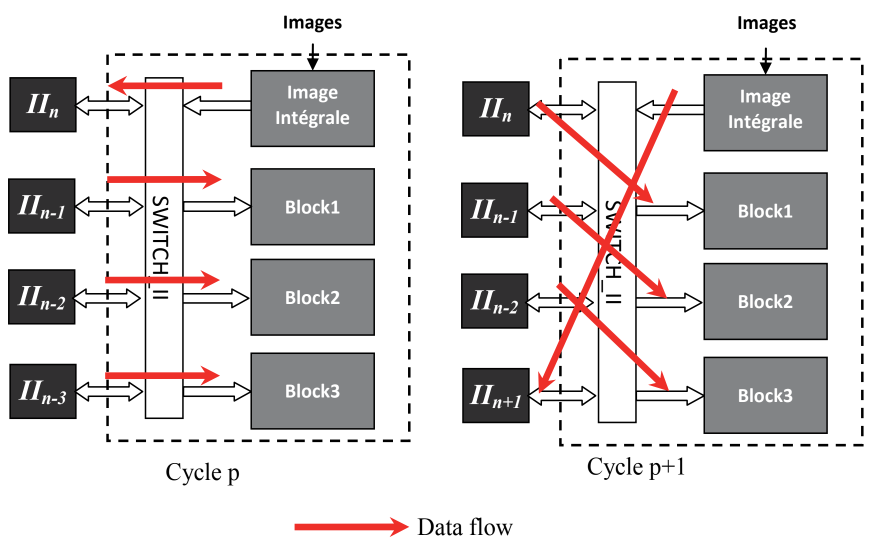


Fig. 13. Integral image interconnection

4.4.3 Final architecture structure

After establishing the interconnectivity of our architectural structure, we synthesize the whole system which includes Block 1 to 3, the Integral image module, and the switching modules. The results and the performances are shown in Table 5. The FPGA can operate at

a clock speed of 127 MHz. Figure 14 Shows the final proposed architecture which is capable of 47 images per second.
The simulation tests, used in section 4.1 for the functional validation of the SystemC code, were carried out on the VHDL code mixed with a high level test bench (the same SystemC test bench used for the SystemC validation model). The outputs of the VHDL code were compared to the outputs of the OpenCV's implementation. These tests prove that we were able to achieve the same detection results as in using the software provided by OpenCV.

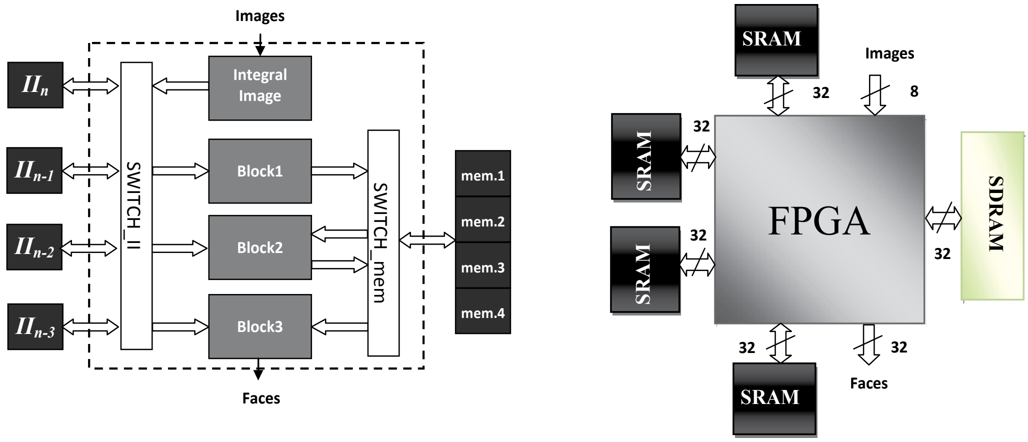


Fig. 14. Proposed architecture

Logic Utilization	Used	Available	Utilization
Number of occupied Slices:	6075	10752	57%
Number of Slice Flip Flops:	1729	21504	8%
Number of 4 input LUTs:	10711	21504	50%
Number of DSPs:	9	48	19%
Maximum frequency	**127 MHz**		

Table 5. The synthesis results of the refined implementation for the entire design

4.5 State-of-the-art comparision

We compare in this section, the performance of our embedded implementation with other known embedded implementations for the boosting based face detection in the literature. Comparisons are made in terms of speed (frame per second) and detection rates.The results of these comparisions are shown in Table 6.
One of the major challenges when trying to implement a real time embedded solution for this type of algorithms is the large number of features to be implemented in a cascade. In fact, the number of used features is usually betwen 2000 and 6000, depending on the training phase. Furthermore, the cascade implementation and the irregular nature of the stages of the cascade, make it extremely difficult to proposed a parallel structure capable of accelerating the detection without degrading the performances.

Implementation	**Image size**	**smallest sub-window size**	**Frames per second**	**Test database**	**Features used**	**Detection rate**
Wei & al. (Wei, Bing, & Chareonsak, 2004)	120x120	24x24	15	CMU	NA	50%
Yang et al.	320x240	24x24	13	P I DB[3]	140	75%
Theocharides (Theocharides, Vijaykrishnan, & Irwin, 2006)	NA	NA	52	P I DB	Less than 150	NA
Lai & al. (Lai, Marculescu, Savvides, & Chen, 2008)	320x240	20x20	40	P I DB	42	75%
Author's implementation	**320x240**	**24x24**	**47**	**CMU**	**2500**	**88%**

Table 6. Comparison of embedded implementation of Boosting based face detection

In the literature we can find a lot of attempts to accelerate the boosting based face detection. However the authors in these works have sought to reduce the overall computation burden, without taking into consideration the local burden of computation at each stage of classification. By consequence, they have abandoned the cascaded architecture in favor of a single complex stage of classification(e.g.only 42 features in the implementation of Lai & al.). Indeed, it is easier to exploit the parallelism of a single stage with several features than the parallelism of a complex cascade with very high data dependencies. However, this approach has two major drawbacks:

- Each sub-window must be evaluated by a large number of features (42 to 150) , when the average number of evaluated features per sub-windows in a cascade is generally less than 10.
- The evaluation of these features must be done in parallel to speed up the detection time. The amount of necessary resources for these computations is thereby increased, which explains the limited number of features used in the hardware implementations. And therefore the detection rates of these implementations are well under the ones set by the software implementations.

However, unlike the listed embedded implementations, our architecture is capable of supporting a large number of features. Indeed, we were able to implement the same full cascade (more than 2500 features) as the "default" one found in OpenCV. Hence the detection rates are the same as the software implementation. Our implementation can achieve up to 47 fps on the CMU image database, while processing about 106 000 sub-windows. The implementation proposed by Theocharides can achieve slightly higher of number of frames per second, but the authors did not provide sufficient details on important factors, such as images size and the smallest sub-window size. These configurations are essential in determining the speed of the detection (and the detection rates), for exemple taking changing the smallest sub-window size from 24x24 to 32x32 can divide the computation burden by 2 and therefore accelerate the detection by a factor of 2.

[3]proprietary image databases

5. Conclusion

This chapter can be considered as a continuation of previously published work (Khattab, Dubois & Miteran 2009) in which we proposed a new architecture for an embedded real-time face detector based on a fast and robust family of methods, initiated by Viola and Jones. The most notable differences between the 2 articles are: The implementation of the third processing block (Block3), the new architecture structure including memories types and interconnection, and finally a full system validation and tests.

First we analyse the sequential structure model which reveals to be irregular in time processing and in load partitioning. Then a new parallel structure model is introduced. This structure proves to be at least 3.4 times faster than the sequential, and provides regularity in time processing.

The design was validated using SystemC. Simulation and hardware synthesis were done, showing that such an algorithm can be fitted easily into a FPGA chip, while having the ability to achieve the state-of-the-art performances in both frame rate and accuracy.

The hardware target, used for the validation, is a FPGA based board, connected to the PC using an USB 2.0 Port. The use of SystemC description enables the design to be easily retargeted for different technologies. The implementation of our SystemC model onto a Xilinx Virtex-4 can achieve a theoretical 47 frames per second detection rate for 320x240 images. And Unlike the state-of-the-art embedded implementation, we were able to implement the whole cascade detector (with all the features) as the one use in the software implementation. This has led to achieve practically the same result in detection rates as in the software implementation.

On the other hand, we proved that SystemC description is not only interesting to explore and validate a complex architecture. It can also be very useful to detect bottlenecks in the dataflow and to accelerate the architecture by exploiting parallelism and pipelining. Then eventually, it can lead to an embedded implementation that achieves state-of-the-art performances, thanks to some synthesis tools. More importantly, it helps developing a flexible design that can be migrated to a wide variety of technologies.

However, experiments have shown that refinements made to the entry SystemC code add up to substantial reductions in size and total execution time. Even though, the extent and effectiveness of these optimizations is largely attributed to the SystemC synthesis tool itself and designer's hardware knowledge and experience. Therefore, one very intriguing perspective is the exploration of this design using other tools for comparison purposes.

Accelerating the first stage can lead directly to a whole system acceleration. In the future, our description could be used as a part of a more complex process integrated in a SoC. We are currently exploring the possibility of a hardware/software solution; by prototyping a platform based on a Wildcard. Recently, we had successful experiences, implementing a similar type of solutions in order to accelerate a "Fourier Descriptors for Object Recognition using SVM"(Smach, Miteran, Atri, Dubois, & Gauthier, 2007) and motion estimation for MPEG-4 coding(Dubois, Mattavelli, Pierrefeu, & Mitéran, 2005) . For example the Integral Image block as well as the first and second stages can be executed in hardware on the wildcard, while the rest can be implemented in software on a Dual core processor.

6. References

Dubois, J., Mattavelli, M., Pierrefeu, L., & Mitéran, J. (2005). Configurable Motion-Estimation Hardware Accelerator Module for the MPEG-4 Reference Hardware Description Platform. *Proceeding of IEEE International Conference on Image processing*.

Freund, Y., & Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. *In Computational Learning Theory: Eurocolt. Springer-Verlag*, 23–37.

Khattab, K., Dubois, J., & Miteran, J. (2009). Cascade Boosting Based Object Detection from High Level Description to Hardware Implementation. *EURASIP Journal of Embedded Systems, 2009 (Design and Architectures for Signal Image Processing)*

Lai, H. C., Marculescu, R., Savvides, M., & Chen, T. (2008). Communication-Aware Face Detection Using Noc Architecture. *in IEEE International Conference on Computer Vision Systems (ICVS)*.

Lienhart, R., Kuranov, A., & Pisarevsky, V. (2003). Empirical analysis of detection cascades of boosted classiers for rapid object detection. *In Proceedings of the 25th DAGM-Symposium*, 297-304.

Lienhart, R., & Maydt, J. (2002). An extended set of haar-like features for rapid object detection. *In Proceedings of the IEEE International Conference on Image Processing*, 900-903.

Moghaddam, B., & Pentland, A. (1997). Probabilistic visual learning for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 696-710.

Papageorgiou, C., Oren, M., & Poggio, T. (1998). A general framework for object detection. *In Proceedings of the IEEE Conference on Computer Vision*, 555-562.

Rowley, H. A., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*, 23-38.

Schneiderman, H., & Kanade, T. (2000). A statistical model for 3d object detection applied to faces and cars. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, *1*, 746-751.

Smach, F., Miteran, J., Atri, M., Dubois, J., & Gauthier, J. P. (2007). An FPGA-based accelerator for Fourier Descriptors computing for color object recognition using SVM. *Journal of Real-Time Image Processing (JRTIP), Springer*, *2*, 249-258.

Sung, K.-K., & Poggio, T. (1998). Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39-51.

Theocharides, T., Vijaykrishnan, N., & Irwin, M. J. (2006). A parallel architecture for hardware face detection in ISVLSI ’06. Washington, DC, USA : IEEE Computer Society, 2006, p. 452. *IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures* (p. 452). Washington, DC, USA.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, *1*, 511-518.

Viola, P., & Jones, M. (2002). Fast and robust classication using asymmetric adaboost and a detector cascade. *In Advances in Neural Information Processing Systems (NIPS), MIT Press*, 1311-1318.

Viola, P., Jones, M., & Snow, D. (2003). Detecting Pedestrians Using Patterns of Motion and Appearance. *IEEE International Conference on Computer Vision (ICCV)* (pp. 734-741).

Wei, Y., Bing, X., & Chareonsak, C. (2004). Fpga implementation of adaboost algorithm for detection of face biometrics. *IEEE InternationalWorkshop on Biomedical Circuits and Systems* (pp. S1/6 - 17-20).

Yang, M.-H. (2004). Recent advances in face detection. Technical report. *IEEE International Conference on Pattern Recognition Tutorial*.

Yang, M.-H., Kriegman, D. J., & Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 34-58.

Yang, M., Wu, Y., Crenshaw, J., Augustine, B., & Mareachen, R. (2006a). Face detection for automatic exposure control in handheld camera. *IEEE International Conference on Computer Vision Systems, ICVS*, *04*(07), 17.

Yang, M., Wu, Y., Crenshaw, J., Augustine, B., & Mareachen, R. (2006b). Face detection for automatic exposure control in handheld camera. *in IEEE International Conference on Computer Vision Systems (ICVS)*. Toronto, Canada.

Part 4

Methods of Face Characterization and Feature Detection

10

Face Discrimination Using the Orientation and Size Recognition Characteristics of the Spreading Associative Neural Network

Kiyomi Nakamura and Hironobu Takano
Toyama Prefectural University
Japan

1. Introduction

With the rapid progress of the information society, biometrics identification technology has been developed for security applications. Biometrics is person authentication using physical features such as the face, fingerprints, irises, etc. It is advantageous for psychological resistance to be minimized; verification using the face is uninvasive compared with fingerprints. Such security systems using remote monitoring are in demand in customs house and airports, etc.
Recently, face recognition by the on-line processing of facial images has been widely applied in various fields and evaluated the face recognition performance using large scale database (Phillips et al., 2007). The representative face recognition method is classified into two categories. The first is a feature-based approach which uses feature vectors created with complex Gabor wavelet coefficients at each node (Wiskott et al., 1997). The second is the holistic or pattern (template) matching approach. The well-known example for the latter is the approach using eigenfaces which are obtained from principal component analysis of a large number of either full face images (Turk & Pentland, 1991) or local feature images of the face, e.g. eyebrow, eye, nose, cheek, mouth, etc (Penev & Atick, 1996). In both approaches, the conventional nearest neighbor algorithm or neural network is used for face classification.
In personal authentication using facial images, it is a common problem to realize robust recognition independent of variations of illumination, orientation, size, pose, and expression, etc. The various methods of orientation recognition for facial image were proposed (Wong et al., 2001; Wu et al., 2006; Su, 2000). The orientation of facial image obtained by these methods is used for the orientation correction before the face (shape) recognition process. On the other hand, the face size is usually normalized by using the information of distance between eyes or face width. Recently, a rotation and size spreading associative neural network (RS-SAN net) was developed based on space and 3-D shape recognition systems in the brain (Nakamura & Miyamoto, 2001). Using RS-SAN net, a personal authentication method, which was not influenced by the orientation and size changes was proposed. The RS-SAN net correctly recognized face shape, orientation and size, regardless of the input orientation and size, once facial images were learned (Nakamura & Miyamoto, 2001; Nakamura & Takano, 2006). However, the face shape recognition performance of the RS-SAN net was slightly low compared with other face recognition methods.
In this chapter, we introduce a novel face recognition method using the characteristics of orientation and size recognition for decreasing false acceptance. Section 2 and 3 describe the outline of the rotation and size spreading associative neural network (RS-SAN net).

Recognition performances of the orientation, size and shape of faces are evaluated in Section 4. Section 5 details the novel face recognition method which introduces the unlearned face rejection with the orientation and size recognition characteristics. Section 6 concludes this chapter.

2. Rotation and size spreading associative neural network

2.1 Structure of the RS-SAN net

The RS-SAN net consists of orientation, size and shape recognition systems shown in Fig.1. The learning and recognition processes of the RS-SAN net are as follows.

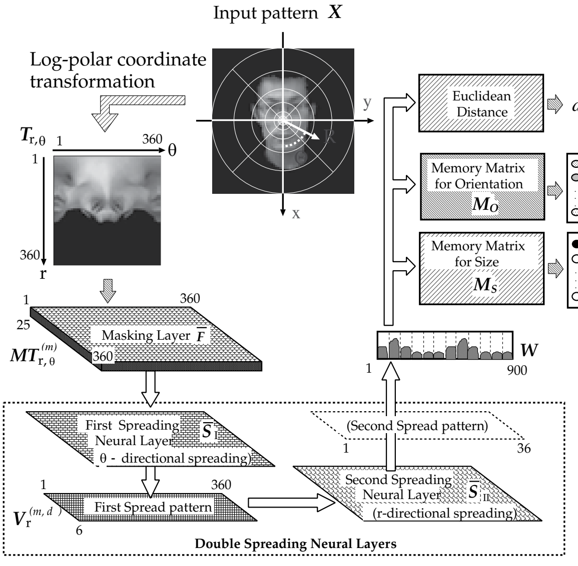


Fig. 1. Structure of the RS-SAN net.

1. The input face pattern $\boldsymbol{X}$ (480 $\times$ 480 pixels) is transformed into a transformed pattern $T_{r,\theta}$ (360 $\times$ 360 pixels) on log-polar coordinates.
2. The transformed pattern is passed through 25 masking layers to produce masked patterns.
3. The double spreading layers spread the 25 masked patterns by weighted orientation and size spreading functions to produce the double spread pattern (900 dimensions).
4. In learning, an orientation memory matrix $\boldsymbol{\mathcal{M}}_{\mathrm{O}}$ and size memory matrix $\boldsymbol{\mathcal{M}}_{\mathrm{S}}$ are obtained from the spread patterns $\boldsymbol{W}_{\mathrm{L}}^{(P)}$ calculated from learning patterns $\boldsymbol{X}^{(P)}$ ($P = 1, \cdots, P_{max}$) and orientation teaching signals $\boldsymbol{TO}^{(P)}$ and size teaching signals $\boldsymbol{TS}^{(P)}$, respectively. The learning was performed in 6 orientations ($0^\circ \sim 300^\circ$ in increments of 60°) $\times$ 6 sizes (same interval in logarithmic scale:1.00, 1.43, 2.04, 2.93, 4.19, 6.00) for respective faces. The spread pattern $\boldsymbol{W}_{\mathrm{L}}^{(P)}$ is also stored in face recognition system for shape recognition.
5. In recognition, the system recognizes the orientation and size at the same time by using the population vectors calculated from the outputs of 30 orientation and size recognition neurons (Georgopoulus et al., 1982). The outputs of orientation and size reecognition neurons are obtained by multiplying the spread patterns $\boldsymbol{W}_{\mathrm{R}}$ calculated from the input pattern $\boldsymbol{X}$ and orientation memory matrix $\boldsymbol{\mathcal{M}}_{\mathrm{O}}$ and size memory matrix $\boldsymbol{\mathcal{M}}_{\mathrm{S}}$, respectively.

The shape is discriminated by the Euclidean distance between the double spread patterns obtained in learning and recognition processes.

2.2 Feature vector calculation

The input face pattern $\boldsymbol{X}$ (480 × 480 pixels) is converted to a transformed pattern $T_{r,\theta}$ (360 × 360 pixels) on the log-polar coordinate system by Eq.(1).

$$T_{r,\theta} = \sum_{i=1}^{3}\sum_{j=1}^{3} I_{x_{ij},y_{ij}} \tag{1}$$

$$(x_{ij} = R\cos\Theta,\ y_{ij} = R\sin\Theta)$$

$$\begin{cases} R = 10^{LI\cdot r} + \left(10^{LI\cdot r} - 10^{LI(r-1)}\right) \times \frac{i}{3} \\ \Theta = (\theta - 1) + \frac{j}{3} \end{cases}$$

$$r,\ \theta = 1, 2, \cdots, 360,\ \ LI = \frac{\log(SR)}{360}$$

where $I_{x_{ij},y_{ij}}$ is the pixel value of the input facial image $\boldsymbol{X}$ at (x_{ij}, y_{ij}) on the Cartesian coordinates and $SR = 240$ is the sampling radius on the input facial image.

The transformed pattern $T_{r,\theta}$ is passed through 25 masking layers to produce 25 masked patterns $MT^{(m)}_{r,\theta}$, $(m = 1, 2, \cdots, 25)$. The masking layers are various spatial filters to extract characteristic features from the transformed image. The various masks $M^{(m)}$ are called spatial filters, and the masked images generated, $MT^{(m)}_{r,\theta}$, are shown in Fig.2. Here, m is a mask number $(m = 1, 2, \cdots, 25)$. A masked image $MT^{(m)}_{r,\theta}$ is calculated by the convolution of the transformed image $T_{r,\theta}$ and a mask $M^{(m)}$. If any pixel value of the masked image is negative $(MT^{(m)}_{r,\theta} < 0)$, the value is set to 0. The masks used in this study extract the edge components of the transformed image. For example, mask $M^{(2)}$ extracts the vertical edge component of a transformed image. The structure of the double spreading layers is shown in Fig.3. The orientation spreading weight $GO^{(d_\theta)}_{\theta}$ $(d_\theta = 1, 2, \cdots, 6)$ in the θ direction has a functional value like that of the Gaussian curve in Eq.(2), and is maximum (1.0) in orientation d_θ as shown in Eq.(3). Similarly, the size spreading weight $GS^{(d_r)}_{r}$ $(d_r = 1, 2, \cdots, 6)$ in the r direction has a functional value like that of the Gaussian curve that is maximum (1.0) in direction d_r as shown in Eqs.(4) and (5). The extent of spreading of the orientation and size information are decided by the spreading coefficients β_θ, β_r in Eqs.(2) and (4), becoming small when these spreading coefficients become larger.

$$F_{S\theta}(x) = \exp\{-\beta_\theta (x - 360n)^2\} \tag{2}$$

$$(-180 + 360n < x \leq 180 + 360n, n = 0, \pm 1, \ldots)$$

$$GO^{(d_\theta)}_{\theta} = F_{S\theta}\{60(d_\theta - 1) - (\theta - 1)\} \tag{3}$$

$$(d_\theta = 1, 2, \cdots, 6\ ,\ \theta = 1, 2, \cdots, 360)$$

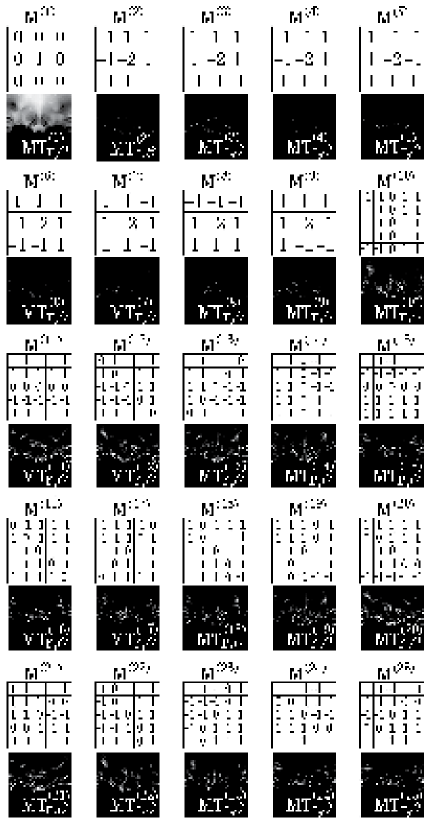


Fig. 2. Masks $M^{(m)}$ and masked images $MT_{r,\theta}^{(m)}$.

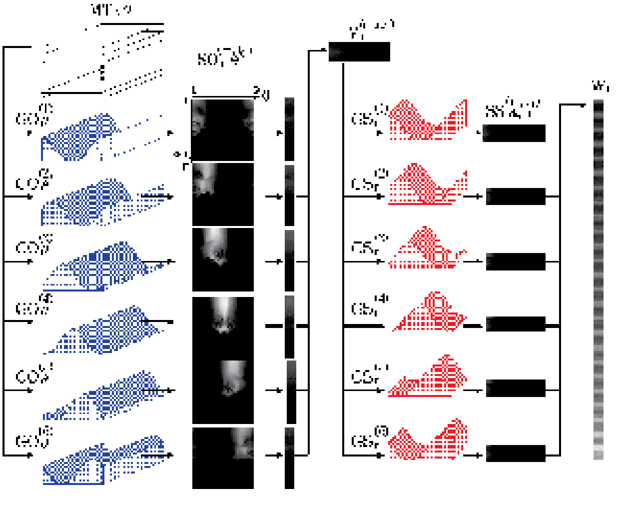


Fig. 3. Structure of the double spreading layers.

$$F_{Sr}(x) = \exp\{-\beta_r(x-360n)^2\} \tag{4}$$
$$(-180+360n < x \leq 180+360n, n = 0, \pm 1, \ldots)$$
$$GS_r^{(d_r)} = F_{Sr}\{60(d_r-1)-(r-1)\} \tag{5}$$
$$(d_r = 1, 2, \cdots, 6\ ,\ r = 1, 2, \cdots, 360)$$

We obtain the spread image $SO_{r,\theta}^{(m,d_\theta)}$ in the θ direction by multiplying the masked image $MT_{r,\theta}^{(m)}$ and $GO_\theta^{(d_\theta)}$ in Eq.(6), and the spread vector $V_r^{(m,d_\theta)}$ by the summation of $SO_{r,\theta}^{(m,d_\theta)}$ concerning θ by Eq.(7). Then, we obtain the spread image $SS_{d_\theta,r}^{(m,d_r)}$ in the r direction by multiplying $V_r^{(m,d_\theta)}$ and $GS_r^{(d_r)}$ in Eq.(8). Finally, the double spread vector $\boldsymbol{W}^{*(P)}$ is obtained by summation of $SS_{d_\theta,r}^{(m,d_r)}$ concerning r by Eq.(9). The dimension of $\boldsymbol{W}^{*(P)}$ is decided by d_r, d_θ and m. This reaches 900 dimensions by multiplying $6(d_\theta) \times 6(d_r) \times 25(m)$ in Eq.(10).

$$SO_{r,\theta}^{(m,d_\theta)} = MT_{r,\theta}^{(m)} \times GO_\theta^{(d_\theta)} \tag{6}$$
$$V_r^{(m,d_\theta)} = \sum_{\theta=1}^{360} SO_{r,\theta}^{(m,d_\theta)} \tag{7}$$
$$SS_{d_\theta,r}^{(m,d_r)} = V_r^{(m,d_\theta)} \times GS_r^{(d_r)} \tag{8}$$
$$W_i^* = \sum_{r=1}^{360} SS_{d_\theta,r}^{(m,d_r)} \tag{9}$$
$$(i = 36 \cdot (m-1) + 6 \cdot (d_\theta - 1) + d_r)$$
$$\boldsymbol{W}^{*(P)} = [W_1^*, \cdots, W_{900}^*]^T \tag{10}$$

To remove the bias of $\boldsymbol{W}^*$ which degrades the recognition performance, the normalized double spread vector $\boldsymbol{W}$ is obtained by Eqs.(11) and (12). As a feature vector of the face pattern $\boldsymbol{X}$, the normalized double spread pattern $\boldsymbol{W}$ is used for both learning (registration) and recollection (recognition).

$$||\boldsymbol{W}^*|| = \sqrt{\sum_{i=1}^{900} W_i^{*2}} \tag{11}$$
$$\boldsymbol{W} = \frac{\boldsymbol{W}^*}{||\boldsymbol{W}^*||} \tag{12}$$

2.3 Recognition neuron

2.3.1 Orientation recognition neuron

The orientation angle of a facial image is indicated by the orientation of a population vector ϕ_o. The ϕ_o is defined as an ensemble of vectors of the orientation recognition neurons $\boldsymbol{YO} = [YO_1, \cdots, YO_{30}]^T$ where each vector points to the neuron's optimally tuned orientation and has a length in proportion to the neuron's output (Georgopoulus et al., 1982). The arrangement of orientation neurons and the orientation population vector are shown in Fig.4. This assumes that the neurons in the parietal cortex (PG) of the brain recognize the axis orientation of an object by population coding, as seen in neurophysiological studies. Each orientation recognition neuron YO_i has a respective representative orientation ψ_i that characterizes the best orientation for the optimal response in Eq.(13). The population

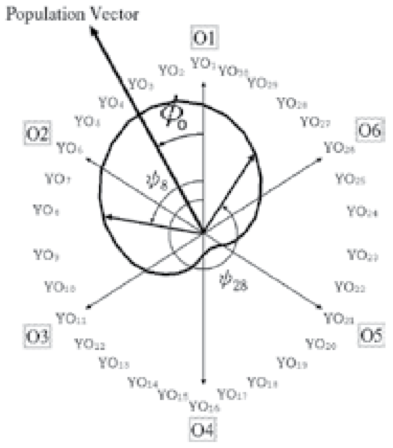


Fig. 4. Arrangement of orientation recognition neurons and population vector.

vector orientation ϕ_o is calculated by the vectorial summation of 30 orientation neurons $(YO_1, \cdots, YO_{30})$ by Eq.(14).

$$\psi_i = \frac{2\pi}{30} \times (i-1) \qquad (i = 1, 2, \cdots, 30) \tag{13}$$

$$\phi_o = \tan^{-1}\left(\sum_{i=1}^{30} YO_i \sin\psi_i \Big/ \sum_{i=1}^{30} YO_i \cos\psi_i\right) \tag{14}$$

2.3.2 Size recognition neuron

The size of a facial image is also indicated as a direction of a population vector ϕ_S. The arrangement of size recognition neurons and size population vector are shown in Fig.5. This assumes that the neurons in the PG of the brain recognize the size of an object by population coding. Each size recognition neuron has a respective representative size that characterizes the best size for the optimal response. For example, the size neuron YS_{10} has a optimal response for the size of the facial image $S4$ (2.93) and the neurons around YS_{10} also have moderate responses to the same size. The population vector size ϕ_S is calculated by the vectorial summation of 30 size recognition neurons $(YS_1, \cdots, YS_{30})$ by Eq.(15). Between sizes $S1$ and $S6$, there are undefined size ranges, because the sizes $S1$ to $S6$ are not continuous. If the size population vector indicates an undefined range, the size of the facial image will not be obtained in Eq.(16).

$$\phi_s = \tan^{-1}\left(\sum_{i=1}^{30} YS_i \sin\psi_i \Big/ \sum_{i=1}^{30} YS_i \cos\psi_i\right) \tag{15}$$

$$\eta = \begin{cases} 10^{\frac{\log 6}{\pi}\cdot\phi_s} & (-\frac{\pi}{5} \le \phi_s \le \frac{6}{5}\pi) \\ undefine & (-\frac{4}{5}\pi \le \phi_s \le -\frac{\pi}{5}) \end{cases} \tag{16}$$

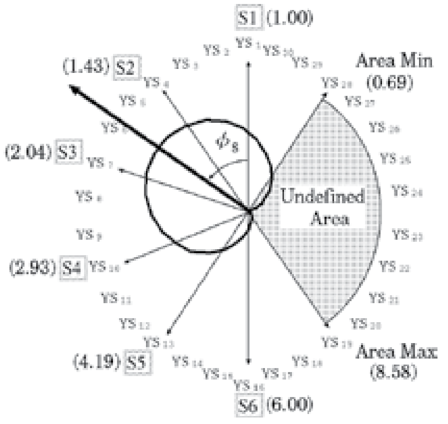


Fig. 5. Arrangement of size recognition neurons and population vector.

3. Learning (registration) and recognition

In the learning process of orientation and size, the RS-SAN net uses generalized inverse learning (Nakano, 1990; Amari, 1978), developed in 6 orientations (0° $\sim$ 300° in increments of 60°) $\times$ 6 sizes (same interval in logarithmic scale:1.00, 1.43, 2.04, 2.93, 4.19, 6.00) for respective faces. In the typical case of learning 10 human faces, the 360 patterns (=10 faces $\times$ 6 orientations $\times$ 6 sizes) normalized double spread patterns $\boldsymbol{W}_{\mathrm{L}}^{(P)}$ ($P = 1, \cdots, 360$) are memorized. In the learning process of the face shape, the specified normalized double spread patterns $\boldsymbol{W}_{\mathrm{L}}^{(P)}$ corresponding to specified orientation and size (typically, orientation = 0° and size = 6.0) of the respective faces are registered.

3.1 Teaching signal

3.1.1 Orientation recognition neuron

The teaching signal for orientation recognition $\boldsymbol{TO}^{(P)}$ is shown in Fig.6. There were six training signals $\boldsymbol{KO}^{(d_o)}$ corresponding to the six orientations d_o to be memorized. The desired outputs of the orientation recognition neurons were broadly tuned to the orientation of the facial image and adjusted to the function in Eq.(17). The desired outputs of orientation

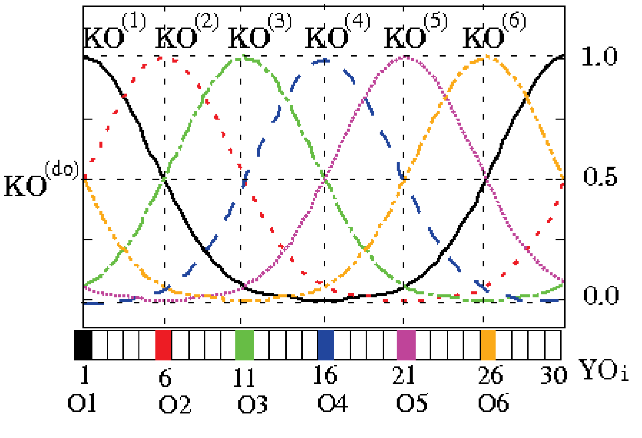


Fig. 6. Teaching signal for orientation recognition.

recognition neurons $\boldsymbol{TO}^{(P)}$ in Eq.(20) are fitted to $\boldsymbol{KO}^{(d_o)}$ which is the Gaussian curve function defined by Eqs.(17) and (18). For example, when the RS-SAN net memorizes orientation $O4(180°)$, the maximum value of the Gaussian curve function in Eq.(17) fits the orientation recognition neuron YO_{16}. This training signal is the same irrespective of the size and shape of the facial image when the orientation of other learned facial image is the same.

$$F_{TO}(x) = \exp\left\{-\alpha_o(x-30n)^2\right\} \tag{17}$$

$$(-15+30n < x \leq 15+30n, n = 0, \pm 1, \ldots)$$

$$KO_i^{(d_o)} = F_{TO}\{5(d_o-1)-(i-1)\} \tag{18}$$

$$(d_o = 1, 2, \cdots, 6\,,\ i = 1, 2, \cdots, 30)$$

$$\boldsymbol{KO}^{(d_o)} = [KO_1^{(d_o)}, KO_2^{(d_o)}, \cdots, KO_{30}^{(d_o)}]^T \tag{19}$$

$$\boldsymbol{TO}^{(P)} = \boldsymbol{KO}^{(d_o)} \tag{20}$$

Here, P $(= 1, \cdots, 360)$ is the training pattern number, d_o $(= 1, \cdots, 6)$ is the training orientation of the P-th training pattern, i $(= 1, \cdots, 30)$ is the number of the orientation recognition neuron, and α_o is the coefficient that defines the tuning width of the teaching signal for orientation recognition neurons.

3.1.2 Size recognition neuron

The teaching signal for orientation recognition $\boldsymbol{TS}^{(P)}$ is shown in Fig.7. There were six size training signals $\boldsymbol{KS}^{(d_s)}$ corresponding to the six sizes d_s to be memorized. The desired outputs of the size memory neurons were broadly tuned to the size of the facial image and adjusted to the function in Eq.(21). The desired outputs of the size recognition neurons $\boldsymbol{TS}^{(P)}$ in Eq.(24) were fitted to $\boldsymbol{KS}^{(d_s)}$, which is the Gaussian curve function defined by Eqs.(21) and (22). For example, when RS-SAN net memorizes size $S2$ (1.43), the maximum value of the Gaussian curve function in Eq.(21) fits the size memory neuron YS_4. This training signal is the same irrespective of the orientation and shape of the facial image when the size of other learned facial image is the same.

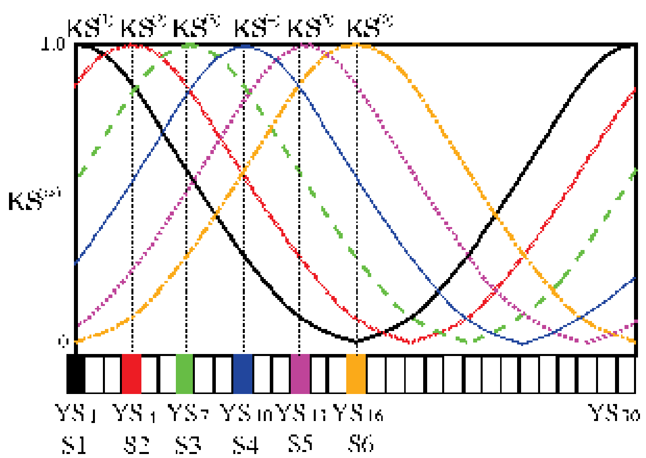


Fig. 7. Teaching signal for size recognition.

$$F_{TS}(x) = \exp\left\{-\alpha_s(x-30n)^2\right\} \tag{21}$$
$$(-15+30n < x \leq 15+30n, n = 0, \pm 1, \ldots)$$
$$KS_j^{(d_s)} = F_{TS}\left\{3(d_s-1)-(j-1)\right\} \tag{22}$$
$$(d_s = 1, 2, \cdots, 6\ ,\ j = 1, 2, \cdots, 30)$$
$$\boldsymbol{KS}^{(d_s)} = [KS_1^{(d_s)}, KS_2^{(d_s)}, \cdots, KS_{30}^{(d_s)}]^T \tag{23}$$
$$\boldsymbol{TS}^{(P)} = \boldsymbol{KS}^{(d_s)} \tag{24}$$

Here, P $(= 1 \sim 360)$ is the training pattern number, d_s $(= 1 \sim 6)$ is the learning size number of the P-th training pattern, j is the size memory neuron number, and α_s is the coefficient that decides the tuning width of teaching signal for size recognition neurons in Eq.(21).

3.2 Learning (registration) process

The RS-SAN net uses generalized inverse learning for orientation and size recognition. The double spread pattern $\boldsymbol{W}_{\mathrm{L}}^{(P)}$ is obtained from the P-th training input pattern in the double spreading layers. The orientation memory matrix $\boldsymbol{\mathcal{M}}_{\mathrm{O}}$ is obtained by associating $\boldsymbol{W}_{\mathrm{L}}^{(P)}$ with the desired outputs of orientation recognition neurons $\boldsymbol{TO}^{(P)}$ by Eq.(27). The size memory matrix $\boldsymbol{\mathcal{M}}_{\mathrm{S}}$ is obtained by associating $\boldsymbol{W}_{\mathrm{L}}^{(P)}$ with the desired outputs of size recognition neurons $\boldsymbol{TS}^{(P)}$ $(P = 1, \cdots, 360)$ by Eq.(28). For the shape learning of the faces, the double spread patterns $\boldsymbol{W}_{\mathrm{L}}^{(P)}$ of specified orientation (= 0°) and size (= 6.0) for the respective faces are registered in the face recognition system.

$$\boldsymbol{\mathcal{X}} = [\boldsymbol{W}_{\mathrm{L}}^{(1)}, \boldsymbol{W}_{\mathrm{L}}^{(2)}, \cdots, \boldsymbol{W}_{\mathrm{L}}^{(360)}] \tag{25}$$
$$\boldsymbol{\mathcal{X}}^{+} = (\boldsymbol{\mathcal{X}}^T\boldsymbol{\mathcal{X}})^{-1}\boldsymbol{\mathcal{X}}^T \tag{26}$$
$$\boldsymbol{\mathcal{M}}_{\mathrm{O}} = \boldsymbol{\mathcal{TO}}\boldsymbol{\mathcal{X}}^{+} \tag{27}$$
$$\boldsymbol{\mathcal{TO}} = [\boldsymbol{TO}^{(1)}, \boldsymbol{TO}^{(2)}, \cdots, \boldsymbol{TO}^{(360)}]$$
$$\boldsymbol{\mathcal{M}}_{\mathrm{S}} = \boldsymbol{\mathcal{TS}}\boldsymbol{\mathcal{X}}^{+} \tag{28}$$
$$\boldsymbol{\mathcal{TS}} = [\boldsymbol{TS}^{(1)}, \boldsymbol{TS}^{(2)}, \cdots, \boldsymbol{TS}^{(360)}]$$

3.3 Recognition process

In the recognition process, the system simultaneously recognizes the orientation and size of the facial image. First, the double spread pattern $\boldsymbol{W}_{\mathrm{R}}$ used for recognition is generated with the input facial image. For face orientation recognition, the orientation memory matrix $\boldsymbol{\mathcal{M}}_{\mathrm{O}}$ is multiplied by $\boldsymbol{W}_{\mathrm{R}}$ in Eq.(29), and the output of orientation recognition neurons $\boldsymbol{YO}$ is obtained. For face size recognition, the size memory matrix $\boldsymbol{\mathcal{M}}_{\mathrm{S}}$ is multiplied by $\boldsymbol{W}_{\mathrm{R}}$ in Eq.(30), and the output of size recognition neurons $\boldsymbol{YS}$ is obtained.

$$\boldsymbol{YO} = \boldsymbol{\mathcal{M}}_{\mathrm{O}}\boldsymbol{W}_{\mathrm{R}} \tag{29}$$
$$\boldsymbol{YS} = \boldsymbol{\mathcal{M}}_{\mathrm{S}}\boldsymbol{W}_{\mathrm{R}} \tag{30}$$

The orientation is recognized by the population vector calculated from the outputs of 30 orientation recognition neurons. The size is also recognized by the population vector calculated from the outputs of 30 size recognition neurons.

The shape is discriminated by the Euclidean distance between the double spread patterns obtained in learning and recognition processes. The value of Euclidean distance (d) in Eq.(31) has the range of $0 \leq d \leq 2$, because the norm of spread pattern is normalized as 1. When it has the minimum value of "0", resemblance is the highest. The double spread pattern $\boldsymbol{W}'_{\mathrm{R}}$ used for the shape recognition is generated by correcting the orientation and size of the input facial image to the pre-determined specified ones (typically, orientation = 0° and size = 6.0). The orientation and size of the corrected facial image correspond to those of the learned face. The orientation and size correction prevents the deterioration of the shape recognition performance.

$$d = ||\boldsymbol{W}_{\mathrm{L}} - \boldsymbol{W}'_{\mathrm{R}}|| \tag{31}$$

4. Face recognition experiment

The characteristics of orientation, size and shape recognition for learned and unlearned faces were investigated with face database collected at The University of Essex (Spacek, 2008). The 70 facial images of 35 subjects (2 images for each subject) were used for recognition experiments. In preprocessing, the background and clothing areas were excluded. The image size and format were converted to 480×480 [pixels] and gray scaled (256 steps), respectively. The facial images in the learning and recognition tests were at size 6.0 and orientation 0°. For the convenience sake, one facial image obtained from 35 subjects was used for training in 6 x 6 orientations and sizes. The orientation and size recognition tests were examined using 6 facial images (another learned facial image and five unlearned faces). We tried 35 sets of recognition tests by changing the learning and recognition facial images one by one. Recognition results were thus obtained for 210 trials consisting of 35 trials for learned faces and 175 for unlearned faces. In shape recognition test, 10 facial images (another learned facial image and 9 unlearned facial images) among 35 subjects were recollected for each learned face. Thus, 350 recognition trials consisting of 35 trials for learned faces and 315 trials for unlearned faces were examined. The shape recognition was evaluated using the false rejection rate (FRR) and false acceptance rate (FAR). When the output of Euclidean distance calculated for learned face is higher than the decision threshold, we considered that the registered face was erroneously rejected and calculated the false rejection rate by counting the trials of false rejection. On the other hand, when the output of Euclidean distance calculated for unlearned face was lower than the decision threshold, we considered that the imposters were accepted incorrectly. We calculated the false acceptance rate by counting the trials of false acceptance.

4.1 Orientation recognition performance

The orientation recognition result for learned and unlearned faces was shown in Fig.8. The horizontal axis is the input face number, and the vertical axis is the recognized orientation angle. The average $\pm$ standard deviation of recognized orientation for learned and unlearned faces were $0.47 \pm 1.89[^\circ]$ and $3.74 \pm 43.69[^\circ]$, respectively. As shown in Fig.8, the recognized orientation of learned faces distributed around 0 degree; however, the recognized orientation of unlearned faces was heavily dispersed (SD was very large). The histogram of absolute error of recognized orientation angle for learned and unlearned faces was shown in Fig.9. The horizontal axis is the absolute error of recognized orientation angle, and the vertical axis is the percentage of facial image included in each bin. The white and black bars show the distribution of the absolute error of recognized orientation for learned and unlearned faces, respectively. The absolute error of recognized orientation for learned faces was less than 4 degrees; however, the absolute error of recognized orientation for most of unlearned faces distributed more than 5 degrees.

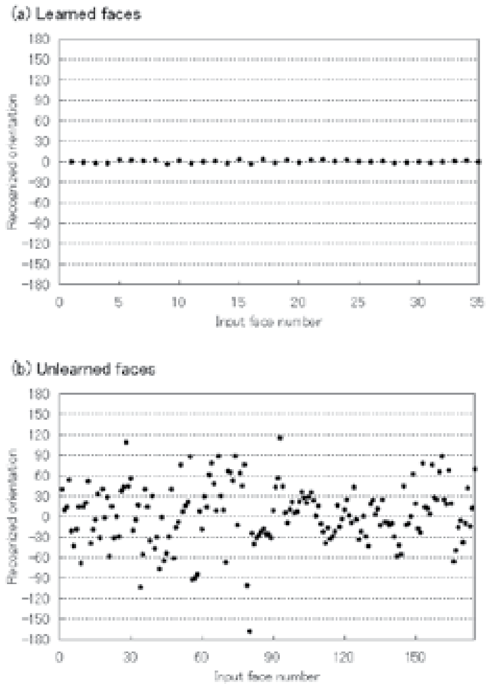


Fig. 8. Orientation recognition result for (a) learned and (b) unlearned faces.

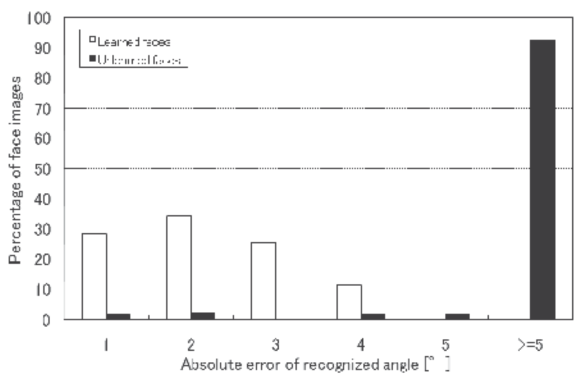


Fig. 9. Histogram of absolute error of recognized orientations for learned and unlearned faces.

4.2 Size recognition performance

The size recognition result for learned and unlearned faces was shown in Fig.10. The horizontal axis is the input face number, and the vertical axis is the recognized size. The average $\pm$ standard deviation of recognized size for learned and unlearned faces were 6.03 ± 0.26 and 5.51 ± 2.78, respectively. As shown in Fig.10, the recognized size of learned faces distributed around 6, which means the registered face size; however, the recognized size of unlearned faces was heavily dispersed (SD was very large). The histogram of absolute error of recognized size for learned and unlearned faces was shown in Fig.11. The horizontal

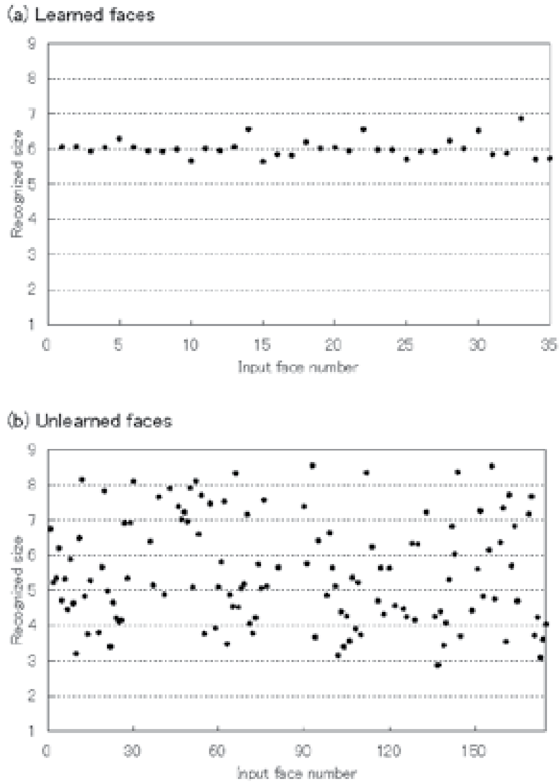


Fig. 10. Size recognition result for (a) learned and (b) unlearned faces.

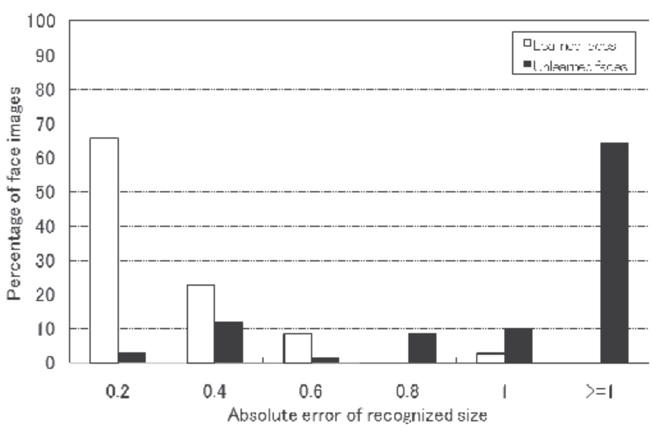


Fig. 11. Histogram of absolute error of recognized size for learned and unlearned faces.

axis is the absolute error of recognized size, and the vertical axis is the percentage of facial image included in each bin. The white and black bars show the distribution of the absolute error of recognized size for learned and unlearned faces, respectively. The absolute error of recognized size for learned faces was less than 1; however, 64 % absolute error of recognized size for unlearned faces distributed more than 1.

4.3 Shape recognition performance

Shape recognition performance was evaluated using equal error rate (EER) determined by finding the point where false acceptance rate intersects the false rejection rate. The result of shape recognition is shown in Fig.12. The horizontal axis is the decision threshold for discriminating between registered faces and imposters. The vertical axis is the FRR and FAR. Circle and solid line show the FAR. Square and dashed line show the FRR. The equal error rate was 2.86 % when the decision threshold of Euclidean distance was 0.13. At and below the decision threshold criterion of 0.06, the FAR was 0 %, even if the FRR was 34 %.

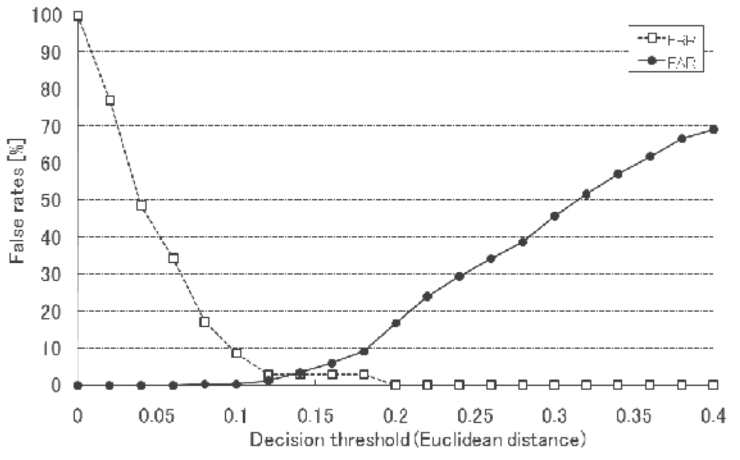


Fig. 12. False acceptance and rejection rates obtained with Euclidean distance.

5. Unlearned face rejection with recognized orientation and size

The orientation and size recognition performances indicated the RS-SAN net had fairly good orientation and size recognition characteristics for learned faces. On the other hand, the orientation angle and size of unlearned faces were hardly recognized because the distributions of recognized orientation angle and size were widely dispersive. Thus, the RS-SAN net can recognize simultaneously both orientation and size of only learned faces. Using the difference of orientation and size recognition characteristics between learned and unlearned faces, the unlearned face would be removed before face discrimination with Euclidean distance. The flowchart of new face (shape) recognition processes is shown in Fig.13. Before shape recognition using Euclidean distance calculated with double spread patterns, the unregistered faces are rejected using the averages and standard deviations of recognized orientation angle (θ_{av}, σ_o) and size (S_{av}, σ_s) for learned faces obtained by the RS-SAN net. The input face is determined as imposter if the recognized orientation is out of $\theta_{av} \pm 3\sigma_o$. If the recognized size is greater than $S_{av} + 3.2\sigma_s$ or less than $S_{av} - 3.2\sigma_s$, the input face is also rejected as imposter. Note that all of learned faces are not rejected with the orientation and size discrimination because the recognized orientation and size for learned faces were within $\theta_{av} \pm 3\sigma_o$ and $S_{av} \pm 3.2\sigma_s$.

The shape recognition performance obtained by new recognition method was shown in Fig.14. The horizontal axis is the decision threshold for discriminating between registered faces and

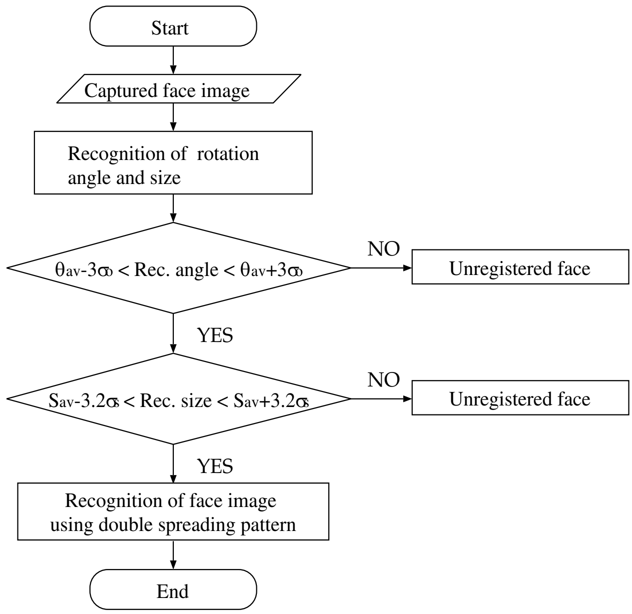


Fig. 13. Flowchart of new shape recognition process using the characteristics of orientation and size recognition.

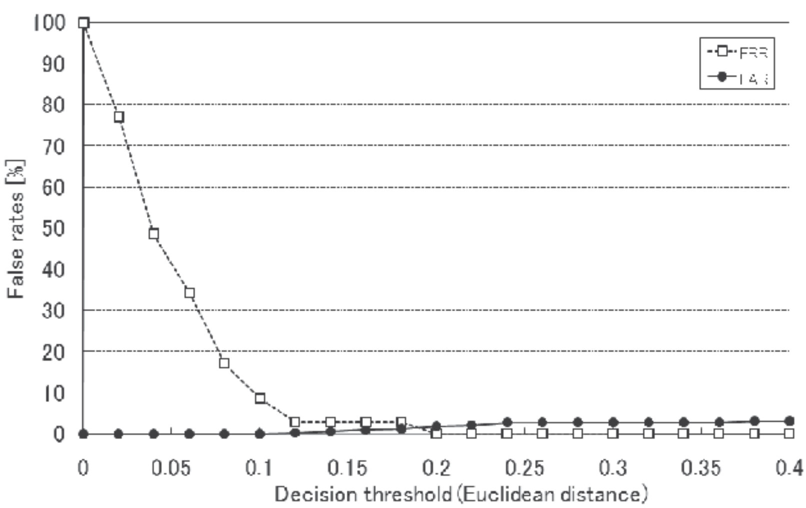


Fig. 14. False acceptance and rejection rates obtained by new face discrimination method with the characteristics of orientation and size recognition.

imposters. The vertical axis is the FRR and FAR. Circle and solid line show the FAR. Square and dashed line show the FRR. The facial images used for learning and recognition are the same as Section 4. This result indicated the FAR drastically decreased. The equal error rate was 1.56 % at the decision threshold of 0.19. When the false acceptance rate was 0 %, the false rejection rate decreased from 34 % to 8.6 %. The criteria of imposter rejection are empirically determined using the experimental results of the orientation and size recognition. To raise reliability of these decision criteria, the orientation and size recognition characteristics of the learned and unlearned faces would be investigated with large-scale database; however, the experimental result indicates that the unregistered face rejection by the recognized orientation and size is very effective to improve the shape recognition performance.

6. Conclusions

In this chapter, we showed the recognition characteristics of the RS-SAN net for the learned and unlearned faces. The RS-SAN net can recognize both orientation angle and size for learned faces. On the other hand, both orientation and size for unlearned faces were not obtained by the RS-SAN net because the recognized orientation and size were heavily dispersed from the orientation and size of input face. In the shape recognition, the equal error rate was 2.86 % at decision threshold of 0.13.

The RS-SAN net has the unique characteristics of the orientation and size recognition. The orientation and size of only learned face were recognized correctly. However, the recognized orientation and size of unlearned faces were heavily scattered. By introducing the unlearned face discrimination with the recognized orientation and size, new shape recognition method was developed. The experimental result of new shape recognition method showed that the false acceptance rate decreased drastically, even in very high decision threshold. The false acceptance rates were almost constant (about $2 \sim 3$ %) across the decision threshold ranging from 0.2 to 0.4. The imposter rejection method using recognized orientation and size provided the effective improvement of the face recognition performance. The equal error rate decreased to 1.56 % at the decision threshold of 0.19, and the false rejection rate also decreased to 8.6 % at a false acceptance rate of 0 %. Though the scale of the face database used in the present study was small, the face recognition performance in the present study was almost comparable with those reported in FRVT 2006 using large scale database (Phillips et al., 2007). The characteristics of recognition algorithm in the present stuty is that the false acceptance rates can be reduced dramatically even in the condition of very high decision threshold. This characteristics will not be concerned with the scale of the database.

In future studies, we will automatically detect the facial area using skin color information or appearance-based method, e.g. Haar-like features (Viola & Jones, 1996), and correct the facial center using positional information of both eyes. In addition, the recognition experiment will be examined with many more samples of facial images to obtain the decision criteria of the imposter rejection combined with the orientation and size recognition.

7. References

Phillips, P. J.; Scruggs, W. T.; O'Toole, A. J.; Flynn, P. J.; Bowyer, K. W.; Schott, C. L. & Sharpe, M. (2007). FRVT 2006 and ISCE 2006 large-scale results, 15.03.2011, Available from http://www.frvt.org/.

Wiskott, L.; Fellous, J.; Kruger, N. & Malsburg, C. von der (1997). Face recognition by elastic bunch graph matching. *PAMI*, Vol.19, No.7, pp.775-779.

Turk, M. & Pentland, A. (1991). Eigenfaces for recognition. *J. Cognitive Neurosci.*, Vol. 3, No. 1, pp. 71-86.

Penev, P. S. & Atick, J. J. (1996). Local feature analysis: A general statistical theory for object representation. *Neural Systems*, Vol. 7, No. 3, pp. 477-500.

Wong, K.-W.; Lam, K.-M. & Siu, W.-C. (2001). An efficient algorithm for human face detection and facial feature extraction under different conditions. *Pattern Recognition*, Vol. 34, No. 10, pp. 1993-2004.

Wu, S.; Jiang, L.; Xie, S. & Yeo, A. C. B. (2006). A robust method for detecting facial orientation in infrared images. *Pattern Recognition*, Vol. 39, No. 2, pp. 303-309.

Su, C.-L. (2000). Face recognition by using feature orientation and feature geometry matching. *J. Intell. Robotic Syst.*, Vol. 28, No. 1, pp. 159-169.

Nakamura, K. & Miyamoto, S. (2001). Rotation, size and shape recognition by a spreading associative neural network. *IEICE Trans. Inf. & Syst.*, Vol. E-84-D, No. 8, pp. 1075-1084.

Nakamura, K. & Takano, H. (2006). Rotation and size independent face recognition by the spreading associative neural network. *Proc. IEEE World Congress on Comp. Intell. (WCCI2006)*, pp. 8213-8219.

Nakano, K. (1990). *An Introduction to Neurocomputing*, Corona Publishing, Tokyo.

Amari, S. (1978). *A Mathematical Principle of Neural Networks*, Sangyo Publishing, Tokyo.

Georgopoulus, A. P.; Kalaska, J. F.; Caminiti, R. & Massey, J. T. (1982). On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *J. Neurosci.*, Vol. 2, No. 11, pp. 1527-1537.

Spacek, L. (2008). Face Recognition Data, 15.03.2011, Available from http://cswww.essex.ac.uk/mv/allfaces/index.html.

Viola, P. & Jones, M. J. (2004). Robust real-time face detection. *Int. J. Computer Vision*, Vol. 57, No. 2, pp. 137-154.

11

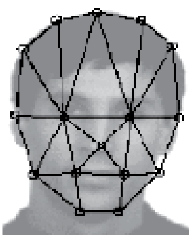
The Methodology for Facial Features Detection

Jacek Naruniec
Warsaw University of Technology, Institute of Radioelectronics
Poland

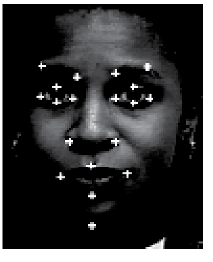
1. Introduction

Face detection is an important preprocessing task in biometric systems based on facial images. The result of the detection derives the localisation parameters and it could be required in various forms (Figure 1), for instance:

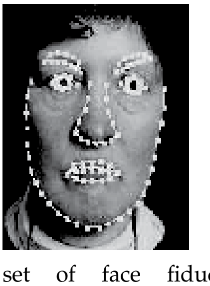
- a rectangle covering the central part of face,
- a larger rectangle including forehead and chin,
- irregular mask of the face area,



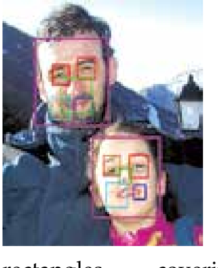
face graph Wiskott et al. (1997)



set of the face fiducial points Vukadinovic & Pantic (2005)



set of face fiducial points placed on face parts contours Cootes et al. (1998)



rectangles covering face parts and the face itself Erukhimov & Lee (2008)

Fig. 1. Different methods of representing face fiducial points and face parts.

- eyes centers,
- multiple face fiducial points,
- contours of the face parts,
- a set of rectangles covering individual parts of the face.

While from human point of view the area parameters are more convincing, for face recognition system, fiducial points are more important since they allow to perform facial image normalization – the crucial task before facial features extraction and face matching.
Facial features localization algorithms are commonly divided into four groups (Ben Jemaa & Khanfir (2009); Celiktutan et al. (2008); Naruniec (2010)):

- appearance-based,
- geometry-based,
- knowledge-based,
- 3D vision-based.

This chapter aims at defining more general scheme for facial features localization, since all of the defined groups have common methodology. Moreover most of the efficient schemes doesn't rely on one of the methods, rather combining few different approaches. The goal of this work isn't concerned about giving the detailed information about each algorithm, but rather to develop the intuition of the approach to the described subject. Review of the given steps is presented in the following sections.

2. General facial features detection scheme

Typical image analysis and in particular facial features detection usually consists of several steps:

1. Preprocessing.
2. Defining regions of interest.
3. Features extraction.
4. Classification.
5. Postprocessing.

In many cases not all parts of the process are used, but their subset is always present. Example of the system consisting of all of the proposed steps may be observed in the face detection algorithm by Discrete Gabor Jets (Naruniec & Skarbek (2007)). Within this approach image is preprocessed by the Gaussian blurring to reduce the influence of the noise. Since all of the detected fiducial points are placed on the edges, the regions of interest are obtained by thresholding the magnitude of the Sobel filters response. Extraction is performed in the circular neighborhood of each edge pixel using FFT, integral image and simple min/max normalization. Classification is based on the modification of linear discriminant analysis adjusted for the two-class (fiducial point/non-fiducial point) problem. In the final steps all of the edges are assigned to one of the five categories: left eye corner, right eye corner, left nose corner, right nose corner or non-face point. Verification of the points - postprocessing step, is performed by fitting the defined points to the face graph.
Subsequent section describe each of the processing steps in more detail.

Fig. 2. Histogram equalization examples. On the left - original images, on the right - processed images. On the first image equalization significantly improves the quality of the face image, while on the latter the effect is opposite.

2.1 Preprocessing and defining ROI

Since facial features are placed on the specific regions, a procedure for removing most of the background can be defined. This stage has to be proceeded very carefully, because removing proper regions at this moment will result in a failure of the whole algorithm. Region of interest (ROI) is defined here as a facial feature candidate point or region.

Depending on the representation of the facial features, methods can be divided to region based and point based. Contour based ROI definition could be also defined, but these methods aren't usually used within this problem and thus will not be discussed here.

In some applications preprocessing may increase the accuracy of the localization. This applies mostly to the cases where the acquisition parameters are insufficient, for example poor lighting, noise or inadequate camera properties. Typical operations performed on the data includes:

- noise removal - Gaussian blurring, median, mean filters;
- lighting normalization - min/max normalization, histogram equalization, removing low-pass frequencies;
- removing camera distortion

However it must be noticed, that sometimes preprocessing can decrease efficiency of the detector. For example blurring could remove edges, that are crucial in many contour based methods. In good lighting conditions histogram equalization could decrease the contrast of the face (Figure 2).

Point based ROI detection can be performed in various ways. Most of the facial features, for example eye corners, mouth corners, nostrils, are placed on the edges. Therefore thresholding the responses of edge filters based for example on Prewitt, Sobel or Roberts operators can significantly reduce the number of analyzed pixels (Figure 3). Further reduction of the number of pixels can be achieved by using corner detectors. There are several methods for accomplishing this task. One of the simplest methods - Moravec corner detector (Moravec (1980)) in based on the assumption, that the sum of absolute differences (SAD) between intensity values of the window anchored in the corner position and windows anchored in

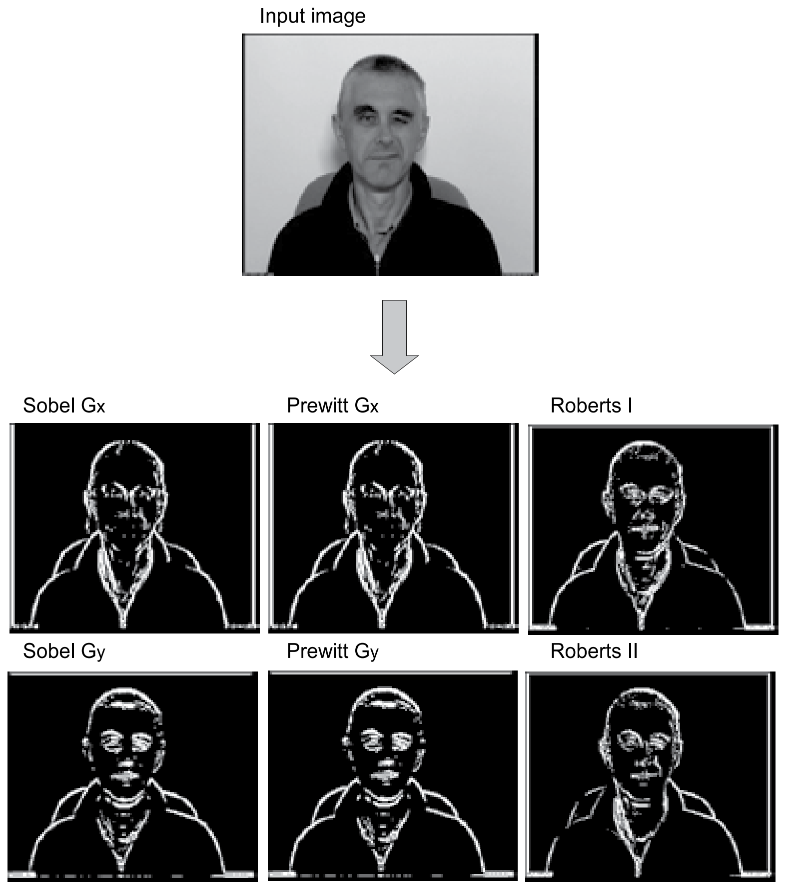


Fig. 3. Popular binarized edge filters responses for the face image.

the closest neighborhood of the analyzed point are high. Unfortunately algorithm doesn't consider directionality. Particularly the SAD can be low for the regions, that highly differ in the directionality of the edges and thus should be marked as corners. This disadvantage has been removed by the Harris corner detector (Harris & Stephens (1996)). Instead of taking intensities of the pixels directly, algorithm analyses following matrix of partial derivatives:

$$M = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, \tag{1}$$

where I_x and I_y defines gradients in x and y axes. Value of the eigenvalues of M define the variation in the edge direction. High value of both eigenvalues define corner.
Another interesting method for detecting corners is Features from Accelerated Segment Test (FAST) (Rosten & Drummond (2006)). Simple algorithm relies on the absolute difference of the analyzed pixel to the neighboring 16 pixels placed on defined circular vicinity. Advantage of such an approach is a very high speed of analysis, while achieving good results in particular applications.

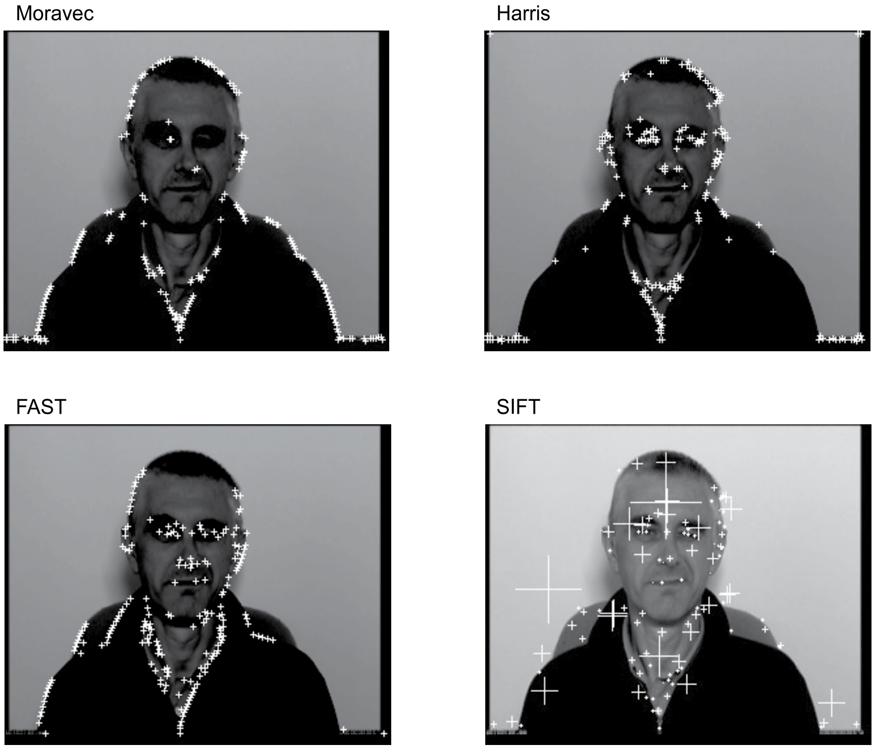
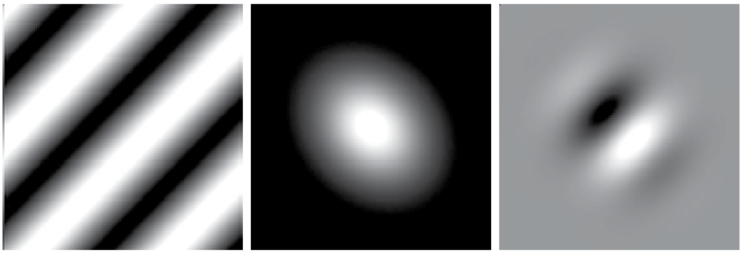


Fig. 4. Results of different corner detectors.

One of the most advanced interest point detection algorithm is Scale-Invariant Feature Transform (SIFT) Lowe (1999). Candidate points (keypoints) are detected using Differences of Gaussians (DoG) thresholding. In the second step non-maximum supression is applied in the 26 elements neighborhood - namely 8 pixels in the vicinity of the pixels and 18 points neighboring analyzed pixel in adjacent scales. In the end the edges are removed from the image by analyzing eigenvalues of the Hessian matrix. This step is explained by the fact, that in the common application of the SIFT algorithm - merging 3D clouds of points, position of the edges may differ at different viewpoints.
The choice of these methods should be adjusted to the particular detected set of points. For example if the goal of the algorithm is to detect corners of the eyes, the SIFT algorithm wouldn't be a good choice, in opposite to the Harris corner detector. Some results of the interest point detection are presented in Figure 4.
If the ROI is specified by the region, there are two most common approaches to this problem. The first one is color segmentation. Because the color of the skin is cumulated in the compact cluster of the RGB color space, skin and non-skin pixels can be distinguished from the image. On the other hand, general skin model, covering all the nationalities and races, is difficult to achieve. Another method for the ROI extraction is face detection. This topic is broadly described (for example in Hjelms & Low (2001); Naruniec (2010)), and thus it won't be analyzed here.



a) b) c)

Fig. 5. Example of defining Gabor filter mask. Real part of a) Gabor sinusoidal carrier, b) gaussian envelope, c) resulting filter by multiplication a) and b).

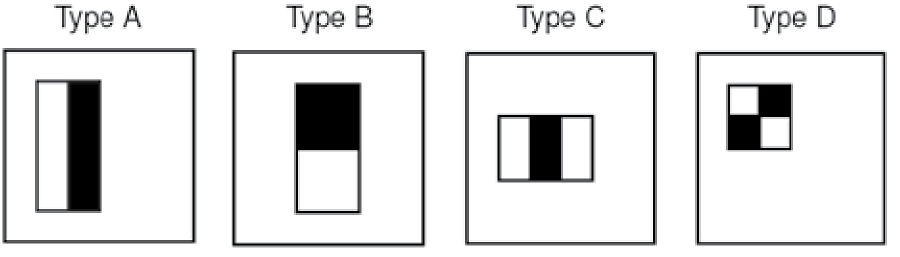


Fig. 6. Regions used in the Viola Jones AdaBoost detector.

2.2 Extraction

After defining initial regions or points of interest, a method for features extraction has to be given. The algorithms differ in the shape of the neighborhood and type of analysis.

One of the most known texture descriptors and facial features descriptor consists of the set of 40 Gabor filter responses (Wiskott et al. (1997)) and this set is called a "jet". Shape of the filter is defined by the two components: sinusoidal carrier and gaussian envelope (Figure 5). Set of functions with 8 orientations and 5 wavelengths form the Gabor jet.

Similarly descriptors formed by the Angular Radial Transform (ART) are computed by convolving the image with created base functions. The transformation is defined in the polar coordinates. Function consists of two components: modulation in angular direction (complex numbers) and sinusoidal function in radial direction (real numbers). In order to gain invariance to rotation, the absolute value of the complex function is taken into further consideration. Usually a set of 33 art coefficients are computed (3 wavelengths in radial direction and 11 in angular direction).

Gabor and ART coefficients are computationaly expensive and therefore inadequate for many real-time applications. Simple alternative to these methods are contrast features used in AdaBoost face detector (Viola & Jones (2001b), Figure 6). Set of such region contrasting filters is used for further AdaBoost classification. Integral image computed to speed up the algorithm provides result of summing any window in the image, in only 4 addition operations.

Another important issue in features extraction is reduction of dimensionality of the data. Simple projection of the data covariance matrix to the eigenvectors in many cases allow to represent vector in more compact form, while preserving most of the signal energy. Result of such principal components analysis (PCA) can be also achieved in a simpler way - by performing SVD decomposition on the zero-mean data and choosing first left singular vectors corresponding to the largest singular values. Another decorrelation technique used for this task is independent components analysis (ICA) (Duda et al. (2000)). It's goal is usually interpeted in audio processing. Assuming, that a sound is produced from many source signals, ICA provides methods for the blind separation of these inputs.

Facial features extraction can be also performed by fitting actual face to the predefined model. One of the most commonly used method in this scenario is active shape proposed by Cootes et al. (1995). Because of the fact, that the points placed on the facial features are highly correlated, authors apply PCA to define parameters that control the shape of the whole face model. In this way changing one parameter results in the deformation of all points present in the grid. Matching is performed by deforming the model in such way, that it fits the edges present in the image. Extension of this work is called active apperance models (Cootes et al. (1998)). Within this approach, the texture information is given in addition to the shape parameters.

It is also worth mentioning, that extraction of the pixel intensities is often followed by transformations such as discrete cosinus transform (DCT), fast fourier transform (FFT) or the wavelets.

2.3 Classification

Classification of the facial features can be defined in several ways. In the simplest case, classification is based solely on the euclidean distance of the descriptor to the predefined models. Such approach is efficient only for the obvious cases, but in most of the algorithms, more advanced techniques are used.

Another basic classification algorithm is based on the Bayes theory for the conditional probability:

$$P(y_i|x) = \frac{P(x|y_i)P(y_i)}{P(x)} \tag{2}$$

where y_i denotes the class of the analysed object and x is a descriptor. The probability $P(x|y_i)$ is usually modeled in the training step by the mean and the covariance. The value of $P(x)$ is abbreviated at the step of descriptor matching and therefore doesn't have any influence for the classification result.

Better solution for the class separation can be achieved by discrimination methods such as linear discriminant analysis (LDA) Fisher (1936). This method takes in consideration the within class variance R_w and between class variance R_b (see figure 7). Minimalization term is defined as follows:

$$J_{LDA}(w) = \frac{w^t R_b w}{w^t R_w w} \tag{3}$$

Function can be minimized by finding the eigenvalues of the $(R_w)^{-1}R_b$ matrix. Other solutions of this problem are based on SVD decomposition. In this case data are projected firstly on the R_w matrix, and the eigenvectors corresponding to the lowest eigenvalues are taken. In the second step, projected data are maximized in the terms of the R_b matrix by choosing eigenvectors corresponding to the largest eigenvalues.It appears, that for some particular problems dual LDA (DLDA) problem formulation (Leszczynski & Skarbek (2007))

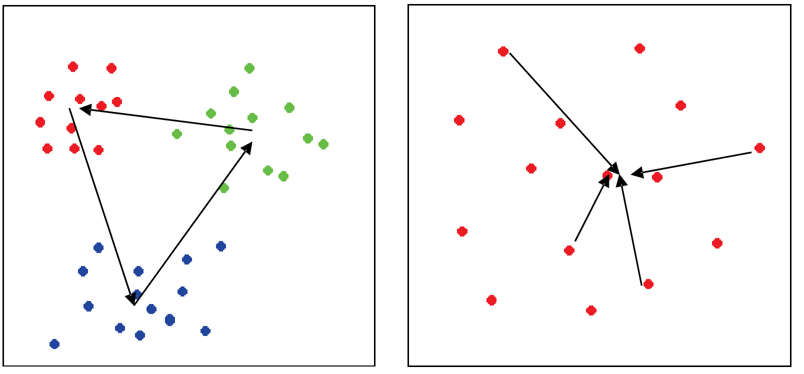


Fig. 7. Graphical illustration of between class (left side) and within class (right side) scatter. Different color correspond to different classes.

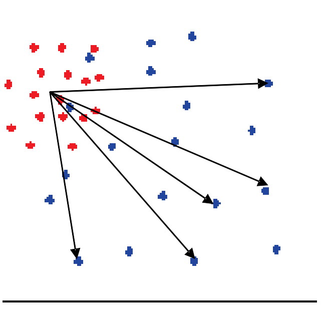


Fig. 8. Graphical illustration of between class variance in modified linear discriminant analysis. Red points correspond to the facial features, while the blue ones - the background.

can give better results. The maximalization function is defined as:

$$J_{DLDA}(w) = \frac{w^t R_w w}{w^t R_b w} \tag{4}$$

The differences in the approaches arise from orthogonality of the vectors minimizing R_w and maximizing R_b. To cope with the problem more generally, SDA analysis can be applied. The initial data is clustered to the moment, in which the angle between two optimization vectors exceeds specified threshold.

In the case of facial features/background two-class classification problem, other solution can give better results. Becouse it is very hard to define mean and variance of the background objects, therefore only facial feature within-class variance can be optimized. Moreover becouse the background class is very differentiated, treating every background example a separate class yields to better results. This assumptions are formulated in the modified linear discriminant analysis (MLDA) - (Naruniec & Skarbek (2007), see figure 8).

Another technique yelding very good separation results is support vector machine (SVM). Algorithm tends to separate two classes by providing two parallel linear hyperspaces leaning on some vectors of the data (support vectors) while retaining large margin between these hyperspaces (figure 9). Extension of this method introduces the error measure, that allow

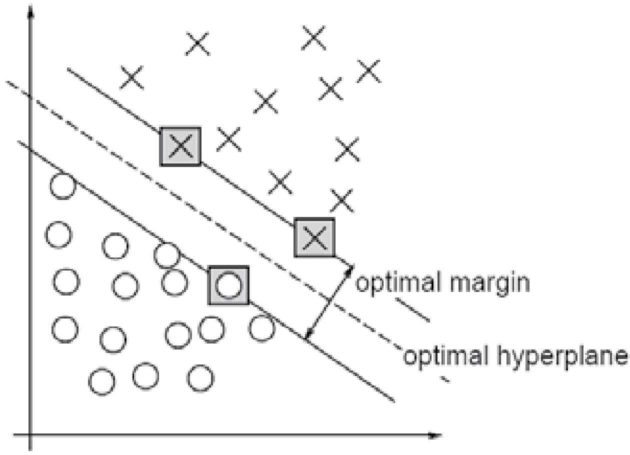


Fig. 9. SVM in two dimensional space. Crosses and circles are the two separated classes. Two lines form hyperplane diving these two distributions. (Cortes & Vapnik (1995)).

some vectors to be misclassified. Also kernel methods performing non-linear classification have been introduced. More informations about this subject can be found in the work of Cortes & Vapnik (1995).
AdaBoost is a method for combining many "weak classifiers" - having poor accuracy results to the very efficient "strong classifiers" Freund & Schapire (1995). In every iteration of the algorithm classifier with the lowest error according to the actual training examples weights, is added to the final classifier. Classification error is defined as:

$$\epsilon(\omega,\theta) = \frac{1}{2}\sum_{i=1}^{L} w_i|\delta_w(o_i) - v_i| \tag{5}$$

where ω denotes weak classifier, θ is the detection threshold, L - number of the training examples, w_i is the weight of the i-th training example and v_i denotes the label of i-th example ("1" for the facial feature, "-1" for the background).
At every iteration weights are updated using following formula:

$$w_{i,t+1} = \frac{w_{i,t}e^{-\gamma_i(o_i)v_i}}{\sum_{i-1}^{L} w_{i,t}e^{-\gamma_i(o_i)v_i}} \tag{6}$$

Costs of the positive or negative decision γ are computed using algorithm heuristics.
AdaBoost method have prooven to give fast and accurate results in the face detection (Viola & Jones (2001a;b)) and region based facial features detection scheme (Goldmann et al. (2006)).

2.4 Postprocessing

After classifying regions or points to the specific class, validation and refining of the selected facial features can be applied.

Fig. 10. Mixture coefficients of the model for face image patches. Bright pixels indicate a high probability for skin, dark pixels indicate a low probability for skin (Hoffmann et al. (2009)).

First remark concerns merging close results. Some detection algorithms may provide many results of the single facial features. In order to combine close fiducial points, simple clustering can be applied. In the case of regions, overlapping windows are merged in order to get single response.

Simplest approach for facial features validation is a geometrical matching. Relations between eyes, nose or mouth can be defined manually by the knowledge, or automatically by analysing specified data set.

Graph based methods convert each of the classified point to the graph node with assigned value, computed for example by the confidence of classification. In the next step possible graph values are compared to the trained mean face model. All the points that fits the model are marked as true faces, while the rest of the points are eliminated as false acceptances.

Validation can be also applied using color information, for example by convolving the face image with the color face patch (see Figure 10 - Hoffmann et al. (2009)).

Accuracy refinement is usually performed by moving facial features contours to the edges or by fitting to the specified model (for example created by the active shapes).

3. Conclusion

In this chapter a methodology for the facial features detection has been given. It describes all the basic semantic analyse stages - preprocessing, defining regions of interest, features extraction, classification and postprocessing.

4. References

Ben Jemaa, Y. & Khanfir, S. (2009). Automatic local Gabor Features extraction for face recognition, *ArXiv e-prints* .

Celiktutan, O., Akakin, H. & Sankur, B. (2008). Multi-attribute robust facial feature localization, *Automatic Face & Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, pp. 1–6.

Cootes, T. F., Edwards, G. J. & J., T. C. (1998). Active appearance models, *Lecture Notes in Computer Science* 1407.

Cootes, T. F., Taylor, C. J., Cooper, D. H. & Graham, J. (1995). Active shape models-their training and application, *Computer Vision and Image Understanding* 61(1): 38–59.

Cortes, C. & Vapnik, V. (1995). Support-vector networks, *Machine Learning* 20: 273–297. 10.1023/A:1022627411411.
URL: *http://dx.doi.org/10.1023/A:1022627411411*

Duda, R. O., Hart, P. E. & Stork, D. G. (2000). *Pattern Classification (2nd Edition)*, 2 edn, Wiley-Interscience.

Erukhimov, V. & Lee, K. (2008). A bottom-up framework for robust facial feature detection, *FG*, pp. 1–6.

Fisher, R. (1936). The use of multiple measurements in taxonomic problems, *Annals of Eugenics* 7: 179–188.

Freund, Y. & Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting, *European Conference on Computational Learning Theory*, pp. 23–37.

Goldmann, L., Monich, h. U. & Sikora, T. (2006). Robust face detection based on components and their topology, Vol. 6077, SPIE, p. 60771V.

Harris, C. & Stephens, M. (1996). A combined corner and edge detector, pp. 147–151.

Hjelms, E. & Low, B. K. (2001). Face detection: A survey, *Computer Vision and Image Understanding* 83(3): 236 – 274.
URL: *http://www.sciencedirect.com/science/article/B6WCX-458P9XF-3/2/25c703bcdc7e96439e46210c9c9fffc2*

Hoffmann, U., Naruniec, J., Yazdani, A. & Ebrahimi, T. (2009). Face Detection Using Discrete Gabor Jets and a Probabilistic Model of Colored Image Patches, *in* J. Filipe & M. S. Obaidat (ed.), *e-Business and Telecommunications, Communications in Computer and Information Science, Volume 48. ISBN 978-3-642-05196-8. Springer-Verlag Berlin Heidelberg, 2009, p. 331*.

Leszczynski, M. & Skarbek, W. (2007). Biometric verification by projections in error subspaces, *RSKT*, pp. 166–173.

Lowe, D. (1999). Object recognition from local scale-invariant features, pp. 1150–1157.

Moravec, H. (1980). Obstacle avoidance and navigation in the real world by a seeing robot rover, *tech. report CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University doctoral dissertation, Stanford University*, number CMU-RI-TR-80-03.

Naruniec, J. (2010). A survey on facial features detection, *International Journal of Electronics and Telecommunications* .

Naruniec, J. & Skarbek, W. (2007). Face detection by discrete gabor jets and reference graph of fiducial points, *Rough Sets and Knowledge Technology*, Springer Berlin / Heidelberg, pp. 187–194.

Rosten, E. & Drummond, T. (2006). Machine learning for high-speed corner detection, *In European Conference on Computer Vision*, pp. 430–443.

Viola, P. & Jones, M. (2001a). Fast and robust classification using asymmetric adaboost and a detector cascade, *Advances in Neural Information Processing System 14*, MIT Press, pp. 1311–1318.

Viola, P. & Jones, M. (2001b). Robust real-time object detection, *International Journal of Computer Vision* .

Vukadinovic, D. & Pantic, M. (2005). Fully automatic facial feature point detection using gabor feature based boosted classifiers, *Proceedings of IEEE ICSMS* .

Wiskott, L., Fellous, J., Krï£¡ger, N. & von der Malsburg, C. (1997). Face recognition by elastic bunch graph matching, *IEEE Trans. Pattern Analysis and Machine Intelligence* 19: 775–779.

12

Exploring and Understanding the High Dimensional and Sparse Image Face Space: a Self-Organized Manifold Mapping

Edson C. Kitani[1], Emilio M. Hernandez[1],
Gilson A. Giraldi[2] and Carlos E. Thomaz[3]
[1]*Universidade de São Paulo, São Paulo, São Paulo,*
[2]*Laboratório Nacional de Computação Científica, Petrópolis, Rio de Janeiro,*
[3]*Centro Universitário da FEI, São Bernardo do Campo, São Paulo,*
Brazil

1. Introduction

Face recognition has motivated several research studies in the last years owing not only to its applicability and multidisciplinary inherent characteristics, but also to its important role in human relationship. Despite extensive studies on face recognition, a number of related problems has still remained challenging in this research topic. It is well known that humans can overcome any computer program in the task of face recognition when artefacts are present such as changes in pose, illumination, occlusion, aging and etc. For instance, young children can robustly identify their parents, friends and common social groups without any previous explicit teaching or learning.

Some recent research in Neuroscience (Kandel et al., 2000; Bakker et al., 2008) has shown that there is some new information about how humans deal with such high dimensional and sparse visual recognition task, indicating that the brain does not memorize all details of the visual stimuli (images) to perform face recognition (Brady et al., 2008). Instead, our associative memory tends to work essentially on the most expressive information (Bakker et al., 2008; Oja, 1982). In fact, theoretical models (Treves and Rolls, 1994; O'Reilly and Rudy, 2001; Norman and O'Reilly, 2003) have indicated that the ability of our memory relies on the capability of orthogonalizing (pattern separation) and completing (pattern prototyping) partial patterns in order to encode, store and recall information (O'Reily and McClelland, 1994; Kuhl et al., 2010). Therefore, subspace learning techniques have a close biological inspiration and reasonability in terms of computational methods to possibly exploring and understanding the human behaviour of recognizing faces.

The aim of this chapter is to study the non-supervised subspace learning called Self-Organizing Map (SOM) (Kohonen, 1982; Kohonen, 1990) based on the principle of prototyping face image observations. Our idea with this study is not only to seek a low dimensional Euclidean embedding subspace of a set of face samples that describes the intrinsic similarities of the data (Kitani et al., 2006; Giraldi et al., 2008; Thomaz et al., 2009; Kitani et al., 2010), but also to explore an alternative mapping representation based on manifold models topologically constrained.

More specifically, the purpose of this work is to navigate on the locally optimal pathways composed of the SOM neurons to minimize inappropriate mappings where the standard SOM might show significant discontinuities and compare such visualization procedures on the original image space to understand the most important information captured by the non-supervised model. To minimize image variations that are not necessarily related to differences between the faces, we will carry out experiments on frontal face images available from two distinct public face databases that have been previously aligned using affine transformations and the directions of the eyes as a measure of reference. In this way, the pixel-wise features extracted from the images correspond roughly to the same location across all subjects. In addition, in order to reduce the surrounding illumination and some image artefacts due to distinct hairstyle and adornments, all the frontal images have been cropped to the size of 193x162 pixels, had their histograms equalized and have been converted to 8-bit gray scale. Our experimental results on the two distinct face image sets show that although the standard SOM can explain the general information extracted by its neurons, its intrinsic self-organized manifolds can be better described by an algorithm based on the principle of the locally optimal pathways and the idea of navigating on the graphs composed of the standard SOM neurons.

The remaining of this chapter is organized as follows. In the next section, we briefly review some literature about perceptual and cognitive processes related to human memory and the mechanisms of pattern completion and pattern separation. Next, in the third section, we provide some background definition of SOM and highlight shortly its biological principle of organization that inspired Kohonen in the early eighty's. Also, in the same section, we introduce the standard SOM algorithm based on the competitive learning rule. The main contribution of the chapter is then presented in the subsequent subsection entitled A Self-Organized Manifold Mapping (SOMM) Algorithm. In this subsection, we describe a new algorithm that is able to understand the information extracted from the data, identifying and explaining the nature of the groups or clusters defined by the SOM manifolds. The two distinct public face databases used to carry out the experiments are described in the fourth section. Next, in the fifth section, we show several experimental results to demonstrate the effectiveness of the SOMM algorithm on providing an intuitive explanation of the topologically constrained manifolds modelled by SOM in well-framed face image analysis. Finally, in the last section of the chapter, we conclude this work, summarizing its main points.

2. Neurological and psychological aspects

Several perceptual and cognitive processes guide the task of face recognition in humans. However, one of the most important processes is the memory. Humans do not memorize all the details and features received by the sensory system (Purves et al., 2001). In fact, the human brain has an outstanding capability of forgetting useless information (Brady et al., 2008, Purves et al., 2001).

Basically, human memory can be divided into two groups: declarative and non-declarative memory (Purves et al., 2001). Declarative memory is related to memorizing facts and events and can be accessed for conscious recollection. Facts are information learned during a high level cognition process, such as studying some specific subject. Events are information that one has had as a life experience, for example: birthday, wedding, etc. Episodes at non-declarative memories, on the other hand, are information that cannot be accessed formally.

In other words, it cannot be explained explicitly by words and neither how it occurs nor happens. Examples of non-declarative memory are: physical skills such as swimming, riding a bicycle, or emotional responses such as fear, happiness, etc. Additionally, memories are also categorized as short-term-memory and long-term-memory (Purves et al., 2001). Short-term memories have a limited capacity to hold information and consequently retain it during short period of time (Anderson; 2005), but long-term ones tend to retain it permanently. The process that converts information into long-term memory is known as memory consolidation (Bear, Connors, Paradiso; 2007). The memory consolidation is part of our learning process and is strongly necessary, for instance, to the face-matching task (Kandel et al., 2000).

The brain area responsible for storing the declarative memory is called the Medial Temporal Lobe (MTL) (Bear, Connors, Paradiso; 2007). The MTL is a complex interconnected systems of the brain and one of its most important structures is the hippocampus. Recent experiments carried out on rats have showed that lesions at the hippocampus might affect our capability of learning and retaining information (Bear, Connors, Paradiso; 2007). Yet, in the past, a computational model presented by Treves & Rolls (Treves and Rolls, 1994) had already indicated that some parts of the hippocampus seem to create a sparse and orthogonalized representation of our sensory input and episodic memories. Currently, there is no doubt that the hippocampus plays an important role to encode new episodic memories and, additionally, to prevent the risk of forgetting past memories (Kandel et al., 2000, Kuhl et al., 2010).

Using high-resolution (1.5 millimeters isotropic voxels) functional Magnetic Resonance Imaging (fMRI), Bakker et al. (Bakker et al., 2008) have studied the activity in the human brain MTL area on a set of pattern visualization experiments. The experiments consisted of presenting to each one of a total of eighteen volunteers a sequence of pictures of common objects, such as apples, toys duck, thread balls, wall outlet and etc. The set of pictures used is composed of 144 subsets of slightly different images of the same object, with essentially variation in pose and rotation. The authors have noticed that several brain structures of the MTL area, especially a specific area of hippocampus named CA1, have been activated when pictures of the same object have been presented repetitively and in a interleaved way.

In fact, our brain process of retrieving information can be further described by two main mechanisms: pattern completion and pattern separation (Kuhl et al., 2010). The mechanism of pattern completion is essentially related to the problem where the incoming pattern of some sensory input and the pattern stored in the memory are not exactly the same, but share some similarities. In the mechanism of pattern separation, the similarities between the incoming and stored patterns, if do exist, are minimal and both patterns have, in contrast, a strong degree of dissimilarities that can be mathematically considered as non-correlated or orthogonal.

This work focuses on the mechanism of pattern completion and the role of the human brain hippocampus as an associative memory to propose a new algorithm for the SOM competitive neural network proposed by Kohonen (Kohonen; 1982). Since this pioneering work, it has been argued that SOM is not only a computational approach for data mining and clustering, but also a credible framework at the functional and neural levels to create a self-organization of the input space (Rolls; 2007) and model the human memory activities of encoding and retrieving information.

3. Self-Organizing Map (SOM)

A formal definition of organization is quite complex because it depends on the context. Some crystal structures are considered highly organized due to their symmetry and structural repetition. Functions and hierarchy organize all biological structures, such as the nervous system, digestive system, circulatory system, etc (Kandel et al., 2000). However, in both cases, the definitions of "organization" are ambiguous.

For crystal structures, one finds symmetries and redundancy; on the other hand, a biological system is organized by functions. However, both definitions have in common the sense of similarity that allows us to cluster and hierarchize input patterns. In other words, organization is an association and composition of parts to explore a whole structure or behavior (Asby, 1962; Atlan, 1974).

According to the definition above, clustering is quite related with similarities or even dissimilarities. SOM is an unsupervised neural network developed by Kohonen (Kohonen, 1982; Kohonen, 1990) based on the biological principle of somatosensory organization. According to Kandel et al. (Kandel et al., 2000), there is a functional organization of perception and movement in human and mammals brain. There is also a specialized area in the brain cortex that organizes information coming from sensory pathways or going to motor control. Somatosensory cortex is the area accounting for organizing stimulus coming from different sensory systems, grouping them according to their similarities. In a similar fashion, motor cortex has surfaces dedicated to controlling parts of the body related to movement. This organization in substructures by functions is well-known by neuroscientists, however, why the brain creates this organization remains unclear (Purves et al., 2001).

Based on the biological principle of organization, Kohonen postulates that there are some reasons to have this organization: a) grouping similar stimulus minimizes neural wiring; b) creates a robust and logical structure in the brain, avoiding "crosstalk"; c) from information organized by attributes a natural manifold structure from input patterns can emerge; and d) reduces dimensionality by creating representations (codebooks vectors) that preserves neighborhood relationship between input patterns. Each codebook, also known as BMU (Best Match Unit), retains the most important invariant features that represent a group of input patterns, characterizing an arguable but intuitively analogous behaviour to the pattern completion mechanism of the human brain.

3.1 The Standard SOM algorithm

SOM can be defined as an unsupervised artificial neural network that maps a nonlinear relationship between input patterns in high dimensional space and makes this relationship an ordered and smoothed mapping of input data manifold. SOM has a competitive learning rule, but does not have a rule of convergence or function to minimize. Instead, the algorithm of SOM works with a number of interactions during weight adaptation.

Figure 1 illustrates a Kohonen network of 3×3 output neurons fully connected to the input layer composed by only two neurons. The network is created from a 2D lattice of 'nodes' composed of the output neurons and the input layer. Each output neuron has a specific position $(x,y)\in\Re^2$ and contains a vector of weights of the same dimension as the input vector. That is, if the network has m output neurons and the training set consists of vectors $(x_0(t),x_1(t),x_2(t),\ldots,x_{n-1}(t))\in\Re^n$, then we have $m\times n$ weights $w_{ij}(t)$, $0\le i\le M-1$, $0\le j\le m-1$ to set.

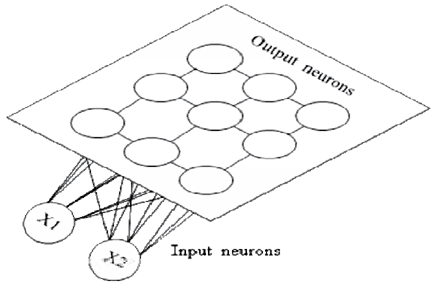


Fig. 1. An illustration of Kohonen network. Note that all input neurons are connected to all output nodes.

The algorithm can be described as follows:

1. Initialize network,
   1.1) Define the number m of output neurons that will compose the map and their lattice position (nodes): $r_0, r_1, r_2, \ldots, r_{m-1} \in \Re^2$,
   1.2) Define $w_{ij}(t)$, $0 \le i \le M-1$, $0 \le j \le m-1$, to be the weight from input neuron i to output neuron j at time t, where M is the size of the set of input training patterns. Initialize weights to small random values. Set the initial radius of neighbourhood around node j, denoted by $\sigma_j(0)$ to be large,
   1.3) Define the number of iteration $T >> M$,
2. Present the input vector $(x_0(t), x_1(t), x_2(t), \ldots, x_{n-1}(t)) \in \Re^n$, where $x_i(t)$ is sent to the input node i at the time t, where n is the dimensionality of input space,
3. Compute the distance d_j between the input vector $x_i(t)$ and each output neuron j, given by $d_j = \min_{0 \le i \le M-1} \left| x_i(t) - w_{ij}(t) \right|$, $0 \le j \le m-1$,
4. Designate the BMU neuron r_c to be one with minimum d_j,
   Update the weights for node r_c and its neighbors, defined by the neighborhood size $\sigma_c(t)$. New weights will be:
   $w_{ij}(t+1) = w_{ij}(t) + \alpha(t) h_{ci}(t) \left(x_i(t) - w_{ij}(t) \right)$,
   where $\alpha(t)$ is the learning factor:
   $\alpha(t) = \alpha_0 \left(1 - \frac{t}{T} \right)$,
   $h_{ci}(t)$ gives the amount of influence that a neuron r_i has on its learning as a function of its distance from the BMU neuron r_c :
   $h_{ci}(t) = \exp\left(-\frac{\left\| r_c - r_i \right\|^2}{2\sigma_c^2(t)} \right)$,
   Finally, $\sigma_c(t)$ define the radius of influence of the BMU, which can be computed by: $\sigma_c(t) = \sigma_0 \exp\left(-\frac{t}{\lambda_c} \right)$, and λ_c is an integer number related to the time of influence of the neighbor radius,
5. Return to step 2 until $t = T$.

From an initial distribution of random weights, the SOM eventually settles into a map of stable zones after some iterations. The term $\alpha(t)$ is a gain term that decreases in time so slowing the learning process. Besides, the neighbourhood size $\sigma_c(t)$ decreases in size as time goes on, thus localizing the area of maximum activity.

3.2 A Self-organized manifold mapping algorithm

Several studies have provided us with some insight about how to interpret the output of SOMs (Brugger et al., 2008; Bauer & Pawelzik, 1992; Kiviluoto, 1995). One of the best-known tools in this regard is the U-Matrix (Ultsch, 2003) that gives us a quantitative summary of the topological relationships between similar data samples. The result of the U-Matrix map is a complex image (coloured or monochromatic) indicating peaks and valleys that represent Euclidean distances between neighboured neurons.

Essentially, the resulting map preserves the topological distribution at the input space of the entire sample data considered. Figure 2 illustrates an example of a coloured U-Matrix map and its hexagonal 5×4 SOM, where each neuron w_{ij}, $0 \le i \le M$, $0 \le j \le m-1$, has been arbitrary identified by a number. It is possible to see at least two groups of patterns in blue separated by a central chain in red. The chain of high values in the U-matrix indicated by the reddish colours is a representation of some prototypes that are far from both groups and probably describe some data outliers with distinct information about the dataset considered.

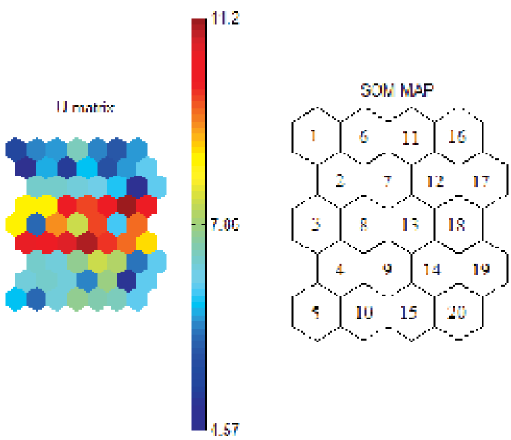


Fig. 2. An illustration of a coloured U-Matrix map and its corresponding SOM where each neuron has been arbitrary identified by a number.

However, to understand the relationship between the information captured in the U-Matrix and the samples, as well as to identify and explain the nature of the groups or clusters defined by the manifolds, it would be helpful to represent all the SOM neurons and their corresponding similarities and dissimilarities on the original data space.

Based on the principle of the locally optimal pathway and the idea of navigating on the neurons that compose the SOM, we propose an algorithm named Self-Organized Manifold Mapping (SOMM) that seeks the pathways or manifolds described by the standard SOM. The SOMM algorithm can be described as follows:

a. Calculate the SOM composed of k neurons using the standard Kohonen´s algorithm. Create the list $A = \{0,1,2,\ldots,k-1\}$;
b. Calculate pairwisely the Euclidean distance d_{ij} of all k neurons;

c. Create a (k x k) matrix with all pairwise distance between all k neurons;

d. Create the list $V = NULL$. Set $r = A_1$, $d_{\min} = \infty$, $i = 0$;

d.1) Insert $V \leftarrow \{r\}$,

d.1.a) $i = i + 1$.

d.2) If $A - V = \varnothing$ go to (e).

d.3) Find $d = \min\left(d_{sr}, \quad s \in A - \{V_{i-1}, V_i\}\right)$. Let $s*$ such that $d = d_{s^*r}$.

d.4) If $s^* \in \{V\}$ go to (e) Else set $r = s^*$ and $V \leftarrow \{r\}$. Go to step (d.1a).

e. LOOP:

e.1) If $V_i = LOOP$ go to (f)

e.2) Find $d = \min(d_{sr}, \quad s \in V - \{V_{i-1}, V_i\})$. Let $s*$ such that $d = d_{s^*r}$.

e.3) Insert $V \leftarrow \{s^*\}$, $i = i + 1$.

f. f) Group BMUs according the order: $V_1, V_2, ...,$

g. If $A - V \neq \varnothing$ then

g.1) $A = (A - V) \cup \{V\}$,

g.2) k = number of elements in A,

g.3) go to (b).

A simple way to explain this algorithm is to understand the output neurons, represented by the weights w_{ij} computed in the SOM algorithm, as a set of nodes of a fully connected graph (Cormen et al. 2001; Pölzlbauer, Rauber, Dittenbach, 2005; Mayer, Rauber, 2010) in the parameter space. Each edge in this graph has a cost given by the Euclidean distance between its ends. Therefore, the $k \times k$ matrix calculated in steps (b)-(c) is a symmetric one holding the edge costs in the graph.

More specifically, in step (d) it is created a list V and in step (d.1) the algorithm inserts in V each visited node. Given a node r, the step (d.3) seeks for the closest neuron $s*$ such that $s^* \notin V - \{V_{i-1}, V_i\}$; that means, $s*$ does not belong to the last visited edge of the graph.This step implements a greedy algorithm that makes the locally optimal choice at each stage generating a *locally optimal pathway* that connects a subset of SOM neurons. This is necessary because the idea is to generate a pathway that crosses different clusters but without losing the notion of similarity in the parameter space. If $s^* \in \{V\}$ then we have a loop, like the one exemplified in Figure 3. In this case, the pathway that starts at node 1 ends in the loop $(11) \to (3) \to (4) \to (11)$. The step (e) completes the pathway, which in this figure is composed by the sequence: $V = (1) \to (2) \to (11) \to (3) \to (4) \to (11)$.

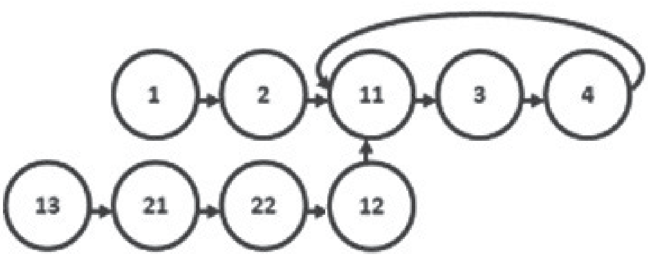


Fig. 3. Two connected pathways with a common loop. The first one starts at node 1 and finally enters in the loop $(11) \to (3) \to (4) \to (11)$, whereas the second path starts in the node 13 and ends in the same loop.

Additionally, the step (g) identifies that there are nodes still not visited by the algorithm. Following the idea of crossing different clusters we must allow that a node $r \in A - V$ might be connected with a node $s \in V$, like node $r = 12$ shown in Figure 3. In terms of the algorithm, it is equivalent to consider V as a node in a new graph (steps (g.1)-(g.2)), compute the new distances in step (b) and seek for another pathway as before. Therefore, this novel algorithm brings the possibility of uncovering clusters not visible by U-Matrix technique or the standard SOM approach.

4. Face databases

We have used frontal images of two distinct face databases publicly available to carry out the experiments. The first database is maintained by the Department of Electrical Engineering of FEI, São Paulo, Brazil (Thomaz and Giraldi, 2010). In this dataset, the number of subjects is equal to 200 (100 men and 100 women) and each subject has two frontal images (one with a neutral or non-smiling expression and the other with a smiling facial expression), so there is a total of 400 images with no significant differences in skin colour to perform the high dimensional and sparse image face analysis. The second dataset is the well-known FERET (Philips et al., 1998) database. In the FERET database, we have considered only 200 subjects (107 men and 93 women) and each subject has two frontal images (one with a neutral or non-smiling expression and the other with a smiling facial expression), providing a total of 400 images with significant differences in skin colour to perform as well the experiments.

To minimize image variations that are not necessarily related to differences between the faces, we previously aligned all the frontal face images using affine transformations and the directions of the eyes as a measure of reference so that the pixel-wise features extracted from the images correspond roughly to the same location across all subjects. Also, in order to reduce the surrounding illumination and some image artefacts owing to distinct hairstyle and adornments, all the frontal images were cropped to the size of 193x162 pixels, had their histograms equalized and were then converted to 8-bit gray scale. Figure 4 illustrates some samples of the FEI (top row) and FERET (bottom row) datasets, highlighting samples of distinct gender, age, facial expression and ethnicity.

Fig. 4. Some samples of the FEI (top row) and FERET (bottom row) frontal images used in the experiments after the pre-processing procedure that aligned, cropped and equalized all the original images to the size of 193x162 pixels.

5. Experimental results

All the experiments have been carried out using the well-known SOM-Toolbox for Matlab created and released by CIS-Helsinki University of Technology (Vesanto, 1999). To address the memory issues related to computing the SOM on high-dimensional datasets, instead of analysing the SOMM algorithm directly on the pre-processed FEI and FERET face images, Principal Component Analysis (PCA) (Fukunaga, 1990) has been applied first to provide dimensionality reduction. However, in order to reproduce the total variability of the sample data, we have composed the PCA transformation matrix by selecting all the principal components with non-zero eigenvalues. Although some of these principal components might represent non-relevant information to understand the differences between the data samples, we are able to represent and further reconstruct the original images without adding any dimensionality reduction artefacts (Kitani et al., 2010).

We have divided our experimental results into two parts. Firstly, we have carried out some face image analyses to understand and visualize the pathways found by the SOMM algorithm where there are subtle differences between the data samples. Thus, we have used a subset of the FEI database composed of non-smiling and smiling face images of females only. Then, in the second part, we have investigated the usefulness of the SOMM algorithm on exploring and understanding the high dimensional and sparse image face space where the differences between the samples are not only related to facial expression but also to gender, ethnicity and age. The goal of the second experiment is to pose an alternative analysis where the differences between the samples are evident, using the whole two FEI and FERET datasets described in the previous section.

Figure 5 illustrates the standard SOM (top left), the pathways described by the SOMM algorithm (bottom left) and their corresponding visualization (top and bottom right) on the original face space using a subset of the FEI database composed of non-smiling and smiling face images of females only. It is important to highlight that since the SOMM navigation is based on the principle of the locally optimal path, it is only possible to visit a new neuron when its distance is minimal regarding all the other neurons previously visited. Therefore, the algorithm explicitly describes the discontinuities present at the high dimensional face image space due to the limited number of input samples. In other words, it is possible to see that SOMM could not find a unique graph that defines a single locally optimal path from non-smiling to smiling female face images. In fact, as shown on the bottom right part of Figure 5, we can see three feasible pathways or clusters: (1) samples that describe a definite smiling facial expression; (2) samples that describe the visual differences from non-convincing to convincing smiling facial expressions; (3) samples that describe the visual differences from non-convincing to convincing non-smiling facial expressions.

In the next two figures, we show the behaviour of the SOMM algorithm on navigating at high dimensional and sparse image face spaces where the differences between the samples are not only related to facial expression but also to gender, ethnicity and age. Figure 6 illustrates the standard SOM (top left), the pathways described by the SOMM algorithm (bottom left) and their corresponding visualization (top and bottom right) on the original face space using the whole set of frontal face images of the FEI database with both gender and facial expression differences.

Analogously to the previous results, three clusters have been found by the SOMM algorithm. Despite the gender differences available on this dataset, SOM has not clearly extracted this information on its standard mapping and neither SOMM has described it in a separated pathway or cluster. The smallest SOMM cluster, composed of 6 neurons, shows samples that describe a definite smiling facial expression with slightly more male facial

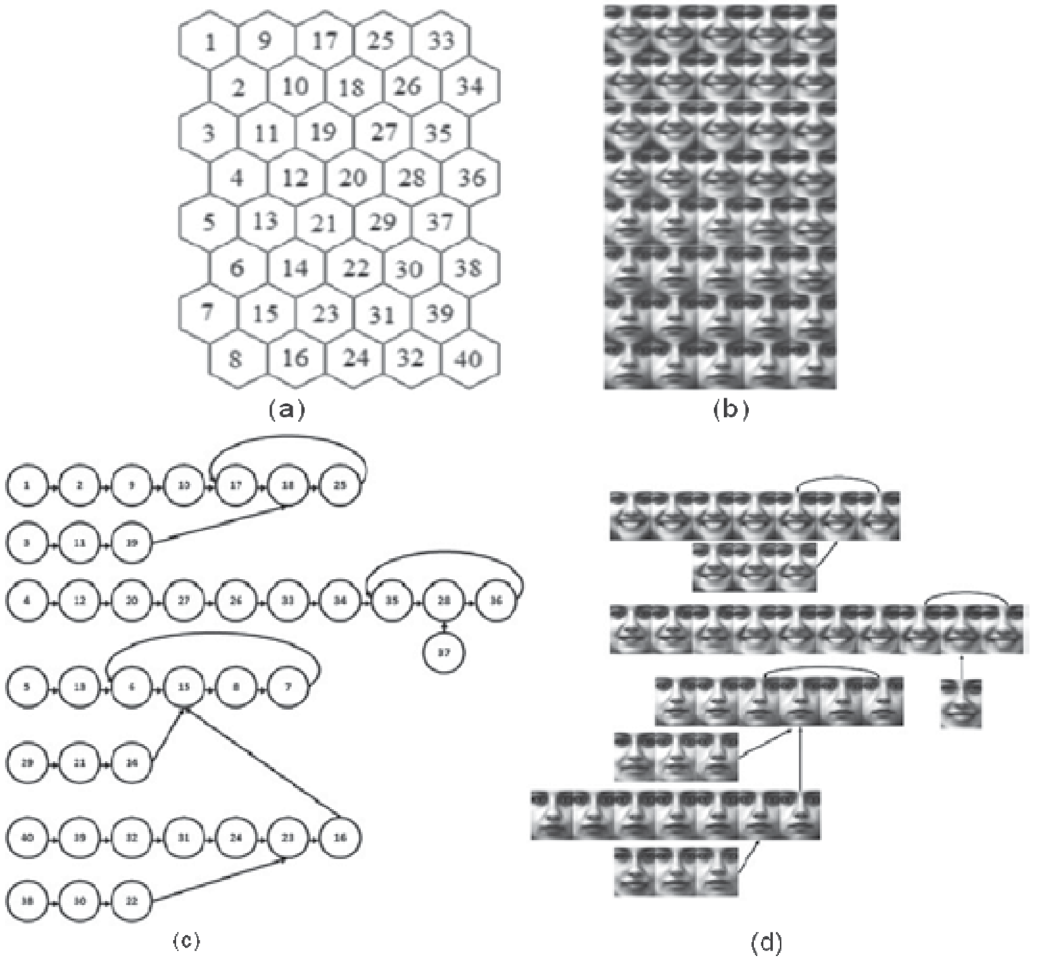


Fig. 5. Exploring the similarities and discontinuities of the high dimensional image face space composed of smiling and non-smiling female face images only of FEI database: standard SOM of size 8x5 (top left); visualization of the SOM neurons (top right); SOMM algorithm navigation (bottom left); visualization of the SOMM clustering (bottom right).

traits than female ones. A similar description is valid for the second smallest SOMM cluster, composed of 8 neurons, but rather with more female facial traits. However, the largest cluster clearly shows that the most expressive information captured by SOMM has been related to changes in facial expression, no matter the gender of the subjects analysed.

The last experimental results using the FERET dataset are presented in Figure 7. It can be seen that the main expressive information captured by SOM have been based on ethnicity and facial expression changes. The visualization of the standard SOM, illustrated on the top right part of Figure 7, shows clearly how the data set has been generally spread along the high dimensional face image space. It is possible to see that when we move from top to bottom we are able to see differences related mainly to ethnicity, no matter the facial expression or gender of the subjects. Besides, navigation on the SOM neurons from left to right highlights essentially information about changes on facial expression with minor

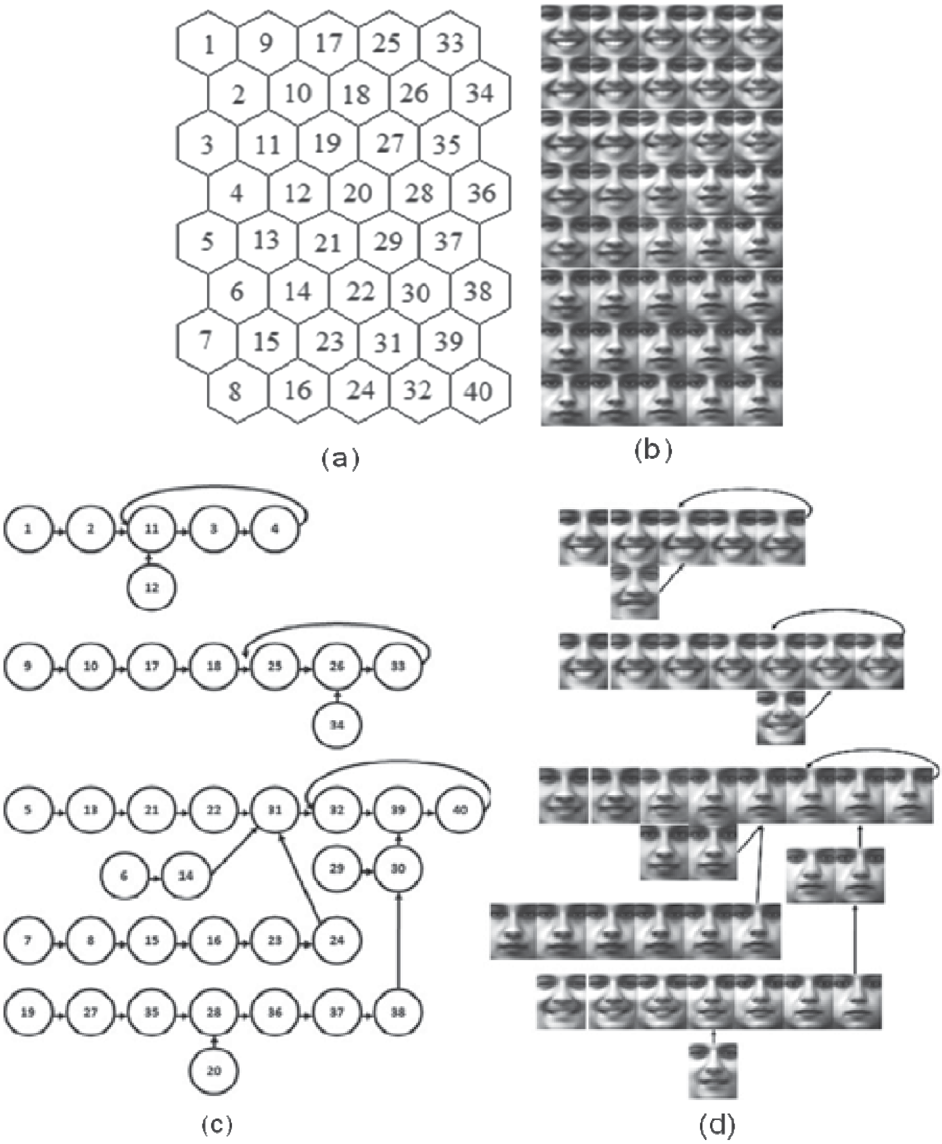


Fig. 6. Exploring the similarities and discontinuities of the high dimensional image face space composed of the whole set of frontal face images of the FEI database: standard SOM of size 8x5 (top left); visualization of the SOM neurons (top right); SOMM algorithm navigation (bottom left); visualization of the SOMM clustering (bottom right).

differences related to gender and ethnicity features. However, not all these pathways are feasible due to the discontinuities of the high dimensional and sparse image face space. In fact, as described by the SOMM algorithm, there are only five clusters possible to move along based on the principle of the locally optimal path. Therefore, although the standard SOM can explain the general information extracted by its neurons, its intrinsic self-organized manifolds have been only explicitly explained by the SOMM algorithm.

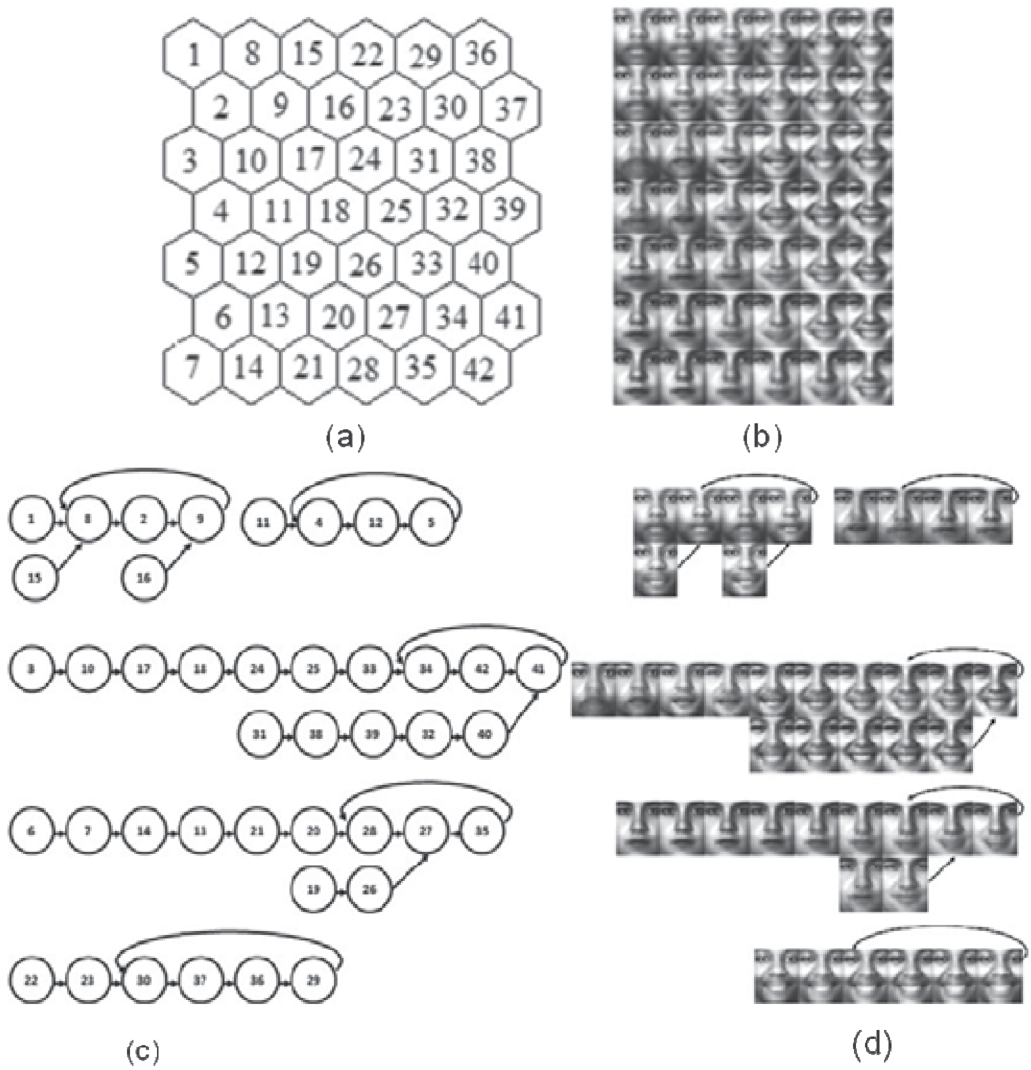


Fig. 7. Exploring the similarities and discontinuities of the high dimensional image face space composed of some frontal face images of the FERET database: standard SOM of size 7x6 (top left); visualization of the SOM neurons (top right); SOMM algorithm navigation (bottom left); visualization of the SOMM clustering (bottom right).

6. Conclusion

In this chapter, we proposed and implemented a self-organized manifold mapping algorithm that allows a better understanding of the information captured by the standard SOM neurons. The method is able not only to identify and explain the nature of the clusters defined by the SOM manifolds, but also to represent all the SOM neurons and their corresponding similarities and dissimilarities on the original data space. To describe the possible self-organized pathways to navigate on the high dimensional and sparse image face space, we constructed a neighbourhood graph on the SOM neurons based on the principle of the locally optimal path. Such graph visualization method explicitly provides information about the number of clusters that describes the sample data under investigation, as well as the specific features extracted and explained by them. We believe that the algorithm proposed might be a powerful tool in SOM analysis, providing an intuitive explanation of the topologically constrained manifolds modelled by SOM and highlighting some perceptual properties commonly present in well-framed face image analysis such as facial expression, ethnicity and gender.

7. Acknowledgment

Portions of research in this paper use subsets of the FERET database of facial images collected under the FERET program.

8. References

Anderson, J. R. (2005). Cognitive Psychology and Its Implications, 6th edition, Worth Publishers, New York USA.

Asby, W. R. (1962). In principles of self organization, H.von Forester & G.W. Zopf eds., pp 255-278, London UK.

Atlan, H. (1974). On a formal definition of organization, Journal of theory in biology, vol.45, pp. 295-304.

Bakker, A., Kirwan, C. B., Miller, M., Stark, C. E. L. (2008). Pattern separation in the human Hippocampal CA3 and Dentate Gyrus, Science Magazine, vol. 319, pp. 1640.

Bauer, H. U., Pawelzik, K. R. (1992). Quantifying the neighborhood preservation of Self-Organizing Feature Maps, IEEE Transaction on Neural Networks, vol. 3, no. 4, pp 570-579.

Bear, M. F., Connors, B. W., Paradiso, M. A., (2007). Neuroscience. Exploring the brain. Lippncott Willianms & Wilkins, 3rd ed.

Brady, T. F., Konkle, T., Alvarez, G. A., Oliva, A. (2008). Visual long-term memory has a massive storage capacity for object details, PNAS, Proceedings of the National Academy of Science of the United States of America, vol. 105, no. 38, pp. 14325-14329.

Brugger, D., Bogdan, M., Rosentiel, W. (2008) Automatic cluster detection in Kohonen´s SOM, IEEE Transaction on Neural Networks, vol. 19, no. 3, pp. 442-459.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2001). Introduction to Algorithms, 2nd ed. MIT Press.

Fukunaga, K. (1990). Introduction to Statistical Pattern Recognition, 2nd ed. Academic Press.

Giraldi, G. A., Rodrigues, P. S., Kitani, E. C., Sato, J. R., Thomaz, C. E. (2008). Statistical Learning Approaches for Discriminant Features Selection, Journal of the Brazilian Computer Society, 14(2), 7-22.

Kandel, E. R., Schwartz, J. H., Jessell, T. M. (2000). Principles of Neural Science, 4th ed., McGraw-Hill.

Kitani, E. C., Hernandez, E. Del. Moral, Thomaz, C. E., Silva, L. A. (2010). Visual Interpretation of Self-Organizing Maps, Proceedings of Neural Networks Brazilian (SBRN), IEEE CS Press, 37-42.

Kitani, E. C., Thomaz, C. E., Gillies, D. F. (2006). A Statistical Discriminant Model for Face Interpretation and Reconstruction, Proceedings of Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI), IEEE CS Press, 247-254.

Kiviluoto, K. (1995). Topology preservation in self-organizing maps, IEEE International Conference on Neural Networks, vol. 1, pp. 294-299, Washington, DC.

Kohonen, T. (1982). Self–Organization and Associative Memory, Springer Verlag, Berlin.

Kohonen, T. (1990). The Self-Organizing Map Proceedings of the IEEE, vol. 78, no.9.

Kuhl, B. A., Shah, A., DuBrow, S., Wagner, A. D. (2010). Resistance to forgetting associated with hippocampus mediated reactivation during new learning, Nature Neuroscience, v. 13, no. 4.

Mayer, R., Rauber, A. (2010). Visualizing clusters in Self-Organizing with minimum Spanning Trees, ICANN´10, Proceedings of the 20th International Conference on Artificial Neural Networks, pp. 426-431.

Norman, K. A., O'Reilly, R. C. (2003). Modeling Hippocampal and neocortical contributions to recognition memory: A complementary learning systems approach, Journal of Psychological Review, vol. 110, pp. 611-646.

O'Reily, R. C., McClelland J. L., (1994). Hippocampal conjunctive encoding, storage and recall: avoiding a trade off, Hippocampus vol. 4, pp. 661-682.

O'Reilly, R. C., Rudy, J. W. (2001). Conjunctive representation in learning and memory: Principles of cortical and Hippocampa function, Journal of Psychological Review, vol. 108, pp. 311-345.

Oja, E., (1982). A simplified neuron model as a Principal Component Analyser, Journal of Mathematical Biology, vol. 15, pp. 267-273.

Philips, P. J., Wechsler, H., Huang, J., Rauss, P. (1998). The FERET database and evaluation procedure for face recognition algorithms, Image and Vision Computing, 16(5), 295-306.

Pölzlbauer, G., Rauber, A., Dittenbach ,M. (2005). Graph projection techniques for Self Organizing Maps, ESANN´2005, European Symposium on Artificial Neural Networks, pp. 533-538.

Purves, D., Augustine, G., Fitzpatrik, D., Katz, L. C., LaMantia, A. S., McNamara, J. O., Willians, S. M. (2001), Neuroscience, 2nd ed., Sinauer Associates.

Rolls, E., T., (2007). An attactor network in the hippocampus: Theory and neurophysiology, Learning and Memory, vol. 14, pp. 714-731.

Thomaz, C. E., Giraldi, G. A. (2010). A new ranking method for Principal Components Analysis and its application to face image analysis, Image and Vision Computing, vol. 28, no. 6, pp. 902-913.

Thomaz, C. E., Amaral, V., Giraldi, G. A., Kitani, E. C., Sato, J. R., Gillies, D. F. (2009). A multi-linear discriminant analysis of 2D frontal face images, Proceedings of Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI), IEEE CS Press, 216-223.

Treves, A., Rolls, E. T. (1994). Computational analysis of the role of the Hippocampus in memory, Hippocapmus, vol. 4, pp. 374-391.

Ultsch, A. (2003). Maps for visualization of high-dimensional data space, in Proc. of Workshop on Self Organizing Maps, WSOM03, pp.225-230.

Vesanto, J. (1999). Self-organizing map in Matlab: the SOM Toolbox, Proc. of the Matlab DSP Conference, Finland, pp. 35-40.

Part 5

Perceptual Aspects of Face Recognition

13

The Effects of Right/Left Temporal Lobe Lesions on the Recognition of Familiar Faces

Guido Gainotti, Monica Ferraccioli and Camillo Marra
Center for Neuropsychological Research, Dept. of Neurosciences, Catholic University of Rome, Italy

1. Introduction

Recognition of familiar people can be based on three main sources of information: the face, the voice and the name, but the face has usually the greatest impact on this important social skill.

For this reason the study of 'prosopagnosia', considered as a form of visual agnosia, specifically concerning the recognition of familiar people through their face, has represented, since the proposal of this term by Bodamer (1947), the dominant and almost exclusive line of research in this field of inquiry. For the same reason, the first cognitive model that has tried to analyse the cognitive and subjective/behavioural stages involved in recognition and identification of familiar people is the Bruce et Young's (1986) model of familiar faces recognition. The first cognitive step of this model is the formation of a view independent structural description of a seen face, which can be compared with all the known faces stored in the Face Recognition Units (FRUs). A similar process was afterwards hypothesized for other sources of person recognition, such as voices and names, by several authors (Brédart et al., 1995; Burton et al., 1990; Burton et al., 1999; Valentine et al., 1996; Young & Burton, 1999), who assumed that the outcome of the corresponding perceptual processing could be matched with information stored in correlative Voice (VRUs) or Name Recognition Units (NRUs). According to all these models, the second step of the people identification process requires the convergence of information stored in these modality-specific units into person-identity nodes (PINs), allowing identification of a particular person and retrieval of the corresponding semantic (biographical) information. The PINs (or the accessed person-specific knowledge) could, in turn, activate the phonological codes underlying the production of the person's proper name.

In spite of the general similarities existing among the model proposed by Bruce and Young (1986) and those offered by following authors, there are also important differences among these models, which concern the locus in which familiarity feelings for the addressed person are generated and in which person-specific information is stored. As for the first point, the Bruce and Young (1986) model assumed that familiarity feelings are generated in the modality-specific recognition units where (for instance) the structural description of a seen face is compared to the familiar faces stored in the FRUs. On the contrary, in the Burton et al. (1990, 1999), Brédart et al. (1995) and Valentine et al. (1996) models, decisions about familiarity are taken at a supra-modal level, namely the PINs, where information from different modalities is combined in person identity nodes. Furthermore, the Bruce and Young's (1986) model assumes that PINs store semantic information, whereas Burton et al. (1990, 1999), Brédart et al. (1995) and Valentine et al. (1996) maintain that PINs do not store semantic information, but provide a modality-free gateway to a single semantic system, where information about people is stored in an amodal format.

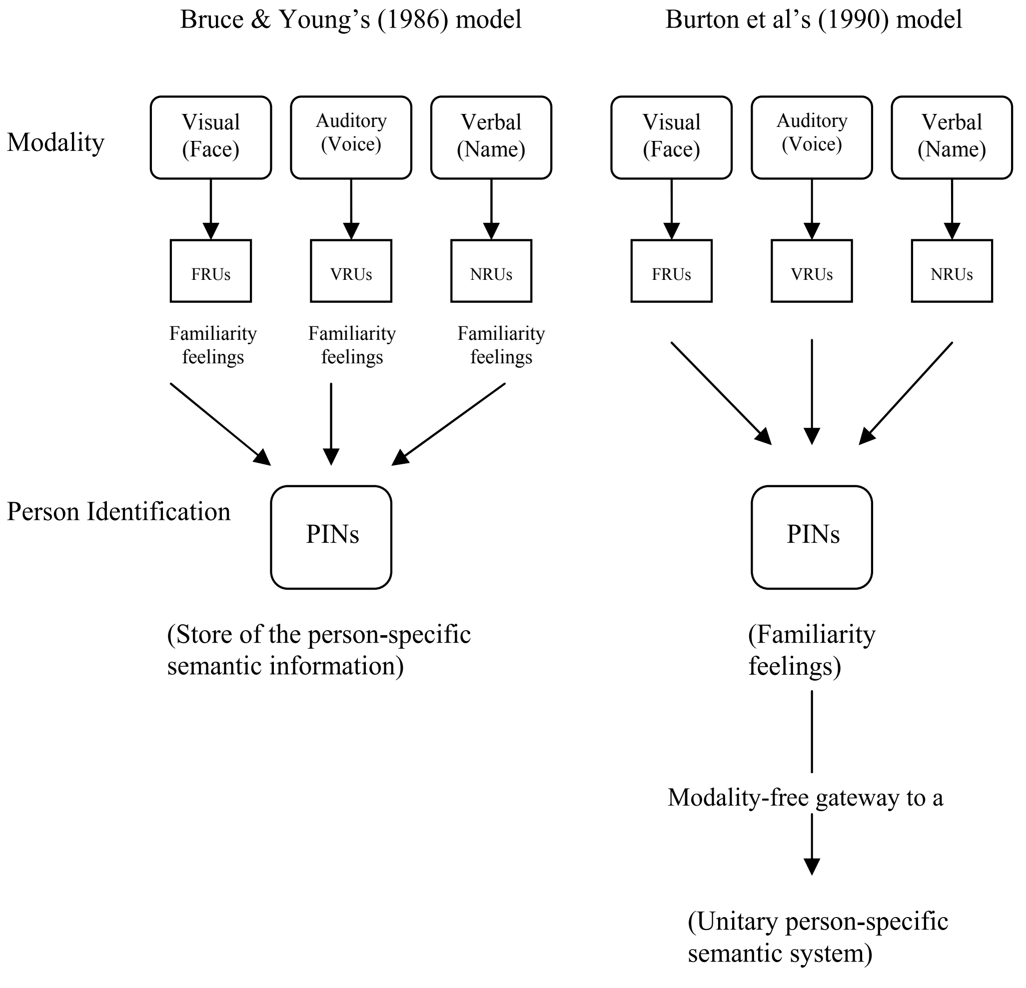


Fig. 1. Main differences between the original Bruce & Young (1986) model and the subsequent, more complex models of familiar people recognition.

Figure 1 reports in a schematic manner the main differences existing between the Bruce and Young (1986) model and the following models (e.g. Burton et al., 1990) with respect to the locus of generation of familiarity feelings and to the relations between PINs and person-specific semantic information.

But even the Burton et al.'s (1990, 1999), Brédart et al.'s (1995) and Valentine et al.'s (1996) statement that information about people is stored in an amodal format in the person-specific semantic system is open to controversies, because some authors (e.g. Snowden et al., 2004; Gainotti et al., 2003 and 2010; Gainotti, 2007a and 2011) maintain that this information is stored in a different format at the hemispheric level, i.e. in a multisensory/pictorial format in the right hemisphere and in a verbally-coded format in the left hemisphere.

Coming back from these general models and controversies to the dominance of face recognition in the identification of famous (or personally familiar) people, it is necessary to clearly distinguish 'prosopagnosia' (a defect of face/people recognition, restricted to the visual modality) from multimodal disorders in familiar people recognition, but this distinction has not been systematically made in the literature, because many patients who showed a multimodal disorder in familiar people recognition have been described as affected by prosopagnosia. This failure to distinguish prosopagnosia from multimodal familiar people recognition disorders is probably due to the dominance of faces in the recognition of known people and can be observed both in anatomo-clinical observations and in group studies. To stress the frequency with which patients affected by a multimodal people recognition defect have been considered as instances of prosopagnosia, and to underline the anatomical locus of lesion that subsume the multimodal forms of familiar people recognition disorders, we will limit ourselves to quote two classical anatomo-clinical observations, and two recent group studies of patients affected by right temporal variant of fronto-temporal degeneration (Hodges, 2000; Snowden et al., 1996; Tyrrel et al., 1990). The first anatomo-clinical observation, originally reported by Bouduresque et al. (1979) and afterwards studied in more details by Sergent & Poncet (1990), concerned a patient (M.me V.) who, after a Herpes Simples Encephalitis (HSE), complained of severe difficulties to recognize familiar people by face, in the absence of intellectual, memory, linguistic or visual defects.

The claim that M.me V's defective recognition of familiar people was not due to a subtle disorder of visual perception was documented by the fact that she showed no problems in the treatment of unknown faces during administration of a test similar to the Benton and Van Allen (1968) face matching test. Bouduresque et al. (1979) also noted that their patient repeatedly claimed being able to identify her family members, by hearing their voice, but that her performance was very poor when voice identification was investigated with an objective procedure. As for the lesion location, she showed on CT scan, a massive damage of the anterior parts of the right temporal lobe (RTL), in keeping with the usual localization of lesions caused by HSE (Gitelman et al., 2001).

The second anatomo-clinical observation concerned a man (LP) reported by De Renzi (1986) and De Renzi et al. (1987), who had also suffered from a previous HSE. This patient showed a widespread semantic disorder, but was unimpaired from the attentional, linguistic and visual point of view (including tests performed with unknown faces) and was considered as a case of 'associative prosopagnosia' (De Renzi et al.,1991), even if he also showed a multimodal defect of familiar people identification. As in the Bouduresque et al.'s (1979) patient, the anatomical lesion involved the antero-mesial parts of the temporal lobes, but this time with a left-sided prevalence. The two group studies relevant to the distinction between prosopagnosia and multimodal familiar people recognition disorders have been reported by Josephs et al. (2008) and by Chan et al. (2009). The first authors, starting from the description of a 'progressive prosopagnosia' in two SD patients (Evans et al., 1995; Joubert et al., 2003), tried to assess with the 'voxel-based morphometry' (VBM) the patterns of gray matter atrophy in SD patients with and without prosopagnosia. Results of this study showed that in SD patients with prosopagnosia atrophy mainly affects the antero-mesial parts of the RTL, whereas in those without prosopagnosia the lesion mainly involves the left temporal lobe. Chan et al. (2009), on the other hand, tried to identify the clinical profile associated with predominantly RTL atrophy and observed that prosopagnosia was reported by 60% of these patients. Note that, just as Bouduresque et al. (1979) and De Renzi (1986), also Josephs et al. (2008) were aware of having made an inappropriate use of the term 'prosopagnosia', because, contrary to what happens in real prosopagnosia, in their patients the person recognition defect was not confined to the visual (face) modality, but also concerned the voice and the name of the known person.

In any case, the just mentioned anatomo-clinical observations and the results of the group studies show that in patients with multimodal familiar people recognition disorders, the lesion lies outside the posterior temporo-occipital network involved in face processing. This network spans, indeed, from the inferior occipital areas ('Occipital Face Area/OFA of Gauthier et al., 2000) to the lateral portion of the mid-fusiform gyrus where is located the Face Fusiform Area (FFA/Kanwisher et al., 1997), whereas in patients showing a multimodal familiar people recognition disorder the lesion mainly involves the anterior parts of the TL.

2. Patterns of familiar people recognition disorders observed in patients with right and left anterior temporal lesions

Since in patients with multimodal familiar people recognition disorders the lesion can involve both the right (as in the Bouduresque et al.'s, 1979 patient) and the left anterior TL (as in the case reported by De Renzi, 1986), it became necessary to assess if familiar people recognition disorders are similar or different in patients with right and left TL lesions and to evaluate if these differences are relevant with respect to the controversies among theoretical models that we have summarized in the first part of the introduction. The first important contribution in this direction has been provided by Snowden et al. (2004), who have argued that a fine-grained investigation of the person-specific semantic impairment obtainable from visual (face) and verbal (name) stimuli in patients with degenerative lesions of the right and left TL could contribute: (a) to evaluate if different patterns of familiar people recognition disorders can be observed in patients with right and left TL lesions and (b) to clarify the debate concerning the 'unitary'(abstract-amodal) or 'non-unitary' (concrete-multisensory vs verbally-coded) format of semantic representations.

One of the cornerstones of this debate turns, in fact, around the hypothesis that dissociations in access to the semantic representation through the visual and the verbal modalities may be due to the 'perceptual affordances' of objects, namely to the perceptual features that could "suggest" which actions can be performed with those objects (Norman, 1988), allowing 'privileged accessibility' from vision to part of the semantic representation (Caramazza et al., 1990). Snowden et al. (2004) reasoned that, since people's faces and names are arbitrary, the study of person-specific semantic information obtainable from visual (face) and verbal (name) stimuli in patients with degenerative lesions of the right and left TL could represent a potentially valuable means of addressing the unitary vs non-unitary semantic systems controversy, ruling out the possible influence of the perceptual affordances of objects. Results of their study showed that semantic information accessed through face and name are different according to the prevalent side of atrophy. Semantic dementia patients with predominantly left temporal lobe atrophy identified faces better than names and performed better on the picture than on the word version of the semantic memory 'Pyramids and Palm

Trees' test (Howard & Patterson, 1992), whereas patients with right temporal lobe atrophy showed the opposite pattern of performance. These data were considered as incompatible with a unitary abstract model of semantic memory. A problem with this study consisted of the fact that, due to the rarity of this disease, the number of patients reported by Snowden et al. (2004) was relatively small and that paired comparisons between patients with right and left TL atrophy did, therefore, seldom reach significance. Since studies of semantic dementia patients typically involve single case studies, we thought that a strategy allowing to further check the Snowden et al.'s (2004) hypothesis could consist in systematically reviewing all the published individual cases of patients with a prevalent damage to the anterior parts of the right or left TL, in whom disorders of person recognition were on the foreground. Results of our review (Gainotti, 2007a) confirmed the findings of Snowden et al. (2004) and offered data provided of theoretical significance, since they were consistent with the Bruce and Young (1986) model, and inconsistent with the alternative models of Burton et al. (1990 and 1999), Bredart et al. (1995) and Valentine et al. (1996), with respect both to the locus of generation of familiarity feelings and to the functions of the PINs.

As for the first point, two main findings suggested that familiarity judgements were generated at the level of the modality-specific recognition units rather than at the PINs level. The first was that familiarity judgements were much more impaired in right than in left TL patients and the second that in patients with RTL lesions familiarity defects were modality-specific, concerning more famous faces than famous names. These findings suggested that familiarity feelings, being modality-specific, should be generated at the level of recognition units and in particular of the FRUs, that could be more represented in the RTL due to the major role played by the right hemisphere in face processing (De Renzi, 1986; De Renzi et al., 1994; Michel et al., 1989).

As for the second point, results of our review were inconsistent for two main reasons with the hypothesis assuming that PINs provide a modality-free gateway to a single system, where semantic information about people is stored in an amodal format. The first was that in patients with a RT damage the loss of person-specific semantic information, was clearly greater from face than from name. The second was that an important imbalance between the amount of person-specific information available from faces and names was also found in right and left TL patients who, showing intact or mildly impaired familiarity judgments, should have (according to the previously mentioned cognitive models) no defect at the PINs level.

A factor that could weaken the relevance of results obtained in our review, with respect to the models of familiar people recognition, was the Haslam et al.'s (2004) observation that in normal subjects both familiarity judgements and access to biographical information are more accurate in response to names than to faces. Now, since in studies considered in our review there were often no normative data, that considered separately familiarity judgement and biographical information obtainable from faces and from names, it was possible that the greater loss of familiarity feelings and of biographical information obtained from faces by RTL patients was in part due to this methodological pitfall. To check if differences observed in our review between patients with right and left anterior TL atrophy were due to the 'normal' differences about familiarity judgements and access to biographical information in response to names and faces reported by Haslam et al. (2004), we conducted a new research (Gainotti et al., 2010) in which we made use of two very well controlled normative studies, recently conducted by Bizzozero et al. (2005) and by Bizzozero, et al. (2007) on Italian participants. In the Bizzozero et al. (2005 and 2007) norms, the influence of age, education and gender on familiarity recognition and on person identification from faces and names had been controlled by means of covariate linear models, removing the effect of each variable and calculating from each subject's raw score the corresponding adjusted score. In a second step, the adjusted scores had been classified into five equivalent scores categories, ranging from 0 (= scores lower than the outer 5% inferential tolerance limits) to 4 (= scores higher than the median value of the sample). Furthermore, in the Bizzozero et al.'s (2005 and 2007) data, the semantic interviews aiming to assess the person identification were restricted to the faces and names correctly judged as familiar by the patient and therefore to people whose PINs should be unimpaired. Possible discrepancies between results obtained from faces and names with this procedure should, therefore, point to a different format of the semantic representation accessed through these different channels and could not be explained on the basis of methodological inconsistencies. The Bizzozero et al. (2005 and 2007) tests of face and name recognition and identification were administered to two patients, showing a selective mild difficulty of familiar people identification and naming due to a predominantly right and left TL atrophy, to see if the conclusions of our previous review were confirmed even with this highly controlled material. If the conclusions of our previous review were correct, the right TL patient should again show a greater impairment of familiarity feelings and of access to person-specific semantic information from faces, whereas, if results of our previous review were biased by a 'normal' advantage of names over faces we should observe in this patient no name advantage in familiarity judgment or access to person-specific semantic information. Data obtained in the right TL patient by Gainotti et al. (2010) confirmed the results of the previous review, since this patient showed: (1) a very impaired familiarity for faces, contrasting with a spared familiarity for names, indicating that familiarity judgments are generated at the level of modality-specific recognition units and not of a supramodal PIN; (2) a prevalent impairment of person-specific information available from faces rather than from names also for people that (being recognized as familiar from their face and name), should be normally represented at the PINs level.

3. The format of person-specific semantic information

Results of our previous review (Gainotti, 2007a) and behavioural data (Gainotti et al., 2010) obtained in a right TL patient, affected by a selective defect of familiar people identification, had a third implication, besides the fact of showing: (a) that familiarity feelings are generated at the level of modality-specific recognition units and (b) that PINs cannot be simply considered as a modality-free gateway to the person-specific semantic system, because they also suggested (c) that semantic information about people is stored in a different format at the level of the right and left temporal lobes. These data, therefore, confirmed the previous results of Snowden et al. (2004) who had shown that semantic dementia patients with predominantly right temporal lobe atrophy are more impaired with faces than with names, whereas patients with left TL atrophy show the opposite pattern of performance. Taken together, data obtained by Snowden et al. (2004) and our results strongly suggested that semantic representations of famous people are not represented in an 'amodal format' in both temporal lobes, but in a pictorial format in the right and in a verbal format in the left temporal lobe. Furthermore the Snowden et al.'s (2004) observations that semantic dementia patients with predominantly right temporal lobe atrophy perform worse on the picture than on the word version of the semantic memory 'Pyramids and Palm Trees' test (Howard & Patterson, 1992), suggest that this different format is not limited the semantic representation of famous people, but also extends to other conceptual domains. This suggestion is supported by both behavioural and neuroimaging data.

Behavioural data in line with the assumption of a prevalent involvement of the left TL in verbal and of the right TL in pictorial aspects of conceptual knowledge, have been obtained by Damasio et al. (1996 and 2004) and Tranel et al. (1997) in patients with focal lesions of the left and right temporal lobes. Damasio et al. (1996 and 2004) showed that defective retrieval of words denoting entities from various conceptual domains (such as famous people, animals or artefacts) was associated with lesions encroaching upon different parts of the left temporal lobe, whereas Tranel et al. (1997) demonstrated that impaired recognition of pictures representing persons, animals or tools was associated with lesions of the homologous areas of the right temporal lobe. According to these authors, both the left and the right temporal lobes play a mediational role in concept retrieval, but in the left hemisphere the activation of the "word" intermediary region promotes the retrieval of lexical knowledge required for word production, whereas in the right hemisphere the recollection of the perceptual properties of a given stimulus promotes the concrete sensorimotor representation of knowledge pertaining to that object.

Other behavioural data consistent with the hypothesis of a different involvement of the left and right temporal lobes in verbal and pictorial aspects of conceptual knowledge have been obtained in SD patients by Ikeda et al. (2006). These authors tested 10 SD patients and 10 matched controls on an object recognition task in which they were invited to choose (from a four-item array) the picture representing "the same thing" as an object picture that they had just inspected and attempted to name. The target in the response array was never physically identical to the studied picture but differed from it for various aspects. The patients whose structural brain imaging revealed major right-temporal atrophy were more impaired than those with an asymmetric pattern characterised by predominant left-sided atrophy, showing that they had a selective defect in the retrieval of the pictorial properties of objects.

3.1 Correlations between cognitive and neuroimaging data, studying person related and conceptual knowledge with verbal and pictorial material

A different role of the right and left ATL has been documented by functional neuroimaging investigations that have taken into account different aspects of familiar people recognition or of conceptual knowledge. Thus, several authors have documented a prevalent activation of the right temporal lobe for famous faces (Ishai et al., 2005), for famous - contrasted with newly learned - faces (Leveroni et al., 2000), during association between faces and person-specific semantic information (Tsukiura et al., 2008) or during a semantic categorization task of famous faces (Brambati et al., 2010). On the other hand, Tsukiura et al. (2008) have shown that the left ATL may mediate associations between names and person-related semantic information and similar results have been obtained by Brambati et al. (2010), who have shown an increased activation of the left anterior TL when subjects were asked to determine whether a stimulus photograph matched with the label of a profession category. Consistent with these results obtained studying different aspects of familiar people recognition are results of investigations which have assessed the correlations between neuroimaging data and conceptual impairment in the verbal and pictorial modality.

Thus, Acres et al. (2009) and Butler et al. (2009), evaluating conceptual knowledge with verbal and pictorial material, and the severity of temporal lobe atrophy with voxel-based measures, have shown that verbal semantic defects are on the foreground when the atrophy mainly affects the left temporal lobe, whereas non-verbal conceptual disorders tend to prevail when the right inferior temporal structures are preferentially disrupted. Similar data have been recently obtained by Mion et al. (2010), who examined with FDG-PET the neural correlates of verbal and non-verbal semantic measures in SD. The semantic verbal task was a picture naming task, whereas the non-verbal semantic task was the 'Camel and Cactus test' (Bozeat et al., 2000), similar to the pictorial version of the semantic memory 'Pyramids and Palm Trees' test (Howard & Patterson, 1992). Regions of interest (ROIs) were the left and right anterior fusiform gyri and the temporal poles. The left anterior fusiform activity predicted performance on the verbal semantic tasks, whereas the right anterior fusiform metabolism predicted performance on the non-verbal semantic task. Furthermore, an additional behavioural study, performed on a wider cohort of SD patients, confirmed that patients with more extensive right TL atrophy are significantly more impaired on tests of non-verbal semantics.

4. Concluding remarks on the implications of these data for models of familiar people recognition

We will conclude this chapter by reporting in a schematic manner in Figure 2 the implications that data concerning: (a) the patterns of familiar people recognition shown by right and left TL patients and (b) the different format of (person-specific or conceptual) knowledge represented in the right and left temporal lobes could have for models of familiar people recognition.

Two main conclusions are suggested by results of investigations surveyed in the previous sections of this chapter and summarized in Figure 2. The first is that results concerning (a) the locus of generation of familiarity feelings, (b) the relationships between PINs and person-specific semantic knowledge and (c) the format of this kind of knowledge are much more consistent with the simpler and older model of Bruce and Young's (1986) than with the more recent and complex models of familiar people recognition proposed by Burton et al. (1990, 1999), Brédart et al. (1995) and Valentine et al. (1996). The second is that, to give a plausible account of data obtained in brain-damaged patients, these models cannot ignore some basic inter-hemispheric differences, such as the critical role of the right hemisphere in the generation of face familiarity feelings and the different format of person-specific semantic knowledge at the level of the right and left hemisphere.

Both these issues have been thoroughly discussed in previous reviews (Gainotti, 2007b and 2011) and will be only shortly considered here as two sides of a unitary phenomenon, namely the more primitive (sensori-motor) organization of the right hemisphere and the more complex, language-mediated organization of the left hemisphere.

Within this context, it is possible to assume that the early familiarity feelings may be automatically elicited through a right-hemisphere subcortical route, allowing a first, unconscious, global recognition of familiar faces and fostering the subsequent distinction of known faces (unconsciously detected) from unfamiliar faces.

From the theoretical point of view, this possibility has been suggested by De Haan et al. (1991) with the following expression: 'When a FRU is activated it will signal that the face is familiar and instigate the retrieval of semantic knowledge concerning the bearer of the face'.

Within the same framework, it seems logical to assume that, far from being represented in an abstract amodal format, every kind of person-specific and conceptual knowledge may consist of a more primitive perceptual-motor knowledge (more represented in the right hemisphere) and of a more complex language-mediated and language-structured knowledge, more represented in the left hemisphere. From this point of view, the prevalent impairment of person-specific information available from faces, that we have documented in patients with a right TL atrophy, could be considered as the most impressive manifestation of the disruption of the multi-sensory/pictorial knowledge that seems typical of the right hemisphere.

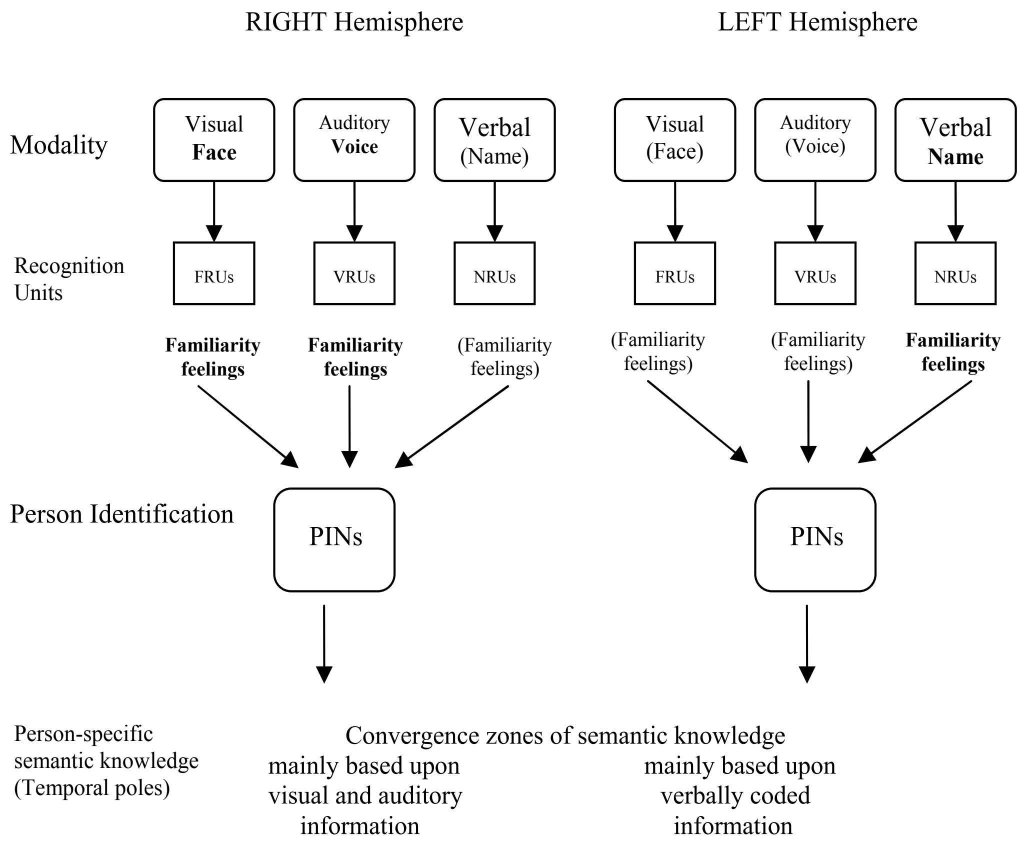


Fig. 2. Main differences between the familiar people recognition disorders shown by patients with right and left anterior temporal lesions. In bold are reported the modalities of people recognition and the corresponding familiarity feelings that are more represented at the level of the right and left hemisphere.

5. References

Acres, K., Taylor, K.I., Moss, H.E., Stamatakis, E.A. & Tyler, L.K. (2009). Complementary hemispheric asymmetries in object naming and recognition: a voxel-based correlational study. *Neuropsychologia*, Vol.47, No.8-9, (July 2009), pp. 836-843, ISSN 0028-3932

Benton, A.L. & Van Allen, M.W. (1968). Impairment in facial recognition in patients with cerebral disease. *Cortex*, Vol.4, No.4, pp. 344-358, ISSN: 0010-9452

Bizzozero, I., Ferrari, F., Bozzoli, S., Saetti, M.C. & Spinnler, H. (2005). Who is who: Italian norms for visual recognition and identification of celebrities. *Neurological Sciences,* Vol.26, No.2, (June 2005), pp. 95-107, ISSN 15901874

Bizzozero, I., Lucchelli, F., Bozzoli, S., Saetti, M.C. & Spinnler, H. (2007). "What do you know about Ho Chi Minh?" Italian norms of proper name comprehension. *Neurological Sciences,* Vol.28, No.1, (March 2007), pp. 16-30, ISSN 1590-1874

Bodamer, J. (1947). Die Prosop-Agnosie. *Archiv für Psychiatrie und Nervenkrankheiten*, Vol. 179, No. (1-2), pp. 6-53

Boudouresques, J., Poncet, M., Cherif, A.A. & Balzamo, M. (1979). Agnosia for faces: evidence of functional disorganization of a certain type of recognition of objects in the physical world. *Bulletin de l'Académie Nationale de Medecine*, Vol.163, No.7, (October 1979), pp. 695-702

Bozeat, S., Lambon Ralph, M.A., Patterson, K., Garrard, P. & Hodges, J.R. (2000). Non-verbal semantic impairment in semantic dementia. *Neuropsychologia,* Vol.38, No.9, pp. 1207–15, ISSN 0028-3932

Brambati SM, Benoit S, Monetta L, Belleville S, Joubert S. (2010) The role of the left anterior temporal lobe in the semantic processing of famous faces. *Neuroimage*, Vol.1, No.53(2), (November 2010), pp. 674-681, ISSN 1053-8119

Brédart, S., Valentine, T., Caldor, A. & Gassi, L. (1995). An interactive activation model of face naming. *The Quarterly Journal of Experimental Psychology,* Vol.48, No.2, (May 1995), pp. 466-486, ISSN 0272-4987

Bruce, V. & Young A.W. (1986). Understanding face recognition. *British Journal of Psychology,* Vol.77, No.3, (August 1986), pp. 305-327

Burton, A.M., Bruce, V. & Hancock, P.J.B. (1999). From pixels to people: a model of familiar face recognition. *Cognitive Science,* Vol.23, No.1, (January 1999), pp. 1-31

Burton, A.M., Bruce, V. & Johnston, R.A., 1990. Understanding face recognition with an interactive activation model. *British Journal of Psychology*, Vol.81, No.3, (August 1990), pp. 361-381, ISSN 2044-8295

Butler, C.R., Bramati, S.M., Miller, B.L. & Gorno-Tempini, M.L. (2009). The neural correlates of verbal and nonverbal semantic processing deficits in neurodegenerative disease. *Cognitive and Behavioral Neurology*, Vol.22, No.2, (June 2009), pp. 73-80, ISSN 1543-3633

Caramazza, A., Hillis, A., Rapp, B.C. & Romani, C. (1990). The multiple semantic hypothesis: Multiple confusions? *Cognitive Neuropsychology*, Vol.7, No.3, pp. 161-189, ISSN 1464-0627

Chan, D., Anderson, V., Pijnenburg, Y., Whitwell, J., Barnes, J., Scahill, R., Stevens, J.M., Barkhof, F., Scheltens, P., Rossor, M.N. & Fox, N.C. (2009). The clinical profile of right temporal lobe atrophy. *Brain,* Vol.132(Pt 5), (May 2009), pp.1287-98, ISSN 0006-8950

Damasio, H., Grabowski, T.J., Tranel, D., Hichwa & R.D., Damasio, A.R. (1996). A neural basis for lexical retrieval. *Nature*, Vol.11, No.380(6574), (April 1996), pp. 499–505 ISSN: 0028-0836

Damasio, H., Tranel, D., Grabowski, T.J., Adolphs, R. & Damasio, A.R. (2004). Neural systems behind word and concept retrieval. *Cognition*, Vol.92, No. 1-2, (May-June 2004), pp. 179-229, ISSN 0010-0277

De Haan, E.H., Young, A.W., Newcombe, F., (1991). A dissociation between sense of familiarity and access to semantic information concerning familiar people. *European Journal of Cognitive Psychology*, Vol.3, No.1, 51-67, ISSN 2044-5911

De Renzi, E. (1986). Current issues in prosopagnosia. In: *Aspects of face processing*, Ellis, H.D., Jeeves, M.A., Newcombe, F. and Young, A.W. (Eds.) 243-252, NATO ASI series, Martinus Nijhoff, Dordrecht

De Renzi, E., Liotti, M., & Nichelli, P. (1987). Semantic amnesia with preservation of autobiographic memory: A case report. *Cortex*, Vol.23, No.4, (December 1987), pp.575-597, ISSN: 0010-9452

De Renzi, E., Faglioni, P., Grossi, D. & Nichelli, P. (1991). Apperceptive and associative forms of prosopagnosia. *Cortex*, Vol.27, No.2, (June 1991), pp. 213-221, ISSN: 0010-9452

De Renzi, E., Perani, D., Carlesimo, G.A., Silveri, M.C. & Fazio, F. (1994). Prosopagnosia can be associated with damage confined to the right hemisphere. An MRI and PET study and a review of the literature. *Neuropsychologia*, Vol.32, No.8, (August 1994), pp. 893-902, ISSN 0028-3932

Evans, J.J., Heggs, A.J., Antoun, N. & Hodges J.R. (1995). Progressive prosopagnosia associated with selective right temporal lobe atrophy. A new syndrome? *Brain*, Vol.118, Pt.1, (February 1995), pp. 1-13, ISSN 0006-8950

Gainotti, G. (2007a). Different patterns of famous people recognition disorders in patients with right and left anterior temporal lesions: A systematic review. *Neuropsychologia*, Vol.45, No.8, (April 2007), pp. 1591-1607, ISSN 0028-3932

Gainotti, G. (2007b). Face familiarity feelings, the right temporal lobe and the possible underlying neural mechanisms. *Brain Research Reviews*, Vol.56, No.1, (November 2007), pp. 214-235, ISSN 0165-0173

Gainotti, G. (2011). The format of conceptual representations disrupted in semantic dementia. (Submitted)

Gainotti, G., Barbier, A. & Marra, C. (2003). Slowly progressive defect in recognition of familiar people in a patient with right anterior temporal atrophy. *Brain,* Vol.126, Pt.4, (April 2003), pp. 792-803, ISSN 0006-8950

Gainotti, G., Ferraccioli, M. & Marra C. (2010). The relation between person identity nodes, familiarity judgment and biographical information. Evidence from two patients with right and left anterior temporal atrophy. *Brain Research*, Vol.1307, No.11, (January 2010), pp. 103-114, ISSN: 00068993

Gauthier, I., Tarr, M.J., Moylan, J., Skudlarski, P., Gore, J.C. & Anderson, A.W. (2000). The fusiform ''face area'' is part of a network that processes faces at the individual level. *Journal of Cognitive Neuroscience*, Vol.12, No.3, (May 2000), pp. 495–504, ISSN 0898929X

Gitelman, D., Ashburner, J., Friston, K., Tyler, L. K. & Price, C. (2001). Voxel-based morphometry of herpes simplex encephalitis. *Neuroimage*, Vol.13, No.4, (April 2001), pp. 623–631, ISSN 1053-8119

Haslam, C., Kay, J., Hanley, J.R. & Lyons, F. (2004). Biographical knowledge: modality specific or modality-neutral? *Cortex*, Vol.40, No.3, (June 2004), pp. 451-466, ISSN 0010-9452

Hodges J.R. (2000) Pick's disease: It's relationship to semantic dementia, progressive aphasia and frontotemporal dementia. In: *Dementia*, J. O'Brien, D. Ames, A. Burns, (Ed.), 741–758, Arnold, London

Howard, D. & Patterson, K. (1992). Pyramids and Palm Trees: access from pictures and words. Thames Valley Test Company, Bury St Edmunds, UK

Ikeda, M., Patterson, K., Graham, K.S., Lambon Ralph, M.A. & Hodges, J.R. (2006). A horse of a different colour: do patients with semantic dementia recognise different versions of the same object as the same? *Neuropsychologia,* Vol.44, No.4, pp. 566-575, ISSN 0028-3932

Ishai, A., Schmidt, C.,F. & Boesinger, P. (2005). Face perception is mediated by a distributed cortical network. *Brain Research Bulletin,* Vol.67, No.1-2, (September 2005), pp. 87-93, ISSN 0361-9230

Josephs, K.A., Whitwell, J.L., Vemuri, P., Senjem, M.L., Boeve, B.F., Knopman, D.S., Smith, G.E., Ivnik, R.J., Petersen, R.C. & Jack, C.R. Jr. (2008). The anatomic correlate of prosopagnosia in semantic dementia. *Neurology*, Vol.71, No.20, (November 2008), pp. 1628-33, ISSN 0028-3878

Joubert, S., Felician, O., Barbeau, E., Sontheimer, A., Guedj, E., Caccaldi, M. & Poncet, M. (2003). Impaired configurational processing in a case of progressive psosopagnosia associated with predominant right temporal lobe atrophy. *Brain*, Vol.126, Pt.11, (November 2003), pp. 2357-50, ISSN 0006-8950

Kanwisher, N., McDermott, J. & Chun, M.M. (1997). The fusiform face area: a module in human extrastriate cortex specialized for face perception. *Journal of Neuroscience*, Vol.17, No.11, (June 1997), pp. 4302–431, ISSN 0270-6474

Leveroni, C.L., Seidenberg M., Mayer A.R., Mead L.A., Binder, J.R.,Rao, S.M., 2000. Neural systems underlying the recognition of familiar and newly learned faces. *Journal of Neuroscience*, Vol.20, No.2, (January 2000), pp. 878-886, ISSN 0270-6474

Michel, F., Poncet, M. & Signoret, J.L. (1989). Les lésions responsables de la prosopagnosie sont-elles toujours bilatérales? *Revue Neurologique*, Vol.146, Paris, pp. 764-770, ISSN 0035-3787

Mion, M., Patterson, K., Acosta-Cabronero, J., Pengas, G., Izquierdo-Garcia, D., Hong, Y.T., Fryer, T.D., Williams, G.B., Hodges, J.R. & Nestor, P.J. (2010). What the left and right anterior fusiform gyri tell us about semantic memory. *Brain*, Vol.133, No.11, (November 2010), pp. 3256-3268, ISSN 0006-8950

Norman D. A. (1988). The Psychology of Everyday Things. New York, Basic Books.

Sergent J. & Poncet M. (1990). From covert to overt recognition of faces in a prosopagnosic patient. *Brain*, Vol.113, Pt.4, (August 1990), pp. 989-1004, ISSN 0006-8950

Snowden, J.S., Neary D. & Mann D.M.A. (1996) *Fronto-temporal lobar degeneration: fronto-temporal dementia, progressive aphasia,semantic dementia.* Churchill Livingstone, New York

Snowden, J.S., Thompson, J.C., Neary, D. (2004). Knowledge of famous faces and names in semantic dementia. *Brain*, Vol.127, Pt.4, (April 2004), pp. 860-872, ISSN 0006-8950

Tranel, D., Damasio, H. & Damasio, AR. (1997). A neural basis for the retrieval of conceptual knowledge. *Neuropsychologia*, Vol.35, No.10, (October 1997), pp. 1319-1327, ISSN 0028-3932

Tsukiura, T., Suzuki, C., ShigemuneY. & Mokizuki-Kawai, H. (2008). Differential contribution of the anterior temporal and medio temporal lobe to the retrieval of memory for person identity information. *Human Brain Mapping*, Vol.29, No.12, (December 2008), pp. 1343-54, ISSN 1065-9471

Tyrrel, P.J., Warrington, E.K., Frackowiak, R.S.J. & Rossor, M.N. (1990). Progressive degeneration of the right temporal lobe studies with positron emission tomography. *Journal of Neurology, Neurosurgery and Psychiatry*, Vol.53, No.12, (December 1990), pp. 1046-1050, ISSN: 00223050

Valentine, T., Brennen, T., Bredart, S., (1996). *The cognitive psychology of proper names.* Routledge, London

Young, A.W. & Burton, A.M. (1999). Simulating face recognition: implications for modelling cognition. *Cognitive Neuropsychology*, Vol.16, No.1, (February 1999), pp. 1-48, ISSN: 1464-0627

Edited by Peter Corcoran

As a baby, one of our earliest stimuli is that of human faces. We rapidly learn to identify, characterize and eventually distinguish those who are near and dear to us. We accept face recognition later as an everyday ability. We realize the complexity of the underlying problem only when we attempt to duplicate this skill in a computer vision system. This book is arranged around a number of clustered themes covering different aspects of face recognition. The first section presents an architecture for face recognition based on Hidden Markov Models; it is followed by an article on coding methods. The next section is devoted to 3D methods of face recognition and is followed by a section covering various aspects and techniques in video. Next short section is devoted to the characterization and detection of features in faces. Finally, you can find an article on the human perception of faces and how different neurological or psychological disorders can affect this.

Photo by dimitris_k / iStock

IntechOpen

ISBN 978-953-51-4471-7