State of the Art in Face Recognition

Edited by Mario I. Chacon M.

State of the Art in Face Recognition

Edited by

Dr. Mario I. Chacon M.

State of the Art in Face Recognition
http://dx.doi.org/10.5772/104
Edited by Dr. Mario I. Chacon M.

Notice

Statements and opinions expressed in the chapters are these of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published chapters. The publisher assumes no responsibility for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained in the book.

We are IntechOpen,
the world's leading publisher of Open Access books
Built by scientists, for scientists

4,400+

Open access books available

118,000+

International authors and editors

130M+

Downloads

151

Countries delivered to

Our authors are among the

Top 1%

most cited scientists

12.2%

Contributors from top 500 universities

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Preface

Notwithstanding the tremendous effort to solve the face recognition problem, it is not possible yet to design a face recognition system with a potential close to human performance. New computer vision and pattern recognition approaches need to be investigated. Even new knowledge and perspectives from different fields like, psychology and neuroscience must be incorporated into the current field of face recognition to design a robust face recognition system. Indeed, many more efforts are required to end up with a human like face recognition system. This book tries to make an effort to reduce the gap between the previous face recognition research state and the future state. Also, the purpose of the book is to present the reader with cutting edge research on the face recognition field. Besides, the book includes recent research works from different world research groups, providing a rich diversity of approaches to the face recognition problem.

This book consists of 12 chapters. The material covered in these chapters presents new advances on computer vision and pattern recognition approaches, as well as new knowledge and perspectives from different fields like, psychology and neuroscience. The chapters are organized into three groups according to their main topic. The first group focuses on classification, feature spaces, and subspaces for face recognition, Chapters 1 to 5. The second group addresses the no trivial techniques of face recognition based on holographic, 3D methods and low resolution video, covered in Chapters 6 to 9. Chapters 10 to 12 cover the third group related to human visual perception aspects on face recognition.

Chapter 1 describes the achievement and perspective trends related to nearest feature classification for face recognition. The authors explain the family of nearest feature classifiers and their modified and extended versions. Among other points they provide a discussion on alternatives of the nearest feature classifiers, indicating which issues are still susceptible to be improved. The authors describe three approaches for generalizing dissimilarity representations, and they include their proposal for generalizing them by using feature lines and feature planes.

Chapter 2 addresses recent subspace methods for face recognition including: singularity, regularization, and robustness. They start dealing with the singularity problem, and the authors propose a fast feature extraction technique, Bi-Directional PCA plus LDA (BDPCA+LDA), which performs LDA in the BDPCA subspace. Then, the authors presents an alternative to alleviate the over-fitting to the training set, proposing a post-processing approach on discriminant vectors, and theoretically demonstrates its relationship with the image Euclidean distance method (IMED). Finally, the authors describe an iteratively reweighted fitting of the Eigenfaces method (IRF-Eigenfaces), which first defines a generalized objective function and then uses the iteratively reweighted least-squares (IRLS)

fitting algorithm to extract the feature vector by minimizing the generalized objective function.

A multi-stage classifier for face recognition undertaken by coarse-to-fine strategy is covered in Chapter 3. The chapter includes a brief description of the DCT and PCA feature extraction methods, as well as the proposed coarse to fine stages, OAA, OAO, and multi-stage classifiers.

In Chapter 4, the authors propose a method to improve the face image quality by using photometric normalization techniques. This technique based on Histogram Equalization and Homomorphic Filtering normalizes the illumination variation of the face image. The face recognition system is based on ANN with features extracted with the PCA method.

The aim of Chapter 5 is to demonstrate the following points: how the feature extraction part is evolved by IPCA and Chunk IPCA, how both feature extraction part and classifier are learned incrementally on an ongoing basis, how an adaptive face recognition system is constructed and how it is effective. The chapter also explains two classifiers based on ANN, the Resource Allocating Network (RAN) and its variant model called RAN-LTM.

Chapter 6 introduces a faster face recognition system based on a holographic optical disc system named FARCO 2.0. The concept of the optical parallel correlation system for facial recognition and the dedicated algorithm are described in the chapter. The chapter presents a faster correlation engine for face, image and video data using optical correlation, and an online face recognition system based on phase information.

The first 3D technique for face recognition is covered in Chapter 7. The authors describe a 3D face mesh modeling for 3D face recognition. The purpose of the authors is to show a model-based paradigm that represents the 3D facial data of an individual by a deformed 3D mesh model useful for face recognition application.

Continuing with 3D methods, the occlusion problem in face recognition system is handled in Chapter 8. In this chapter the authors describe their approach, a full automatic recognition pipeline based on 3D imaging. They take advantage of the 3D data to solve the occlusion problem because it has depth information available.

Chapter 9 presents a model-based approach for simultaneous tracking and increasing super-resolution of known object types in low resolution video. The approach is also based on a 3D mask. The 3D mask allows estimating translation and rotation parameters between two frames which is equivalent to calculating a dense sub-pixel accurate optical flow field and subsequent warping into a reference coordinate system.

The material covered in Chapter 10 is aimed to show how joint knowledge from human face recognition and unsupervised systems may provide a robust alternative compared with other approaches. The chapter includes a detailed description of how low resolution features can be combined with an unsupervised ANN for face recognition.

Chapter 11 addresses the issue of gender classification by information fusion of hair and face. Unlike most face recognition systems, the proposed method in this chapter considers the important role of hair features in gender classification. The chapter presents a study of hair feature extraction and the combination of hair classifier and face classifier. The authors show that the key point of classifier fusion is to determine how classifiers interact with each other. The fusion information method used is based on the fuzzy integral.

Last but not at least, a challenging issue on face recognition is faced in Chapter 12, emotion modeling and facial affect recognition in human-computer and human-robot interaction. In this chapter the authors present a review of prevalent psychology theories on

emotion with the purpose to disambiguate their terminology and identify the fitting computational models that can allow affective interactions in the desired environments.

It is our interest, editors and chapter authors that this book contributes to a fast and deep development on the challenging filed of face recognition systems.

We also expect the reader really finds this book both helpful and promising.

January 2009

Editor

Dr. Mario I. Chacon M.
Chihuahua Institute of Technology,
Mexico

Contents

Trends in Nearest Feature Classification for Face Recognition – Achievements and Perspectives

Mauricio Orozco-Alzate and César Germán Castellanos-Domínguez
Universidad Nacional de Colombia Sede Manizales
Colombia

1. Introduction

Face recognition has become one of the most intensively investigated topics in biometrics. Recent and comprehensive surveys found in the literature, such as (Zhao et al., 2003; Ruiz-del Solar & Navarrete, 2005; Delac & Grgic, 2007), provide a good indication of how active are the research activities in this area. Likewise in other fields in pattern recognition, the identification of faces has been addressed from different approaches according to the chosen representation and the design of the classification method. Over the past two decades, industrial interests and research efforts in face recognition have been motivated by a wide range of potential applications such identification, verification, posture/gesture recognizers and intelligent multimodal systems. Unfortunately, counter effects are unavoidable when there is a heavily increased interest in a small research area. For the particular case of face recognition, most of the consequences were pointed out by three editors of the well-known *Pattern Recognition Letters* journal. The following effects on the publication of results were discussed by Duin et al. (2006):

1. The number of studies in face recognition is exploding and always increasing. Some of those studies are rather obvious and straightforward.
2. Many of the submitted papers have only a minor significance or low citation value. As a result, journals receive piles of highly overlapping related papers.
3. Results are not always comparable, even though the same data sets are used. This is due to the use of different or inconsistent experimental methodologies.

A par excellence example of the situation described above is the overwhelming interest in linear dimensionality reduction, especially in the so-called small sample size (SSS) case. It is one of the most busy study fields on pixel-based face recognition. Indeed, the SSS problem is almost always present on pixel-based problems due to the considerable difference between dimensions and the number of available examples. In spite of that apparent justification, most of the published works in this matter are minor contributions or old ideas phrased in a slightly different way. Of course, there are good exceptions, see e.g. (Nhat & Lee, 2007; Zhao & Yuen, 2007; Liu et al., 2007). Our discussion here should not be interpreted as an attack to authors interested in dimensionality reduction for face recognition; conversely, we just want to explain why we prefer to focus in subsequent stages of the pattern recognition system instead of in dimensionality reduction. In our opinion, making a significant contribution in

linear dimensionality reduction is becoming more and more difficult since techniques have reached a well-established and satisfactory level. In contrast, we consider that there are more open issues in previous and subsequent stages to representation such as preprocessing and classification.

At the end of the nineties, a seminal paper published by Li and Lu (1999) introduced the concept of feature line. It consists in an extension of the classification capability of the nearest neighbor method by generalizing two points belonging to the same class through a line passing by those two points (Li, 2008). Such a line is called *feature line*. In (Li & Lu, 1998), it was suggested that the improvement gained by using feature lines is due to their faculty to expand the representational ability of the available feature points, accounting for new conditions not represented by the original set. Such an improvement was especially observed when the cardinality of the training set (sample size) per class is small. Consequently, the nearest feature line method constitutes an alternative approach to attack the SSS problem without using linear dimensionality reduction methods. In fact, the dimensionality is increased since the number of feature lines depends combinatorially on the number of training points or objects per class. Soon later, a number of studies for improving the concept of feature lines were reported. A family of extensions of the nearest feature line classifier appeared, mainly encompassing the nearest feature plane classifier, the nearest feature space classifier and several modified versions such as the rectified nearest feature line segment and the genetic nearest feature plane. In addition, an alternative classification scheme to extend the dissimilarity-based paradigm to nearest feature classification was recently proposed.

In the remaining part of this chapter, we will explain in detail that family of nearest feature classifiers as well as their modified and extended versions. Our exposition is organized as follows. In Section 2, a literature review of prototype-based classification is given. It ranges from the classical nearest neighbor classifier to the nearest feature space classifier, reviewing also modifications of the distance measure and several editing and condensing methods. In addition, we provide a detailed discussion on the modified versions of the nearest feature classifiers, mentioning which issues are still susceptible to be improved. The framework of dissimilarity representations and dissimilarity-based classification is presented in Section 3. We present three approaches for generalizing dissimilarity representations, including our own proposal for generalizing them by using feature lines and feature planes. Finally, a general discussion, overall conclusions and opportunities for future work are given in Section 4.

2. Prototype-based face recognition

Several taxonomies for pattern classification methods have been proposed. For instance, according to the chosen representation, there is a dichotomy between structural and statistical pattern recognition (Bunke & Sanfeliu, 1990; Jain et al., 2000; Pękalska & Duin, 2005a). According to the criterion to make the decision, classification approaches are divided into density-based and distance-based methods (Duda et al., 2001). Similarly, another commonly-stated division separates parametric and nonparametric methods. This last dichotomy is important for our discussion on prototype based face recognition.

Parametric methods include discriminant functions or decision boundaries with a predefined form, e.g. hyperplanes, for which a number of unknown parameters are estimated and plugged into the model. In contrast, nonparametric methods do not pre-

define a model for the decision boundary; conversely, such a boundary is directly constructed from the training data or generated by an estimation of the density function. The first type of nonparametric approaches encompasses the prototype based classifiers; a typical example of the second type is the Parzen window method.

Prototype based classifiers share the principle of keeping copies of training vectors in memory and constructing a decision boundary according to the distances between the stored prototypes and the query objects to be classified (Laaksonen, 1997). Either the whole training feature vectors are retained or a representative subset of them is extracted to be prototypes. Moreover, those prototypes or training objects can be used to generate new representative objects which were not originally included in the training set. Representations of new objects are not restricted to feature points; they might be lines, planes or even other functions or models based on the prototypes such as clusters or hidden Markov models (HMMs).

2.1 The nearest neighbor classifier

The simplest nonparametric method for classification should be considered k-NN (Cover & Hart, 1967). Its first derivation and fundamental theoretical properties gave origin to an entire family of classification methods, see Fig. 1. This rule classifies x by assigning it the class label $\hat{c}$ most frequently represented among the k nearest prototypes; i.e., by finding the k neighbors with the minimum distances between x and all prototype feature points $\{x_{ci}, 1 \le c \le C, 1 \le i \le n_c\}$. For k=1, the rule can be written as follows:

$$d(x, x_{\hat{c}i}) = \min_{1 \le c \le C;\ 1 \le i \le n_c} d(x, x_{ci}), \tag{1}$$

where $d(x,x_{ci}) = \| x - x_{ci} \|$ is usually the Euclidean norm. In this case, the number of distance calculations is $n = \sum_{c=1}^{C} n_c$.

The k-NN method has been successfully used in a considerable variety of applications and has an optimal asymptotical behavior in the Bayes sense (Devroye et al., 1996); nonetheless, it requires a significant amount of storage and computational effort. Such a problem can be partly solved by using the condensed nearest neighbor rule (CNN) (Hart, 1968). In addition, the k-NN classifier suffers of a potential loss of accuracy when a small set of prototypes is available. To overcome this shortcoming, many variations of the k-NN method were developed, including the so-called nearest feature classifiers. Such methods derived from the original k-NN rule can be organized in a family of prototype-based classifiers as shown in Fig. 1.

2.2 Adaptive distance measures for the nearest neighbor rule

In order to identify the nearest neighbor, a distance measure has to be defined. Typically, a Euclidean distance is assumed by default. The use of other Minkowski distances such as Manhattan and Chebyshev is also convenient, not just for interpretability but also for computational convenience. In spite of the asymptotical optimality of the k-NN rule, we never have access to an unlimited number of samples. Consequently, the performance of the k-NN rule is always influenced by the chosen metric.

Several methods for locally adapting the distance measure have been proposed. Such an adaptation is probabilistically interpreted as an attempt to produce a neighborhood with an
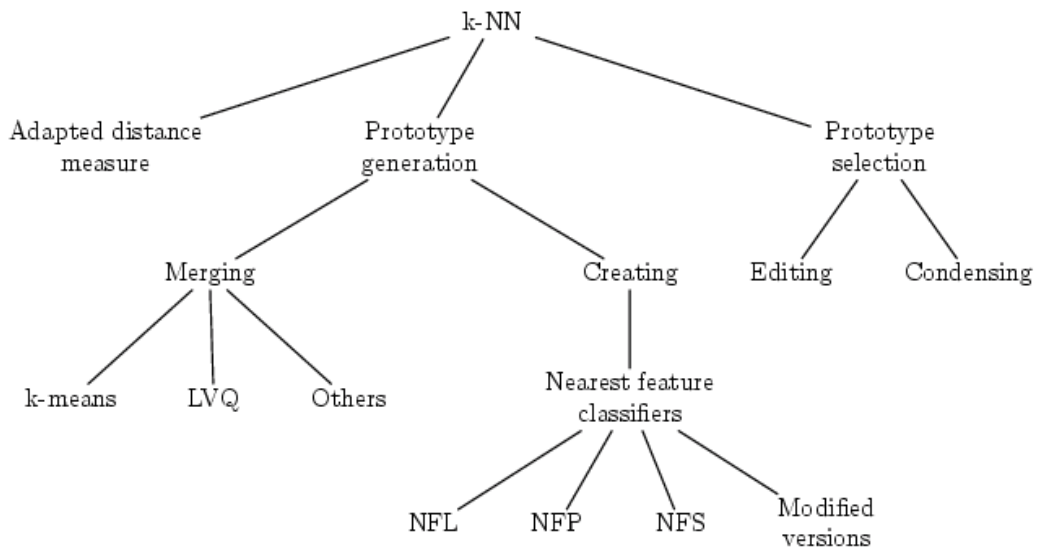
Fig. 1. Family of prototype-based classifiers.

a posteriori probability approximately constant (Wang et al., 2007). Among the methods aimed to local adaptation, the following must be mentioned:

a. The flexible or customized metric developed by Friedman (1994). Such a metric makes use of the information about the relative relevance of each feature. As a result, a new method is generated as a hybrid between the original k-NN rule and the tree-structured recursive partitioning techniques.

b. The adaptive metric method by Domeniconi et al. (2002). They use a χ^2 distance analysis to compute a flexible metric for producing neighborhoods that are adaptive to query locations. As a result, neighborhoods are constricted along the most relevant features and elongated along the less relevant ones. Such a modification locally influences class conditional probabilities, making them smother in the modified neighborhoods.

c. Approaches for learning distance metrics directly from the training examples. In (Goldberger et al., 2004), it was proposed a method for learning a Mahalanobis distance by maximizing a stochastic variation of the k-NN leave-one-out error. Similarly, Weinberger et al. (2005) proposed a method for learning a Mahalanobis distance by applying semidefinite programming. These concepts are close to approaches for building trainable similarity measures. See for example (Paclík et al., 2006b; Paclík et al., 2006a).

d. A simple adaptive k-NN classification algorithm based on the concept of statistical confidence (Wang et al., 2005; Wang et al., 2006). This approach involves a local adaptation of the distance measure, similarly to the other methods mentioned above. However, this method also includes a weighting procedure to assign a weight to each nearest neighbor according to its statistical confidence.

e. In (Wang et al., 2007), the same authors of the adaptation by using statistical confidence proposed a simple and elegant approach based on a normalization of the Euclidean or Manhattan distance from a query point to each training point by the shortest distance between the corresponding training point to training points of a different class. Such a new normalized distance is not symmetric and therefore is generally not a metric.

f.   Other adaptations of the distance and modifications of the rule include the works by Hastie & Tibshirani (1996), Sánchez et al. (1998), Wilson & Martínez (1997), Avesani et al. (1999) and Paredes & Vidal (2000).

2.3 Prototype generation

The nearest neighbor classifier is sensitive to outliers, e.g. erroneously chosen, atypical or noisy prototypes. In order to overcome this drawback, several techniques have been proposed to tackle the problem of prototype optimization (Pękalska et al., 2006). A fundamental dichotomy in prototype optimization divides the approaches into prototype generation and prototype selection. In this section we will discuss some techniques of the first group. Prototype selection techniques are reviewed in §2.4.

Prototype generation techniques are fundamentally based on two operations on an initial set of prototypes: first, merging and averaging the initial set of prototypes in order to obtain a smaller set which optimizes the performance of the k-NN rule or; second, creating a larger set by creating new prototypes or even new functions or models generated by the initial set, see also Fig. 1. In this section we refer only to merging techniques. The second group — which includes the nearest feature classifiers— deserves a separated section. Examples of merging techniques are the following:

a.   The k-means algorithm (MacQueen, 1967; Duda et al., 2001). It is considered the simplest clustering algorithm. Applied to prototype optimization, this technique aims to find a subset of prototypes generated from the original ones. New prototypes are the means of a number of partitions found by merging the original prototypes into a desired number of clusters. The algorithm starts by partitioning the original representation or prototype set $R=\{p_1, p_2, …p_N\}$ into M initial sets. Afterwards, the mean point, or centroid, for each set is calculated. Then, a new partition is constructed by associating each prototype with the nearest centroid. Means are recomputed for the new clusters. The algorithm is repeated until it converges to a stable solution; that is, when prototypes no longer switch clusters. It is equivalent to observe no changes in the value of means or centroids. Finally, the new set of merged or averaged prototypes is composed by the M means: $R_\mu=\{\mu_1, \mu_2, …, \mu_M\}$.

b.   The learning vector quantization (LVQ) algorithm (Kohonen, 1995). It consists in moving a fixed number M of prototypes p_i towards to or away from the training points x_i. The set of generated prototypes is also called *codebook*. Prototypes are iteratively updated according to a learning rule. In the original learning process, the delta rule is used to update prototypes by adding a fraction of the difference between the current value of the prototype and a new training point x. The rule can be written as follows:

$$p_i(t+1)=p_i(t) + \alpha(t)[x(t) - p_i(t)], \tag{2}$$

where α controls the learning rate. Positive values of α move p_i towards x; conversely, negative values move p_i away from x.

In statistical terms, the LVQ learning process can be interpreted as a way to generate a set of prototypes whose density reflects the shape of a function s defined as (Laaksonen, 1997):

$$s(x) = P_j f_j(x) - \max_{k \neq j} P_k f_k(x), \tag{3}$$

where P_j and f_j are the a priori probability and the probability density functions of class j, respectively. See (Holmström et al., 1996) and (Holmström et al., 1997) for further details.

c. Other methods for generating prototypes include the learning k-NN classifier (Laaksonen & Oja, 1996), neuralnet-based methods for constructing optimized prototypes (Huang et al., 2002) and cluster-based prototype merging procedures, e.g. the work by Mollineda et al. (2002).

2.4 Nearest feature classifiers

The nearest feature classifiers are geometrical extensions of the nearest neighbor rule. They are based on a measure of distance between the query point and a function calculated from the prototypes, such as a line, a plane or a space. In this work, we review three different nearest feature rules: the nearest feature line or NFL, the nearest feature plane or NFP and the nearest feature space or NFS. Their natural extensions by majority voting are the k nearest feature line rule, or k-NFL, and the k nearest feature plane rule, or k-NFP (Orozco-Alzate & Castellanos-Domínguez, 2006). Two recent improvements of NFL and NFP are also discussed here: the rectified nearest feature line segment (RNFLS) and the genetic nearest feature plane (G-NFP), respectively.

Nearest Feature Line

The *k nearest feature line* rule, or k-NFL (Li & Lu, 1999), is an extension of the k-NN classifier. This method generalizes each pair of prototype feature points belonging to the same class: $\{x_{ci}, x_{cj}\}$ by a linear function L_{ij}^c, which is called *feature line* (see Fig. 2). The line is expressed by the span $L_{ij}^c = sp(x_{ci}, x_{cj})$. The query x is projected onto L_{ij}^c as a point p_{ij}^c. This projection is computed as

$$p_{ij}^c = x_{ci} + \tau(x_{cj} - x_{ci}), \tag{4}$$

where $\tau = (x - x_{ci})(x_{cj} - x_{ci}) / \| x_{cj} - x_{ci} \|^2$. Parameter τ is called the position parameter. When $0 < \tau < 1$, p_{ij}^c is in the interpolating part of the feature line; when $\tau > 1$, p_{ij}^c is in the forward extrapolating side and; when $\tau < 0$, p_{ij}^c is in the backward extrapolating part. The two special cases when the query point is exactly projected on top of one of the points generating the feature line correspond to $\tau = 0$ and $\tau = 1$. In such cases, $p_{ij}^c = x_{ci}$ and $p_{ij}^c = x_{cj}$, respectively. The classification of x is done by assigning it the class label $\hat{c}$ most frequently represented among the k nearest feature lines, for k=1 that means:

$$d(x, L_{\hat{i}\hat{j}}^{\hat{c}}) = \min_{1 \le c \le C;\ 1 \le i,j \le n_c;\ i \ne j} d(x, L_{ij}^c), \tag{5}$$

where $d(x, L_{ij}^c) = \| x - p_{ij}^c \|$. In this case, the number of distance calculations is:

$$n_L = \sum_{c=1}^{C} n_c(n_c - 1) / 2 \tag{6}$$
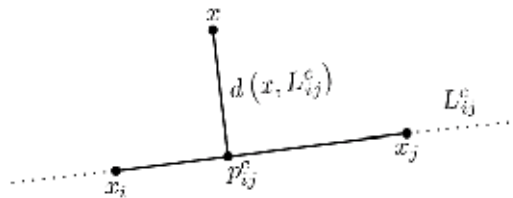
Fig. 2. Feature line and projection point onto it.

The nearest feature line classifier is supposed to deal with variations such as changes in viewpoint, illumination and face expression (Zhou et al., 2000). Such variations correspond to new conditions which were possibly not represented in the available prototypes. Consequently, the k-NFL classifier expands the representational capacity of the originally available feature points. Some typical variations in image faces taken from the *Biometric System Lab* data set (Cappelli et al., 2002) are shown in Fig. 3.

Fig. 3. Samples from the *Biometric System Lab* face dataset. Typical variations in face images are illustrated: illumination (first row), expression (second row) and pose (third row).

Nearest Feature Plane

The k nearest feature plane rule (Chien & Wu, 2002), or k-NFP, is an extension of the k-NFL classifier. This classifier assumes that at least three linearly independent prototype points are available for each class. It generalizes three feature points $\{x_{ci}, x_{cj}, x_{cm}\}$ of the same class by a feature plane F_{ijm}^c (see Fig. 4); which is expressed by the span $F_{ijm}^c = \mathrm{sp}\left(x_{ci}, x_{cj}, x_{cm}\right)$. The query x is projected onto F_{ijm}^c as a point p_{ijm}^c. See Fig. 4. The projection point can be calculated as follows:

$$p_{ijm}^c = X_{ijm}^c \left(X_{ijm}^{c\ T} X_{ijm}^c\right)^{-1} X_{ijm}^{c\ T} x, \tag{7}$$

where $X_{ijm}^c = \left[x_{ci}\ x_{cj}\ x_{cm}\right]$. Considering k=1, the query point x is classified by assigning it the class label ĉ, according to

$$d\left(x, F_{ij\hat{m}}^{\hat{c}}\right) = \min_{1 \le c \le C;\ 1 \le i,j,m \le n_c;\ i \ne j \ne m} d\left(x, F_{ijm}^c\right), \tag{8}$$

where $d\left(x, F_{ijm}^c\right) = \left\| x - p_{ijm}^c \right\|$. In this case, the number of distance calculations is:

$$n_F = \sum_{c=1}^{C} n_c (n_c - 1)(n_c - 2)/6 \tag{9}$$
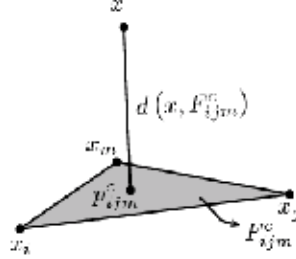
Fig. 4. Feature plane and projection point onto it.

Nearest Feature Space

The *nearest feature space* rule (Chien & Wu, 2002), or NFS, extends the geometrical concept of k-NFP classifier. It generalizes the independent prototypes belonging to the same class by a feature space $S^c = sp\left(x_{c1}, x_{c2}, \ldots, x_{cn_c}\right)$. The query point x is projected onto the C spaces as follows

$$p^c = X^c \left(X^{cT} X^c\right)^{-1} X^{cT} x, \tag{10}$$

where $X^c = \left[x_{c1}\ x_{c2}\ \cdots\ x_{cn_c}\right]$. The query point x is classified by assigning it the class label ĉ, according to

$$d\left(x, S^{\hat{c}}\right) = \min_{1 \le c \le C} d\left(x, S^c\right) = \min_{1 \le c \le C} \left\| x - p^c \right\| \tag{11}$$

Always, C distance calculations are required. It was geometrically shown in (Chien & Wu, 2002) that the distance of x to F_{ijm}^c is smaller than that to the feature line. Moreover, the distance to the feature line is nearer compared with the distance to two prototype feature points. This relation can be written as follows:

$$d\left(x, F_{ijm}^c\right) \le \min\left(d\left(x, L_{ij}^c\right), d\left(x, L_{jm}^c\right), d\left(x, L_{mi}^c\right)\right) \le \min\left(d\left(x, x_{ci}\right), d\left(x, x_{cj}\right), d\left(x, x_{cm}\right)\right) \tag{12}$$

In addition,

$$d\left(x, S^c\right) = \min_{1 \le c \le C} d\left(x, F_{ijm}^c\right) \tag{13}$$

In consequence, k-NFL classifier is supposed to capture more variations than k-NN, k-NFP should handle more variations of each class than k-NFL and NFS should capture more variations than k-NFP. So, it is expected that k-NFL performs better than k-NN, k-NFP is more accurate than k-NFL and NFS outperforms k-NFP.

Rectified Nearest Feature Line Segment

Recently, two main drawbacks of the NFL classifier have been pointed out: extrapolation and interpolation inaccuracies. The first one was discussed by (Zheng et al., 2004), who also

proposed a solution termed as nearest neighbor line (NNL). The second one —interpolation inaccuracy— was considered in (Du & Chen, 2007). They proposed an elegant solution called *rectified nearest feature line segment* (RNFLS). Their idea is aimed to overcome not just the interpolation problems but also the detrimental effects produced by the extrapolation inaccuracy. In the subsequent paragraphs, we will discuss both inaccuracies and the RNFLS classifier.

Extrapolation inaccuracy. It is a major shortcoming in low dimensional feature spaces. Nonetheless, its harm is limited in higher dimensional ones such those generated by pixel-based representations for face recognition. Indeed, several studies related to NFL applied to high dimensional feature spaces have reported improvements in classification performance; see for instance (Li, 2000; Li et al., 2000; Orozco-Alzate & Castellanos-Domínguez, 2006; Orozco-Alzate & Castellanos-Domínguez, 2007). In brief, the extrapolation inaccuracy occurs when the query point is far from the two points generating the feature line L but, at the same time, the query is close to the extrapolating part of L. In such a case, classification is very likely to be erroneous. Du & Chen (2007) mathematically proved for a two-class problem that the probability that a feature line L^1 (first class) trespasses the region R_2 (second class) asymptotically approaches to 0 as the dimension becomes large. See (Du & Chen, 2007) for further details.

Interpolation inaccuracy. This drawback arises in multi-modal classification problems. That is, when one class c_i has more than one cluster and the territory between two of them belongs to another class c_j, $i \neq j$. In such a case, a feature line linking two points of the multi-modal class will trespass the territory of another class. Consequently, a query point located near to the interpolating part of the feature line might be erroneously assigned to the class of the feature line.

As we stated before, the two above-mentioned drawbacks are overcome by the so-called rectified nearest feature line segment. It consists in a two step correction procedure for the original k-NFL classifier. Such steps are a segmentation followed by a rectification. Segmentation consists in cutting off the feature line in order to preserve only the interpolating part which is called a feature line segment $\widetilde{L}_{ij}^c$. Segmentation is aimed to avoid the extrapolation inaccuracy. When the orthogonal projection of a query point onto L_{ij}^c is in the interpolating part; that is, in $\widetilde{L}_{ij}^c$, the distance of such query point to $\widetilde{L}_{ij}^c$ is computed in the same way that the distance to L_{ij}^c, i.e. according to Eqs. (4) and (5). In contrast, when the projection point p_{ij}^c is in the extrapolating part, the distance to $\widetilde{L}_{ij}^c$ is forced to be equal to the distance of the query point to one of the extreme point of $\widetilde{L}_{ij}^c$: x_{ci} if p_{ij}^c is in the backward extrapolating part and x_{cj} if p_{ij}^c is the forward extrapolating part, respectively. See Fig. 5.
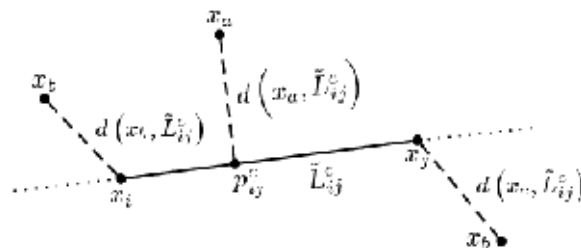

Fig. 5. Feature line segment and distances to it for three cases: projection point in the interpolating part, projection point in the backward extrapolating part and projection point in the forward extrapolating part.

Afterwards, the rectification procedure is achieved in order to avoid the effect of the interpolation inaccuracy. It consists in removing feature line segments trespassing the territory of another class. To do so, the concept of territory must be defined. Indeed, Du & Chen (2007) define two types of territory. The first one is called *sample territory* which, for a particular feature point x, stands for a ball centered at x with a radius equals to the distance from x to its nearest neighbor belonging to a different class. The second one —the *class territory*— is defined as the union of all sample territories of feature points belonging to the same class. Then, for each feature line segment, we check if it trespass the class territory of another class or not. In the affirmative case, we proceed to remove that feature line segment from the representation set. Finally, classification is performed in a similar way to Eq. (5) but replacing feature lines L_{ij}^c by those feature line segments $\widetilde{L}_{ij}^c$ which were not removed during the rectification.

Center-based Nearest Neighbor Classifier and Genetic Nearest Feature Plane
The k-NFL and k-NFP classifiers tend to be computationally unfeasible as the number of training objects per class grows. Such a situation is caused by to the combinatorial increase of combinations of two and three feature points, see Eqs. (6) and (9). Some alternatives to overcome this drawback have been recently published. Particularly, the center-based nearest neighbor (CNN) classifier (Gao & Wang, 2007) and the genetic nearest feature plane (G-NFP) (Nanni & Lumini, 2007). The first one is aimed at reducing the computation cost of the k-NFL classifier by using only those feature lines linking two feature points of the same class and, simultaneously, passing by the centroid of that class. In such a way, only a few feature lines are kept (authors call them center-based lines) and computation time is therefore much lower. The G-NFP classifier is a hybrid method to reduce the computational complexity of the k-NFP classifier by using a genetic algorithm (GA). It consists in a GA-based prototype selection procedure followed by the conventional method to generate feature lines. Selected prototypes are centroids of a number of intra-class clusters found by the GA.

2.5 Prototype selection
Prototype selection methods aim at the reduction of the initial set of prototypes while maintaining an acceptable classification accuracy or even increasing it. There are two groups of prototypes selection methods as shown in Fig. 1. See also (Wilson & Martínez, 2000) and (Lozano et al., 2006). Editing methods (Wilson, 1972; Devijver & Kittler, 1982; Aha et al., 1991) remove noisy and/or close border prototypes in order to avoid overlapping and smoothing the resulting decision boundaries. In other words, they are intended to produce a subset of prototypes forming homogeneous clusters in the feature space. Condensing algorithms try to select a small subset of prototypes while preserving classification performance as good as possible. Condensing may involve just a pure selection of prototypes (Hart, 1968; Tomek, 1976; Toussaint et al., 1985; Dasarathy, 1990; Dasarathy, 1994) or include a modification of them (Chang, 1974; Chen & Józwik, 1996; Ainslie & Sánchez, 2002; Lozano et al., 2004a; Lozano et al., 2004b).

2.6 HMM-based sequence classification
There are two standard ways for classifying sequences using HMMs. The first one is referred to as ML_{OPC} (Maximum-likelihood, one per class HMM-based classification).

Assume that a particular object x, a face in our case of interest, is represented by a sequence O and that C HMMs, $\{\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(C)}\}$, has been trained; i.e. there is one trained HMM per class. Thus, the sequence O is assigned to the class showing the highest likelihood:

$$\text{Class}(O) = \arg\max_c P(O|\lambda^{(c)}) \tag{14}$$

The likelihood $P(O, \lambda^{(c)})$ is the probability that the sequence O was generated by the model $\lambda^{(c)}$. The likelihood can be estimated by different methods such as the Baum-Welch estimation procedure (Baum et al., 1970) and the forward-backward procedure (Baum, 1970). This approach can be considered analogous to the nearest feature space classification (see §2.4), with the proximity measure defined by the likelihood function.

The second method, named as ML_{OPS} (Maximum-likelihood, one per sequence HMM-based classification) consists in training one HMM per each training sequence $O_i^{(c)}$, where c denotes the class label. Similarly to (14), it can be written as:

$$\text{Class}(O) = \arg\max_c(\arg\max_i P(O|\lambda_i^{(c)})) \tag{15}$$

This method is analogous to the nearest neighbor classifier. Compare (1) and (15).

3. Dissimilarity representations

In (Orozco-Alzate & Castellanos-Domínguez, 2007) we introduced the concepts of dissimilarity-based face recognition and their relationship with the nearest feature classifiers. For convenience of the reader and for the sake of self–containedness, we repeat here a part of our previously published discourse on this matter.

A dissimilarity representation of objects is based on their pairwise comparisons. Consider a representation set $R:=\{p_1, p_2, \ldots, p_n\}$ and a dissimilarity measure d. An object x is represented as a vector of the dissimilarities computed between x and the prototypes from R, i.e. $D(x,R)=[d(x,p_1), d(x,p_2), \ldots, d(x,p_n)]$. For a set T of N objects, it extends to an N×n dissimilarity matrix (Pękalska & Duin, 2005a):

$$D(T,R) = \begin{bmatrix} d_{11} & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & d_{22} & d_{23} & \cdots & d_{2n} \\ d_{31} & d_{32} & d_{33} & \cdots & d_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & d_{n3} & \cdots & d_{nn} \end{bmatrix}, \tag{16}$$

where $d_{jk} = D(x_j, p_k)$.

For dissimilarities, the geometry is contained in the definition, giving the possibility to include physical background knowledge; in contrast, feature-based representations usually suppose a Euclidean geometry. Important properties of dissimilarity matrices, such as metric nature, tests for Euclidean behavior, transformations and corrections of non-Euclidean dissimilarities and embeddings, are discussed in (Pękalska & Duin, 2005b).

When the entire T is used as R, the dissimilarity representation is expressed as an N×N dissimilarity matrix D(T,T). Nonetheless, R may be properly chosen by prototype selection procedures. See §2.5 and (Pękalska et al., 2006).

3.1 Classifiers in dissimilarity spaces

Building a classifier in a dissimilarity space consists in applying a traditional classification rule, considering dissimilarities as features; it means, in practice, that a dissimilarity-based classification problem is addressed as a traditional feature-based one. Even though the nearest neighbor rule is the reference method to discriminate between objects represented by dissimilarities, it suffers from a number of limitations. Previous studies (Pękalska et al., 2001; Pękalska & Duin, 2002; Paclík & Duin, 2003; Pękalska et al., 2004; Orozco-Alzate et al., 2006) have shown that Bayesian (normal density based) classifiers, particularly the linear (LDC) and quadratic (QDC) normal based classifiers, perform well in dissimilarity spaces and, sometimes, offer a more accurate solution. For a 2-class problem, the LDC based on the representation set R is given by

$$f(D(x,R)) = \left[D(x,R) - \frac{1}{2} \left(\mathbf{m}_{(1)} + \mathbf{m}_{(2)} \right) \right]^T \times C^{-1} \left(\mathbf{m}_{(1)} - \mathbf{m}_{(2)} \right) + \log \frac{P_{(1)}}{P_{(2)}} \tag{17}$$

and the QDC is derived as

$$f(D(x,R)) = \sum_{i=1}^{2} (-1)^i \left(D(x,R) - \mathbf{m}_{(i)} \right)^T \times C_{(i)}^{-1} \left(D(x,R) - \mathbf{m}_{(i)} \right) + 2\log \frac{p_{(1)}}{P_{(2)}} + \log \frac{\left| C_{(1)} \right|}{\left| C_2 \right|} \tag{18}$$

where C is the sample covariance matrix, $C_{(1)}$ and $C_{(2)}$ are the estimated class covariance matrices, and $\mathbf{m}_{(1)}$ and $\mathbf{m}_{(2)}$ are the mean vectors, computed in the dissimilarity space D(T,R). $P_{(1)}$ and $P_{(2)}$ are the class prior probabilities. If C is singular, a regularized version must be used. In practice, the following regularization is suggested for r=0.01 (Pękalska et al., 2006):

$$C_{reg}^r = (1-r)C + r\,diag(C) \tag{19}$$

Nonetheless, regularization parameter should be optimized in order to obtain the best possible results for the normal density based classifiers.

Other classifiers can be implemented in dissimilarity spaces, usually by a straightforward implementation. Nearest mean linear classifiers, Fisher linear discriminants, support vector machines (SVMs), among others are particularly interesting for being used in generalized dissimilarity spaces. In addition, traditional as well as specially derived clustering techniques can be implemented for dissimilarity representations, see (Pękalska & Duin, 2005c) for a detailed discussion on clustering techniques in dissimilarity representations.

3.2 Generalization of dissimilarity representations

Dissimilarity representations were originally formulated as pairwise constructs derived by object to object comparisons. Nonetheless, it is also possible to define them in a wider form, e.g. defining representations based on dissimilarities with functions of (or models built by) objects. In the general case, representation objects used for building those functions or models do not need labels, allowing for semi-supervised approaches in which the unlabeled objects are used for the representation and not directly for the classifier, or might even be artificially created, selected by an expert or belong to different classes than the ones under consideration (Duin, 2008).

We phrase such a wider formulation as a *generalized dissimilarity representation*. In spite of the potential to omit labels, to the best of the authors' knowledge, all the current generalization

procedures −including ours− make use of labels. At least three different approaches for generalizing dissimilarity representations have been proposed and developed independently: generalization by using hidden Markov models (Bicego et al., 2004), generalization by pre-clustering (Kim, 2006) and our own proposal of generalizing dissimilarity representations by using feature lines and feature planes. In this subsection, we discuss the first two approaches, focusing particularly in their motivations and methodological principles. The last one is discuss in §3.3.

Dissimilarity-based Classification Using HMMs

It can be easily seen that likelihoods $P(O|\lambda^{(c)})$ and/or $P(O|\lambda_i^{(c)})$ can be interpreted as similarities; e.g. Bicego et al. (2004) propose to use the following similarity measure between two sequences O_i and O_j:

$$d_{ij} = d(O_i,O_j) = \log P(O_i|\lambda_j)/T_i \qquad (20)$$

where T_i is the length of the sequence O_i, introduced as a normalization factor to make a fair comparison between sequences of different length. Notice that, even though we use d to denote a dissimilarity measure, in (20) we are in fact referring to a similarity. Nonetheless, the two concepts are closely related and even used indistinctively as in (Bicego et al., 2004). In addition, there exist some ways of changing a similarity value into a dissimilarity value and vice versa (Pękalska & Duin, 2005a).

A HMM-based dissimilarity might be derived by measuring the likelihood between all pairs of sequences and HMMs. Consider a representation set $R=\{O_1^{(1)},O_2^{(1)},…,O_M^{(C)}\}$. A dissimilarity representation for a new sequence O is given in terms of the likelihood of having generated O with the associated HMMs for each sequence in R. Those HMMs are grouped in the representation set $R_\lambda=\{\lambda_1^{(1)},\lambda_2^{(1)},…,\lambda_M^{(C)}\}$. In summary, the sequence O is represented by the following vector: $D(O,R_\lambda)=[d_1 \quad d_2 \cdots d_M]$. For a training set $T=\{O_1,O_2,…,O_N\}$, it extends to a matrix $D(T,R_\lambda)$ as shown in Fig. 6.

$$D(T,R_\lambda) = \begin{matrix} & \lambda_1^{(1)} & \lambda_2^{(1)} & \cdots & \lambda_m^{(c)} & \cdots & \lambda_M^{(C)} \\ O_1 & d_{11} & d_{12} & \cdots & d_{1m} & \cdots & d_{1M} \\ O_2 & d_{21} & d_{12} & \cdots & d_{2m} & \cdots & d_{2M} \\ O_3 & d_{31} & d_{32} & \cdots & d_{3m} & \cdots & d_{3M} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ O_N & d_{N1} & d_{N2} & \cdots & d_{Nm} & \cdots & d_{NM} \end{matrix}$$

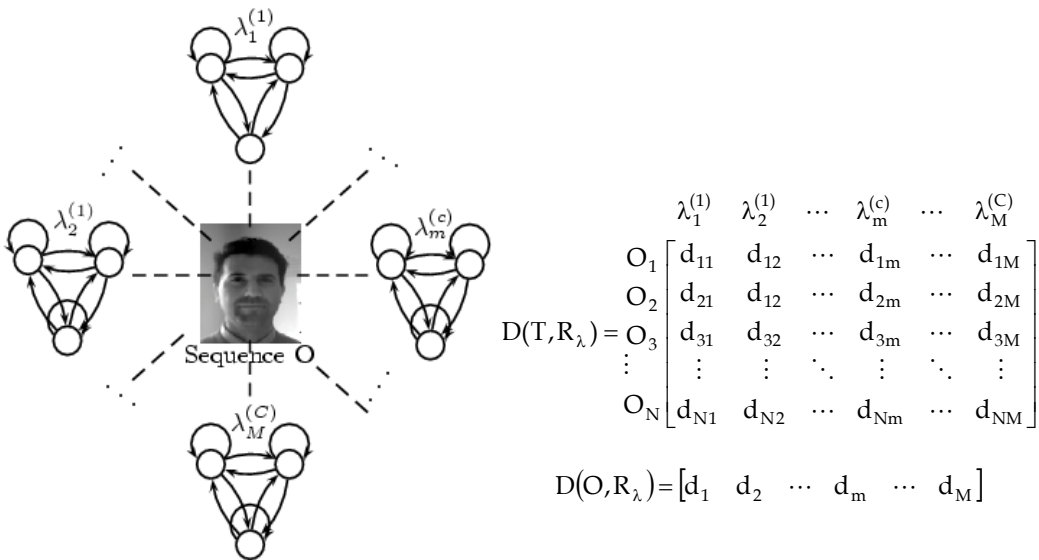$$D(O,R_\lambda) = \begin{bmatrix} d_1 & d_2 & \cdots & d_m & \cdots & d_M \end{bmatrix}$$

Fig. 6. Generalization of a dissimilarity representation by using HMMs.

On top of the generalized dissimilarity representation $D(T,R_\lambda)$, a dissimilarity-based classifier can be built.

Generalization by Clustering Prototypes

Kim (2006) proposed a methodology to overcome the SSS problem in face recognition applications. In summary, the proposed approach consists in:

1. Select a representation set R from the training set T.
2. Compute a dissimilarity representation $D(T,R)$ by using some suitable dissimilarity measure.
3. For each class, perform a clustering of R into a few subsets $Y_m^{(c)}$, c=1,…,C and i= m,…, M; that is, M clusters of objects belonging to the same class. Any clustering method can be used; afterwards, the M mean vectors $\hat{Y}_i^{(c)}$, are computed by averaging each cluster.
4. A dissimilarity based classifier is built in $D(T,R_Y)$. Moreover, a fusion technique may be used in order to increase the classification accuracy.

Kim's attempt to reduce dimensionality by choosing means of clusters as representatives can be interpreted as a generalization procedure. Similarly to the case of generalization by HMMs, this generalization procedure by clustering prototypes is schematically shown in Fig. 7.

$$D(T,R_\lambda) = \begin{array}{c} \\ O_1 \\ O_2 \\ O_3 \\ \vdots \\ O_N \end{array} \begin{array}{ccccccc} \lambda_1^{(1)} & \lambda_2^{(1)} & \cdots & \lambda_m^{(c)} & \cdots & \lambda_M^{(C)} \\ \left[\begin{array}{cccccc} d_{11} & d_{12} & \cdots & d_{1m} & \cdots & d_{1M} \\ d_{21} & d_{12} & \cdots & d_{2m} & \cdots & d_{2M} \\ d_{31} & d_{32} & \cdots & d_{3m} & \cdots & d_{3M} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & d_{Nm} & \cdots & d_{NM} \end{array}\right] \end{array}$$

$$D(O,R_\lambda) = \begin{bmatrix} d_1 & d_2 & \cdots & d_m & \cdots & d_M \end{bmatrix}$$
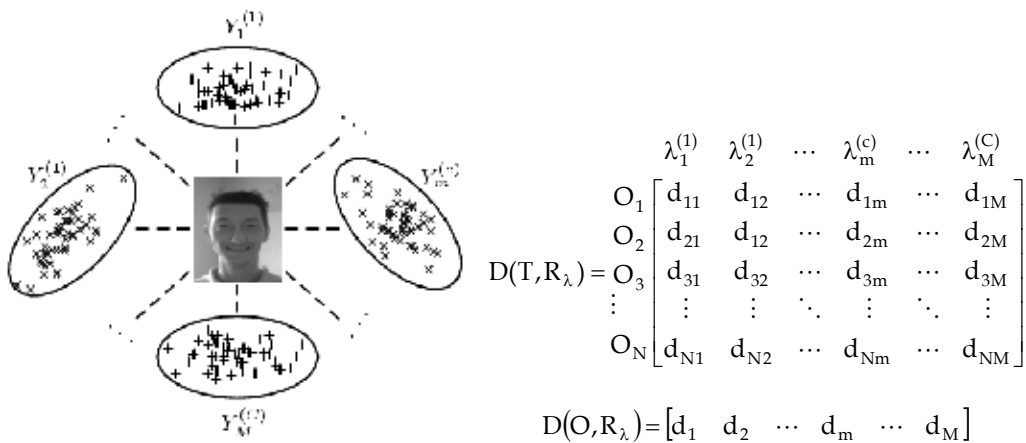
Fig. 7. Generalization of dissimilarity representations by clustering prototypes.

As we explained above, our generalization method by feature lines and feature planes can be included in the family of model- or function-based generalization procedures. It will be presented in detail in the subsequent section. For the sake of comparison, here we point to a few relevant coincidences and differences between the generalizations by HMMs and clustering and our proposed approach by feature lines and feature planes. See Fig. 8 and compare it against Figs. 6 and 7. Firstly, notice that the representation role is played in our case by a function generated by two representative objects, e.g. the so-called feature lines: $\left\{L_m^c\right\}$. A given object x is now represented in terms of its dissimilarities to a set R_L of representative feature lines. It also extends to $D(T,R_L)$ for an entire training set. One remarkable difference is that our approach, in principle, leads to a higher dimensional space; i.e. M > N. In contrast, HMM- and cluster-based approaches leads in general to low dimensional spaces: M < N.
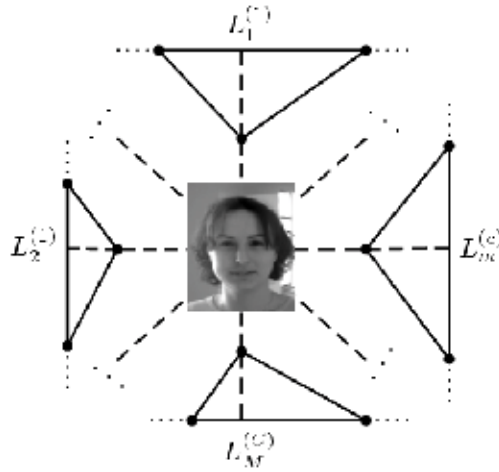
Fig. 8. Generalization of dissimilarity representations by feature lines.

3.3 Generalization by feature lines and feature planes

Our generalization consists in creating matrices $D_L(T,R_L)$ and $D_F(T,R_F)$ by using the information available at the original representation $D(T,R)$, where subindexes L and F stand for feature lines and feature planes respectively. This generalization procedure was proposed in (Orozco-Alzate & Castellanos-Domínguez, 2007) and (Orozco-Alzate et al., 2007a). In this section, we review our method as it was reported in the above-mentioned references but also including some results and remarks resulted from our most recent discussions and experiments.

$D_L(T,R_L)$ and $D_F(T,R_F)$ are called *generalized dissimilarity representations* and their structures are:

$$D_L(T,R_L)=\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{array}\begin{array}{ccccc} L_1 & L_2 & L_3 & \cdots & L_n \\ \left[\begin{array}{ccccc} d_{11} & d_{12} & d_{13} & \cdots & d_{1n_L} \\ d_{21} & d_{22} & d_{23} & \cdots & d_{2n_L} \\ d_{31} & d_{32} & d_{33} & \cdots & d_{3n_L} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & d_{N3} & \cdots & d_{Nn_L} \end{array}\right] \end{array}, \tag{21}$$

where $d_{jk}=D_L(x_j,L_k)$; and

$$D_F(T,R_F)=\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{array}\begin{array}{ccccc} F_1 & F_2 & F_3 & \cdots & F_n \\ \left[\begin{array}{ccccc} d_{11} & d_{12} & d_{13} & \cdots & d_{1n_F} \\ d_{21} & d_{22} & d_{23} & \cdots & d_{2n_F} \\ d_{31} & d_{32} & d_{33} & \cdots & d_{3n_F} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & d_{N3} & \cdots & d_{Nn_F} \end{array}\right] \end{array}, \tag{22}$$

where $d_{jk}=D_F(x_j, F_k)$.

D(T, R_L) and D(T, R_F) are high dimensional matrices because the original representation set R is generalized by combining all the pairs (R_L) and all the triplets (R_F) of prototypes of the same class. Consequently, a proper procedure for prototype selection (dimensionality reduction) is needed in order to avoid the curse of the dimensionality. Another option is to use a strong regularization procedure. In general, all the possible dissimilarities between objects are available but the original feature points are not. Nonetheless, it is possible to compute the distances to feature lines from the dissimilarities. The problem consists in computing the height of a scalene triangle as shown in Figure 9.
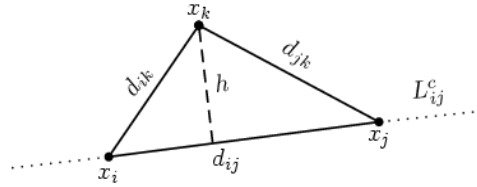
Fig. 9. Height of a scalene triangle corresponding to the distance to a feature line.

Let us define s=(d_{jk}+d_{ij}+d_{ik})/2. Then, the area of the triangle is given by:

$$A = \sqrt{s(s-d_{jk})(s-d_{ij})(s-d_{ik})};$$ (23)

but it is also known that area, assuming d_{ij} as base, is:

$$A = \frac{d_{ij}h}{2}$$ (24)

So, we can solve (23) and (24) for h, which is the distance to the feature line. The generalized dissimilarity representation in (21) is constructed by replacing each entry of D(T,R_L) by the corresponding value of h. The distance d_{ij} in Fig. 9 must be an intraclass one.

Computing the distances to the feature planes in terms of dissimilarities consists in calculating the height of an irregular (scalene) tetrahedron as shown in Fig. 10.
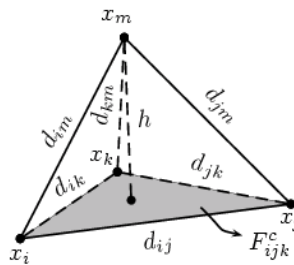

Fig. 10. Height of an irregular tetrahedron corresponding to the distance to a feature plane.

Let us define s=(d_{jk}+d_{ij}+d_{ik})/2. Then, the volume of a tetrahedron is given by:

$$V = \frac{h\sqrt{s(s-d_{jk})(s-d_{ij})(s-d_{ik})}}{3};$$ (25)

but volume is also (Uspensky, 1948):

$$V^2 = \frac{1}{288}\begin{vmatrix} 0 & d_{ij}^2 & d_{ik}^2 & d_{im}^2 & 1 \\ d_{ij}^2 & 0 & d_{jk}^2 & d_{jm}^2 & 1 \\ d_{ik}^2 & d_{jk}^2 & 0 & d_{km}^2 & 1 \\ d_{im}^2 & d_{jm}^2 & d_{km}^2 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{vmatrix} \qquad (26)$$

So, we can solve (25) and (26) for h, which is the distance to the feature plane. The generalized dissimilarity representation in (22) is constructed by replacing each entry of $D(T,R_F)$ by the corresponding value of h. Distances d_{ij}, d_{ik} and d_{jk} in Fig. 10 must be intraclass.

Experiments (Orozco-Alzate and Castellanos-Domínguez, 2007; Orozco-Alzate et al., 2007a) showed that nearest feature rules are especially profitable when variations and conditions are not fully represented by the original prototypes; for example the case of small or non-representative training sets. The improvement in such a case respect to the reference method —the k-NN rule— is due to the feature lines/planes' ability to expand the representational capacity of the available points, accounting for new conditions not fully represented by the original set. Those are precisely the conditions in face recognition problems, where the number of prototypes is typically limited to few images per class and the number of classes is high: tens or even one hundred persons. As a result, the effectiveness of the nearest feature rules is remarkable for this problem.

Generalized dissimilarity representations by feature lines and feature planes are not square, having two or three zeros per column for feature lines and feature planes respectively. Firstly, we have considered generalizations of metric representations because the generalization procedure requires constructing triangles and tetrahedrons and, as a result, generalizing non-metric dissimilarity representations might produce complex numbers when solving equations for heights. An apparently promising alternative to take into account is the use of refining methods for non-metric dissimilarities; see (Duin & Pękalska, 2008). Such refining methods lead to pseudo-Euclidean embedded spaces or to the computation of distances on distances which is equivalent to apply the k-NN rule in the dissimilarity space.

In order to construct classifiers based on generalized dissimilarity representations, we should proceed similarly as dissimilarity-based classifiers are built; see §3.1. That is, using a training set T and a representation set R containing prototype examples from T. Prototype lines or planes considered must be selected by some prototype selection procedure; classifiers should be built on $D(T,R_L)$ and $D(T,R_F)$. Different sizes for the representation set R must be considered. In (Orozco-Alzate et al., 2007b), we proved an approach for selecting middle-length feature lines. Experiments showed that they are appropriate to represent moderately curved subspaces.

Enriching the dissimilarity representations implies a considerable number of calculations. The number of feature lines and planes grows rapidly as the number of prototypes per class increases; in consequence, computational effort may become high, especially if a generalized representation is computed for an entire set. When applying traditional statistical classifiers to dissimilarity representations, dissimilarities to prototypes may be treated as features. As a result, classifiers built in enriched dissimilarity spaces are also subject to the curse of dimensionality phenomenon. In general, for generalized dissimilarity representations

$D_g(T,R_g)$, the number of training objects is small relative to the number of prototype lines or planes.

According to the two reasons above, it is important to use dimensionality reduction techniques —feature extraction and feature selection methods— before building classifiers in generalized dissimilarity representations. Systematic approaches for prototype selection such as exhaustive search and the forward selection process lead to an optimal representation set; however, they require a considerable number of calculations. Consequently, due to the increased dimensionality of the enriched representations, the application of a systematic prototype selection method will be computationally expensive. Nonetheless, it has been shown that non-optimal and computationally simple procedures such as *Random* and *RandomC* may work well (Pękalska et al., 2006) and that simple geometrical criteria such the length of the lines (Orozco-Alzate et al., 2007b) may be wise for selecting representation subsets.

4. Conclusion

We started this chapter with a critic about the overwhelming research efforts in face recognition, discussing benefits and drawbacks of such a situation. The particular case of studies related to linear dimensionality reduction is a *per antonomasia* example of an enormous concentration of attention in a small and already mature area. We agree that research topics regarding preprocessing, classification and even nonlinear dimensionality reduction are much more promising and susceptible of significant contributions than further studies in LDA. Afterwards, we reviewed the state of the art in prototype-based classification for face recognition. Several techniques and variants have been proposed since the early days of the nearest neighbor classifier. Indeed, an entire family of prototype-based methods arose; some of the family members are entirely new ideas, others are modifications or hybrid methods. In brief, three approaches in prototype-based classification can be distinguished: modifications of the distance measure, prototype generation and prototype selection methods. The last two approaches present dichotomies as shown in Fig. 1. We focus our discussion on nearest feature classifiers and their improved versions as well as on their use in dissimilarity-based classification.

The main advantage of the RNFLS classifier is its property of generating feature line segments that are more concentrated in distribution than the original feature points. In addition, RNFLS corrects the interpolation and extrapolation inaccuracies of the k-NFL classifier, allowing us to use feature lines (in fact, feature line segments) in low dimensional classification problems. k-NFL was originally proposed and successfully used just in high dimensional representations such pixel-based representations for face recognition; however, thanks to the improvement provided by RNFLS, feature line-based approaches are also applicable now to feature-based face recognition. The k-NFP classifier is computationally very expensive. G-NFP can reduce the effort to a manageable amount, just by using a simple two-stage process: a GA-based prototype selection followed by the original k-NFP algorithm.

We presented and compare three different but related approaches to generalize dissimilarity representations by using HMMs, clustering techniques and feature lines/planes respectively. The first two are model-based extensions for a given dissimilarity matrix and lead, in general, to lower dimensional dissimilarity spaces. In contrast, our methodology produces high dimensional dissimilarity spaces and, consequently, proper prototype

selection methods must be applied. Generalized dissimilarity representations by feature lines and feature planes are in fact enriched instead of condensed representations. Consequently, they have the property of accounting for entirely new information not originally available in the given dissimilarity matrix. Potential open issues for further research are alternative methods for feature line/plane selection such as sparse classifiers and linear programming-based approaches. In order to extend methods based on feature lines and feature planes to the case of indefinite representations, correction techniques for non-Euclidean data are also of interest.

5. References

Aha, D. W., Kibler, D. F., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6(1):37–66.

Ainslie, M. C. & Sánchez, J. S. (2002). Space partitioning for instance reduction in lazy learning algorithms. In *2nd International Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning*, pages 13–18, Helsinki (Finland).

Avesani, P., Blanzieri, E., & Ricci, F. (1999). Advanced metrics for class-driven similarity search. In *DEXA '99: Proceedings of the 10th International Workshop on Database & Expert Systems Applications*, page 223, Washington, DC, USA. IEEE Computer Society.

Baum, L. E. (1970). An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequality*, 3:1–8.

Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171.

Bicego, M., Murino, V., & Figueiredo, M. A. T. (2004). Similarity-based classification of sequences using hidden Markov models. *Pattern Recognition*, 37(12):2281–2291.

Bunke, H. & Sanfeliu, A., editors (1990). *Syntactic and structural pattern recognition –Theory and applications*, volume 7 of *World Scientific Series in Computer Science*. World Scientific.

Cappelli, R., Maio, D., & Maltoni, D. (2002). Subspace classification for face recognition. In *Proceedings of the International Workshop on Biometric Authentication, (ECCV 2002 - Copenhagen)*, volume 2359/2002 of *Lecture Notes in Computer Science*, pages 133–142, London, UK. Springer-Verlag.

Chang, C.-L. (1974). Finding prototypes for nearest neighbor classifiers. *IEEE Trans. Comput.*, 23(11):1179–1184.

Chen, C. H. & Józwik, A. (1996). A sample set condensation algorithm for the class sensitive artificial neural network. *Pattern Recognition Letters*, 17(8):819–823(5).

Chien, J.-T. & Wu, C.-C. (2002). Discriminant waveletfaces and nearest feature classifiers for face recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(12):1644–1649.

Cover, T. M. & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Trans. Inform. Theory*, IT-13(1):21–27.

Dasarathy, B. V. (1990). *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, CA.

Dasarathy, B. V. (1994). Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design. *IEEE Trans. Systems, Man, and Cybernetics*, 24(3):511–517.

Delac, K. & Grgic, M., editors (2007). *Face Recognition*. Artificial Intelligence. I-TECH Education and Publishing, Vienna, Austria.

Devijver, P. A. & Kittler, J. (1982). *Pattern Recognition: a Statistical Approach*. Prentice Hall International, London.

Devroye, L., Györfi, L., & Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*, volume 31 of *Stochastic Modelling and Applied Probability*. Springer, Berlin.

Domeniconi, C., Peng, J., & Gunopulos, D. (2002). Locally adaptive metric nearest-neighbor classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(9):1281–1285.

Du, H. & Chen, Y. Q. (2007). Rectified nearest feature line segment for pattern classification. *Pattern Recognition*, 40(5):1486–1497.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification*. Wiley-Interscience, New York, 2 edition.

Duin, R. P. W. (2008). Personal communication.

Duin, R. P. W., Loog, M., & Ho, T. K. (2006). Recent submissions in linear dimensionality reduction and face recognition. *Pattern Recognition Letters*, 27(7):707–708.

Duin, R. P. W. & Pękalska, E. (2008). On refining dissimilarity matrices for an improved NN learning. In *19th International Conference on Pattern Recognition ICPR 2008. Tampa, USA*. Accepted to be presented.

Friedman, J. H. (1994). Flexible metric nearest neighbor classification. Technical report, Department of Statistics, Stanford University.

Gao, Q.-B. & Wang, Z.-Z. (2007). Center-based nearest neighbor classifier. *Pattern Recognition*, 40(1):346–349.

Goldberger, J., Roweis, S., Hinton, G., & Salakhutdinov, R. (2004). Neighborhood component analysis. In Saul, L. K., Weiss, Y., & Bottou, L., editors, *Advances in Neural Information Processing Systems. Proceedings of the 17th Eighteenth Annual Conference on Neural Information Processing Systems NIPS2004*, volume 17, pages 513–520, Cambridge, MA. Neural Information Processing Systems Foundation, MIT Press.

Hart, P. E. (1968). The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, IT-14(3):515–516.

Hastie, T. & Tibshirani, R. (1996). Discriminant adaptive nearest neighbor classification and regression. In Touretzky, D. S., Mozer, M. C., & Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems*, volume 8, pages 409–415. The MIT Press.

Holmström, L., Koistinen, P., Laaksonen, J., & Oja, E. (1996). Comparison of neural and statistical classifiers — theory and practice. Technical report a13, Rolf Nevanlinna Institute, Helsinki.

Holmström, L., Koistinen, P., Laaksonen, J., & Oja, E. (1997). Neural and statistical classifiers-taxonomy and two case studies. *IEEE Transactions on Neural Networks*, 8(1):5–17.

Huang, Y. S., Chiang, C. C., Shieh, J. W., & Grimson, W. E. L. (2002). Prototype optimization for nearest-neighbor classification. *Pattern Recognition*, 35(6):1237–1245.

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(1):4–37.

Kim, S.-W. (2006). On using a dissimilarity representation method to solve the small sample size problem for face recognition. In Blanc-Talon, J., Philips, W., Popescu, D., & Scheunders, P., editors, *Advanced Concepts for Intelligent Vision Systems, Proceedings of the 8th International Conference, ACIVS 2006*, volume 4179 of *Lecture Notes in Computer Science*, pages 1174–1185, Antwerp, Belgium. Springer.

Kohonen, T. (1995). *Self-Organizing Maps*. Number 30 in Information Sciences. Springer, Germany.

Laaksonen, J. (1997). *Subspace Classifiers in Recognition of Handwritten Digits*. PhD thesis, Helsinki University of Technology. http://lib.tkk.fi/Diss/199X/isbn9512254794/.

Laaksonen, J. & Oja, E. (1996). Classification with learning k-nearest neighbors. In *Proceedings of the IEEE International Conference on Neural Networks (ICNN'96). Washington, DC, USA*, volume 3, pages 1480–1483. IEEE.

Li, S. Z. (2000). Content-based classification and retrieval of audio using the nearest feature line method. *IEEE Trans. Speech and Audio Process*, 8(5):619–625.

Li, S. Z. (2008). Nearest feature line. *Scholarpedia*, 3(3):4357. Available at http://www.scholarpedia.org/article/Nearest_feature_line.

Li, S. Z., Chan, K. L., & Wang, C. (2000). Performance evaluation of the nearest feature line method in image classification and retrieval. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(11):1335–1349.

Li, S. Z. & Lu, J. (1998). Generalizing capacity of face database for face recognition. In *Proceedings of Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 402–407, Nara, Japan. IEEE Computer Society.

Li, S. Z. & Lu, J. (1999). Face recognition using the nearest feature line method. *IEEE Trans. Neural Networks*, 10(2):439–443.

Liu, J., Chen, S., Tan, X., & Zhang, D. (2007). Efficient pseudoinverse linear discriminant analysis and its nonlinear form for face recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(8):1265–1278.

Lozano, M., Sánchez, J. S., & Pla, F. (2004a). Using the geometrical distribution of prototypes for training set condensing. In *Current Topics in Artificial Intelligence, Lecture Notes in Artificial Intelligence 3040*, pages 618–627. Springer-Verlag.

Lozano, M., Sotoca, J. M., Sánchez, J. S., & Pla, F. (2004b). An adaptive condensing algorithm based on mixtures of gaussians. In *7é Congrés Català d'Intel.ligència Artificial, Frontiers in Artificial Intelligence and Applications 113*, pages 225–232, Barcelona (Spain). IOS Press.

Lozano, M., Sotoca, J. M., Sánchez, J. S., Pla, F., Pękalska, E., & Duin, R. P. W. (2006). Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces. *Pattern Recognition*, 39(10):1827–1838.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Cam, L. M. L. & Neyman, J., editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

Mollineda, R. A., Ferri, F. J., & Vidal, E. (2002). An efficient prototype merging strategy for the condensed 1-nn rule through class-conditional hierarchical clustering. *Pattern Recognition*, 35(12):2771–2782.

Nanni, L. & Lumini, A. (2007). Genetic nearest feature plane. *Expert Systems with Applications*. In Press, Corrected Proof.

Nhat, V. D. M. & Lee, S. (2007). Image-based subspace analysis for face recognition. In Delac, K. & Grgic, M., editors, *Face Recognition*, Artificial Intelligence, chapter 17, pages 321–336. I-TECH Education and Publishing, Vienna, Austria.

Orozco-Alzate, M. & Castellanos-Domínguez, C. G. (2006). Comparison of the nearest feature classifiers for face recognition. *Machine Vision and Applications (Springer)*, 17(5):279–285.

Orozco-Alzate, M. & Castellanos-Domínguez, C. G. (2007). Nearest feature rules and dissimilarity representations for face recognition problems. In Delac, K. & Grgic, M., editors, *Face Recognition*, Artificial Intelligence, chapter 18, pages 337–356. I-TECH Education and Publishing, Vienna, Austria.

Orozco-Alzate, M., Duin, R. P. W., & Castellanos-Domínguez, C. G. (2007a). Generalizing dissimilarity representations using feature lines. In Rueda, L., Mery, D., & Kittler, J., editors, *Progress in Pattern Recognition, Image Analysis and Applications. Proceedings of the 12th Iberoamerican Congress on Pattern Recognition CIARP 2007*, volume 4756 of *Lecture Notes in Computer Science*, pages 370–379, Viña del Mar-Valparaiso, Chile. IAPR, Springer.

Orozco-Alzate, M., Duin, R. P. W., & Castellanos-Domínguez, C. G. (2007b). On selecting middle-length feature lines for dissimilarity-based classification. In *XII Simposio de Tratamiento de Señales, Imágenes y Visión Artificial, STSIVA 2007*, Universidad del Norte. Capítulo de la Sociedad de Procesamiento de Señales, IEEE Sección Colombia; Departamento de Eléctrica y Electrónica – Fundación Universidad del Norte y Rama Estudiantil IEEE – UniNorte.

Orozco-Alzate, M., García-Ocampo, M. E., Duin, R. P. W., & Castellanos-Domínguez, C. G. (2006). Dissimilarity-based classification of seismic volcanic signals at Nevado del Ruiz volcano. *Earth Sciences Research Journal*, 10(2):57–65.

Paclík, P. & Duin, R. P. W. (2003). Dissimilarity-based classification of spectra: computational issues. *Real Time Imaging*, 9:237–244.

Paclík, P., Novovicova, J., & Duin, R. P. W. (2006a). Building road sign classifiers using trainable similarity measure. *IEEE Trans. Intelligent Transportation Systems*, 7(3):293–308.

Paclík, P., Novovicova, J., & Duin, R. P. W. (2006b). A trainable similarity measure for image classification. In Tang, Y. Y., Wang, S. P., Lorette, G., Yeung, D. S., & Yan, H., editors, *Proc. of the 18th Int. Conf. on Pattern Recognition (ICPR2006, Hong Kong, China, August 2006)*, volume 3, pages 391–394, Los Alamitos. IEEE Computer Society Press.

Paredes, R. & Vidal, E. (2000). A class-dependent weighted dissimilarity measure for nearest neighbor classification problems. *Pattern Recogn. Lett.*, 21(12):1027–1036.

Pękalska, E. & Duin, R. P. W. (2002). Dissimilarity representations allow for building good classifiers. *Pattern Recognition Lett.*, 23:943–956.

Pękalska, E. & Duin, R. P. W. (2005a). *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*, volume 64 of *Machine Perception and Artificial Intelligence*. World Scientific, Singapore.

Pękalska, E. & Duin, R. P. W. (2005b). *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*, volume 64 of *Machine Perception and Artificial Intelligence*, chapter 3, pages 89–145. World Scientific, Singapore.

Pękalska, E. & Duin, R. P. W. (2005c). Further data exploration. In *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*, chapter 7, pages 289–332. World Scientific.

Pękalska, E., Duin, R. P. W., Günter, S., & Bunke, H. (2004). On not making dissimilarities Euclidean. In *Proceedings of Structural and Statistical Pattern Recognition*, pages 1143–1151, Lisbon, Portugal.

Pękalska, E., Duin, R. P. W., & Paclík, P. (2006). Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, 39(2):189–208.

Pękalska, E., Paclík, P., & Duin, R. P. W. (2001). A generalized kernel approach to dissimilarity based classification. *J. Mach. Learn. Res.*, 2(2):175–211.

Ruiz-del Solar, J. & Navarrete, P. (2005). Eigenspace-based face recognition: a comparative study of different approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 35(3):315–325.

Sánchez, J. S., Pla, F., & Ferri, F. J. (1998). Improving the k-NCN classification rule through heuristic modifications. *Pattern Recognition Letters*, 19(13):1165–1170.

Tomek, I. (1976). Two modifications of CNN. *IEEE Transactions on Systems, Man, and Cybernetics*, 6:679–772.

Toussaint, G. T., Bhattacharya, B. K., & Poulsen, R. S. (1985). The application of Voronoi diagrams to nonparametric decision rules. In Billard, L., editor, *Computer Science and Statistics: The Interface*. Elsevier, Amsterdam.

Uspensky, J. V. (1948). *Theory of Equations*. McGraw-Hill.

Wang, J., Neskovic, P., & Cooper, L. N. (2005). An adaptive nearest neighbor rule for classifications. In Yeung, D. S., Liu, Z. Q., Wang, X. Z., & Yan, H., editors, *Advances in Machine Learning and Cybernetics. Proceedings of the 4th International Conference on Machine Learning and Cybernetics ICMLC2005. Guangzhou, China*, volume 3930 of *Lecture Notes in Computer Science. Sublibrary: LNAI*, pages 548–557, Berlin. Springer.

Wang, J., Neskovic, P., & Cooper, L. N. (2006). Neighborhood size selection in the k-nearest-neighbor rule using statistical confidence. *Pattern Recognition*, 39(3):417–423.

Wang, J., Neskovic, P., & Cooper, L. N. (2007). Improving nearest neighbor rule with a simple adaptive distance measure. *Pattern Recogn. Lett.*, 28(2):207–213.

Weinberger, K., Blitzer, J., & Saul, L. K. (2005). Distance metric learning for large margin nearest neighbor classification. In Weiss, Y., Schölkopf, B., & Platt, J., editors, *Advances in Neural Information Processing Systems. Proceedings of the 18th Annual Conference on Neural Information Processing Systems NIPS2005*, volume 18, pages 1473–1480, Cambridge, MA. Neural Information Processing Systems Foundation, MIT Press.

Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, 2(3):408–421.

Wilson, D. R. & Martínez, T. R. (1997). Improved heterogeneous distance functions. *J. Artif. Intell. Res. (JAIR)*, 6:1–34.

Wilson, D. R. & Martínez, T. R. (2000). Reduction techniques for instance-based learning algorithms. *Mach. Learn.*, 38(3):257–286.

Zhao, H. T. & Yuen, P. C. (2007). Incremental linear discriminant analysis for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 38(1):210–221.

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

Zheng, W., Zhao, L., & Zou, C. (2004). Locally nearest neighbor classifiers for pattern classification. *Pattern Recognition*, 37:1307–1309.

Zhou, Z., Li, S. Z., & Chan, K. L. (2000). A theoretical justification of nearest feature line method. In *Proceedings of the 15th International Conference on Pattern Recognition ICPR2000, Barcelona, Spain*, volume II, pages 2759–2762. IAPR, IEEE Computer Society.

Subspace Methods for Face Recognition: Singularity, Regularization, and Robustness

Wangmeng Zuo, Kuanquan Wang and Hongzhi Zhang
Harbin Institute of Technology
China

1. Introduction

Face recognition has been an important issue in computer vision and pattern recognition over the last several decades (Zhao et al., 2003). While human can recognize faces easily, automated face recognition remains a great challenge in computer-based automated recognition research. One difficulty in face recognition is how to handle the variations in expression, pose and illumination when only a limited number of training samples are available.

Currently, face recognition methods can be grouped into three categories, feature-based, holistic-based, and hybrid approaches (Zhao et al., 2003). Feature-based approaches, which extract local features such as the locations and local statistics of the eyes, nose, and mouth, had been investigated in the beginning of the face recognition research (Kanade, 1973). Recently, with the introduction of elastic bunch graph matching (Wiskott, 1997) and local binary pattern (Timo, 2004), local feature-based approaches have shown promising results in face recognition. Holistic-based approaches extract a holistic representation of the whole face region, and have robust recognition performance under noise, blurring, and partial occlusion. After the introduction of Eigenfaces (Turk & Pentland, 1991) and Fisherfaces (Belhumeur et al., 1997), holistic-based approaches were extensively studied and widely applied to face recognition. Motivated by human perception system, hybrid approaches use both local feature and the whole face region for face recognition, and thus are expected to be potentially effective in improving recognition accuracy.

In holistic-based face recognition, feature extraction is fundamental, which can be revealed from three aspects. First, the input facial image is high dimensional and most current recognition approaches suffer from the "curse of dimensionality" problem. Thus a feature extraction step is necessary. Second, facial image usually contains less discriminative or unfavorable information for recognition (e.g., illumination). By making use of feature extraction, this information can be efficiently suppressed while retaining discriminative information. Third, feature extraction can greatly reduce the dimensionality of facial image, and this reduces the system's memory and computational requirements.

Subspace method, which aims to reduce the dimension of the data while retaining the statistical separation property between distinct classes, has been a natural choice for facial feature extraction. Face images, however, are generally high dimensional and their within-class variations is much larger than the between-class variations, which will cause the serious performance degradation of classical subspace methods. By far, various subspace methods have been proposed and applied to face recognition.

1.1 Previous work

At the beginning, linear unsupervised method, such as principal component analysis (PCA), was used to extract the holistic feature vectors for facial image representation and recognition (Turk & Pentland, 1991). Other unsupervised methods, such as independent component analysis (ICA) and non-negative matrix factorization (NMF), have been subsequently applied to face recognition (Bartlett et al., 2002; Zafeiriou et al., 2006).

Since the unsupervised methods do not utilize the class label information in the training stage, it is generally believed that the supervised methods are more effective in dealing with recognition problems. Fisher linear discriminant analysis (LDA), which aims to find a set of optimal discriminant vectors that map the original data into a low-dimensional feature space, is then gaining popularity in face recognition. In 1996, Fisher linear discriminant analysis was applied to face recognition, and subsequently was developed into one of the most famous face recognition approaches, Fisherfaces (Swets & Weng, 1996; Belhumeur et al., 1997). In face recognition, the data dimensionality is much higher than the size of the training set, leading to the small sample size problem (the SSS problem). Currently there are two popular strategies to solve the SSS problem, the transform-based and the algorithm-based. The transform-based strategy first reduces the dimensions of the original image data and then uses LDA for feature extraction, while the algorithm-based strategy finds an algorithm to circumvent the SSS problem (Yang & Yang, 2003; Yu & Yang, 2001).

Face recognition usually is highly complex and can not be regarded as a linear problem. In the last few years, a class of nonlinear discriminant analysis techniques named as kernel discriminant analysis has been widely investigated for face recognition. A number of kernel-methods, such as kernel principal component analysis (KPCA), kernel Fisher's discriminant analysis, complete kernel Fisher discriminant (CKFD), and kernel direct discriminant analysis (KDDA), have been developed (Liu, 2004; Yang, 2002; Yang et al., 2005b; Lu et al., 2003). Most recently, manifold learning methods, such as isometric feature mapping (ISOMAP), locally linear embedding (LLE), and Laplacian eigenmaps, have also shown great potential in face recognition (Tenenbaum et al., 2000; Roweis & Saul, 2000; He et al., 2005).

As a generalization of vector-based methods, a number of tensor discrimination technologies have been proposed. The beginning of tensor discrimination technology can be traced back to 1993, where a 2D image matrix based algebraic feature extraction method is proposed for image recognition (Liu et al., 1993). As a new development of the 2D image matrix based straightforward projection technique, a two-Dimensional PCA (2DPCA) approach was suggested for face representation and recognition (Yang et al., 2004). To further reduce computational cost, researchers had developed several BDPCA and generalized low rank approximations of matrices (GLRAM) approaches (Ye, 2004; Zuo et al., 2005a). Motivated by multilinear generalization of singular vector decomposition (Lathauwer et al., 2000), a number of alterative supervised and unsupervised tensor analysis methods have been proposed for facial image or image sequence feature extraction (Tao et al., 2005; Yan et al., 2007).

1.2 Organization of this chapter

Generally, there are three issues which should be addressed in the development of subspace methods for face recognition, singularity, regularization, and robustness. First, the dimensionality of facial image usually is higher than the size of the available training set,

which results in the singularity of the scatter matrices and causes the performance degradation (known as the SSS problem). So far, considerable research interests have been given to solve the SSS problem. Second, another unfavorable effect of the SSS problem is that, a limited sample size can cause poor estimation of the scatter matrices, resulting in an increase in the classification error. Third, noisy or partially occluded facial image may be inevitable during the capture and communication stage, and thus the robust recognition should be addressed in the development of subspace methods.

In this chapter, we introduce the recent development of subspace-based face recognition methods in addressing these three problems. First, to address the singularity problem, this chapter proposes a fast feature extraction technique, Bi-Directional PCA plus LDA (BDPCA+LDA), which performs LDA in the BDPCA subspace. Compared with the PCA+LDA framework, BDPCA+LDA needs less computational and memory requirements, and can achieve competitive recognition accuracy. Second, to alleviate the over-fitting to the training set, this chapter suggests a post-processing approach on discriminant vectors, and theoretically demonstrates its relationship with the image Euclidean distance method (IMED). Third, to improve the robustness of subspace method over noise and partial occlusion, this chapter presents an iteratively reweighted fitting of the Eigenfaces method (IRF-Eigenfaces), which first defines a generalized objective function and then uses the iteratively reweighted least-squares (IRLS) fitting algorithm to extract the feature vector by minimizing the generalized objective function. Finally, two popular face databases, the AR and the FERET face databases, are used to evaluate the performance the proposed subspace methods.

2. BDPCA+LDA: a novel method to address the singular problem

In face recognition, classical LDA always encounters the SSS problem, where the data dimensionality is much higher than the size of the training set, leading to the singularity of the within-class scatter matrix $\mathbf{S}_w$. A number of approaches have been proposed to address the SSS problem. One of the most successful approaches is subspace LDA which uses a dimensionality reduction technique to map the original data to a low-dimensional subspace. Researchers have applied PCA, latent semantic indexing (LSI), and partial least squares (PLS) as pre-processors for dimensionality reduction (Belhumeur et al., 1997; Torkkola, 2001; Baeka & Kimb, 2004). Among all the subspace LDA methods, over the past decade, the PCA plus LDA approach (PCA+LDA), where PCA is first applied to eliminate the singularity of $\mathbf{S}_w$, and then LDA is performed in the PCA subspace, has received significant attention (Belhumeur et al., 1997). The discarded null space of $\mathbf{S}_w$, however, may contain some important discriminant information and cause the performance deterioration of Fisherfaces. Rather than discarding the null space of $\mathbf{S}_w$, Yang proposed a complete PCA+LDA method which simultaneously considered the discriminant information both in the range space and the null space of $\mathbf{S}_w$ (Yang & Yang, 2003).

In this section, we introduce a fast subspace LDA technique, Bi-Directional PCA plus LDA (BDPCA+LDA). BDPCA, which assumes that the transform kernel of PCA is separable, is a natural extension of classical PCA and a generalization of 2DPCA (Yang et al., 2004). The separation of the PCA kernel has at least three main advantages: lower memory requirement, faster training and feature extraction speed.

2.1 Linear discriminant analysis

Let $\mathbf{M}$ be a set of data, $\mathbf{M} = \{\mathbf{x}_{11}, \cdots, \mathbf{x}_{1n_1}, \cdots, \mathbf{x}_{ij}, \cdots, \mathbf{x}_{Cn_C}\}$, where $\mathbf{x}_{ij}$ is the jth training sample of the ith class, and n_i is the number of samples of the ith class, C is the number of classes. The sample $\mathbf{x}_{ij}$ is a one-dimensional vector or a vector representation of the corresponding image $\mathbf{X}_{ij}$. LDA and PCA are two classical dimensionality reduction techniques. PCA, an optimal representation method in a minimization of mean-square error sense, has been widely used for the representation of shape, appearance, and video (Jolliffe, 2001). LDA is a linear dimensionality reduction technique which aims to find a set of the optimal discriminant vectors by maximizing the class separability criterion (Fukunaga, 1990). In the field of face recognition, LDA is usually assumed more effective than PCA because LDA aims to find the optimal discriminant directions.

Two main tasks in LDA are calculation of the scatter matrices, and selection of the class separability criterion. Most LDA algorithms involve the simultaneous maximization of the trace of a scatter matrix and minimization of the trace of another matrix. LDA usually makes use of two scatter matrices, such as the within-class scatter matrix $\mathbf{S}_w$ and the between-class scatter matrix $\mathbf{S}_b$. The within-class scatter matrix $\mathbf{S}_w$, the scatter of samples around their class mean vectors, is defined as

$$\mathbf{S}_w = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{n_i} (\mathbf{x}_{ij} - \overline{\mathbf{x}}_i)(\mathbf{x}_{ij} - \overline{\mathbf{x}}_i)^T . \tag{1}$$

The between-class scatter matrix $\mathbf{S}_b$, the scatter of class mean vectors around the global mean vector, is defined as

$$\mathbf{S}_b = \frac{1}{N} \sum_{i=1}^{C} n_i (\overline{\mathbf{x}}_i - \overline{\mathbf{x}})(\overline{\mathbf{x}}_i - \overline{\mathbf{x}})^T , \tag{2}$$

where $\overline{\mathbf{x}} = \sum_{i=1}^{C} \sum_{j=1}^{n_i} \mathbf{x}_{ij} / N$ is the global mean vector, $\overline{\mathbf{x}}_i = \sum_{j=1}^{n_i} \mathbf{x}_{ij} / n_i$ is the mean vector of class i, and $N = \sum_{i=1}^{C} n_i$ is the total number of training samples.

The most famous class separability criterion is the Fisher's discriminant criterion

$$J_F(\mathbf{w}) = \frac{\mathbf{w} \mathbf{S}_b \mathbf{w}}{\mathbf{w} \mathbf{S}_w \mathbf{w}} . \tag{3}$$

The set of discriminant vectors $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_{d_{LDA}}]$ corresponding to the maximization of the Fisher's discriminant criterion can be obtained by solving the generalized eigenvalue problem $\mathbf{S}_b \mathbf{W} = \mathbf{S}_w \mathbf{W} \mathbf{\Lambda}$. Since both $\mathbf{S}_b$ and $\mathbf{S}_w$ are symmetric matrices, the *simultaneous diagonalization* technique can be used to calculate the set of discriminant vectors $\mathbf{W}$.

2.1.1 Simultaneous diagonalization

Fig. 1 uses a three-class problem to illustrate the procedure of simultaneous diagonalization in computing the discriminant vectors of LDA. The distribution of each class and the distributions of within- and between-class scatter are depicted in Fig. 1(a) and (b). Simultaneous diagonalization tries to find a transformation matrix $\mathbf{\Phi}$ that satisfies $\mathbf{\Phi}^T \mathbf{S}_w \mathbf{\Phi} = \mathbf{I}$ and $\mathbf{\Phi}^T \mathbf{S}_b \mathbf{\Phi} = \mathbf{\Lambda}_g$, where $\mathbf{I}$ is an identity matrix and $\mathbf{\Lambda}_g$ is a diagonal matrix.
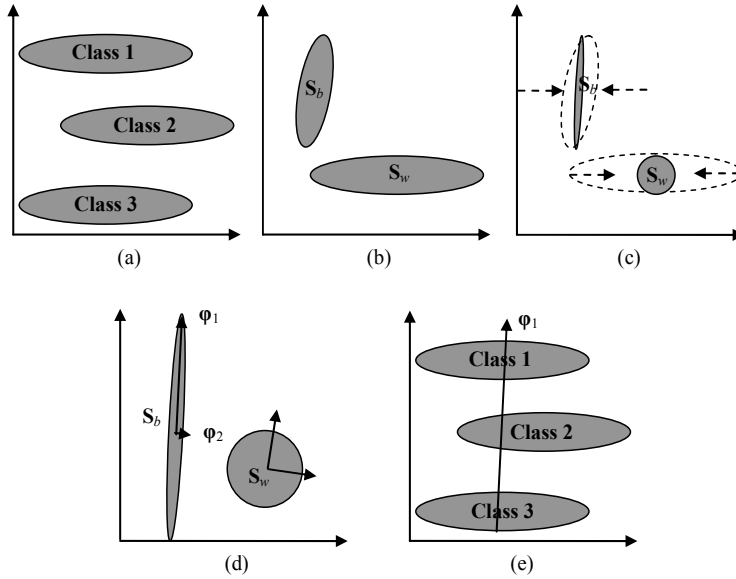
Fig. 1. Procedure of simultaneous diagonalization: (a) The distributions of the three-class problem, (b) The within-and between-class distributions, (c) Whitening the within-class distribution, and correspondingly transform the between-class distribution. (d) Calculating the eigenvectors of the transformed between-class distribution. (e) Illustration of the first discriminant vector.

The procedure of simultaneous diagonalization contains three steps:

Step 1. Whitening $\mathbf{S}_w$. PCA is used to whiten the within-class distribution to an isotropic distribution by a transformation matrix $\mathbf{\Theta}_{wh}$. Then, matrix $\mathbf{\Theta}_{wh}$ is used to transform the between-class scatter $\hat{\mathbf{S}}_b = \mathbf{\Theta}_{wh}{}^T \mathbf{S}_b \mathbf{\Theta}_{wh}$.

Step 2. Calculation of the eigenvectors $\mathbf{\Psi}$ and eigenvalues $\mathbf{\Lambda}_g$ of $\hat{\mathbf{S}}_b$.

Step 3. Computation of the transformation matrix $\mathbf{\Phi} = \mathbf{\Theta}_{wh}\mathbf{\Psi}$, where $\mathbf{\Phi} = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2]$ is the set of generalized eigenvectors of $\mathbf{S}_w$ and $\mathbf{S}_b$.

2.2 BDPCA+LDA: algorithm
2.2.1 Bi-directional PCA

To simplify our discussion, in the following, we adopt two representations of an image, $\mathbf{X}$ and $\mathbf{x}$, where $\mathbf{X}$ is a representation of an image matrix and $\mathbf{x}$ is a representation of an image vector. $\mathbf{X}$ and $\mathbf{x}$ represent the same image.

Given a transform kernel (e.g., principal component vector) $\mathbf{w}_i$, an image vector $\mathbf{x}$ can be projected into $\mathbf{w}_i$ by $y_i = \mathbf{w}_i^T \mathbf{x}$. In image transform, if the transform kernel is *product-separable*, the image matrix $\mathbf{X}$ can be projected into $\mathbf{w}_i$ equivalently by $y_i = \mathbf{w}_{i,C}^T \mathbf{X} \mathbf{w}_{i,R}$, where $\mathbf{w}_{i,C}$ and $\mathbf{w}_{i,R}$ are the corresponding column transform kernel and row transform kernel of $\mathbf{w}_i$. In PCA, assuming all the eigenvectors $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_d]$ are *product-separable*, there are two equivalent ways to extract the feature of an image $\mathbf{x}$, $\mathbf{y} = \mathbf{w}^T \mathbf{x}$ (vector-based way) and $\mathbf{Y} = \mathbf{W}_C^T \mathbf{X} \mathbf{W}_R$ (matrix-based way), where $\mathbf{W}_C$ and $\mathbf{W}_R$ are the column and row projection matrices.

2DPCA assumes the column projection matrix $\mathbf{W}_C$ is an $m\times m$ identity matrix, and the criterion of classical PCA will degenerate to

$$J(\mathbf{w}) = \mathbf{w}^T \mathbf{G}_t \mathbf{w} , \tag{4}$$

where $\mathbf{w}$ is a unitary column vector, $\mathbf{w}^T\mathbf{w} = 1$, and $\mathbf{G}_t$ is the image covariance matrix defined as $\mathbf{G}_t = \sum_{i=1}^{N}(\mathbf{X}_i - \bar{\mathbf{X}})^T(\mathbf{X}_i - \bar{\mathbf{X}})$. Compared with PCA, 2DPCA has several significant advantages. First, 2DPCA is simpler and more straightforward to use for image feature extraction. Second, experimental results consistently show that 2DPCA is better than PCA in terms of recognition accuracy. Third, 2DPCA is computationally more efficient than PCA and significantly improve the speed of image feature extraction (Yang et al., 2004).

Bi-Directional PCA (BDPCA) extracts representative feature from image $\mathbf{X}$ by $\mathbf{Y} = \mathbf{W}_C^T \mathbf{X} \mathbf{W}_R$ yet it is difficult to simultaneously determine optimal $\mathbf{W}_C$ and $\mathbf{W}_R$ in an analytic framework. However, a number of alternative approaches have been proposed to compute the optimal column and row projection matrices $\mathbf{W}_C$ and $\mathbf{W}_R$. In the following, we summary the three main strategies for dealing with this:

1. The Hierarchical Strategy (Yang et al., 2005a). Hierarchical strategy adopts a two-step framework to calculate $\mathbf{W}_C$ and $\mathbf{W}_R$. First a 2DPCA is performed in horizontal direction and the second 2DPCA is performed on the row-compressed matrix in vertical direction (**H1**), as shown in Fig. 2(a). It is obvious that we can adopt an alternative method, first perform 2DPCA in vertical direction and then in horizontal direction (**H2**).
2. The Iterative Strategy. In (Ye, 2005), Ye proposed an iterative procedure for computing $\mathbf{W}_C$ and $\mathbf{W}_R$. After the initialization of $\mathbf{W}_{C0}$, the procedure repeatedly first updates $\mathbf{W}_R$ according to $\mathbf{W}_C$, and then updates $\mathbf{W}_C$ according to $\mathbf{W}_R$ until convergence (**I1**), as shown in Fig. 2(b). Theoretically, this procedure can only be guaranteed to be convergent to locally optimal solution of $\mathbf{W}_C$ and $\mathbf{W}_R$. Their experimental results also show that, for image data with some hidden structure, the iterative algorithm may converge to the global solution, but this assertion does not always hold.
3. The Independence Assumption (Zuo et al., 2006). One disadvantage of the hierarchical strategy is that are always confronted with the choice of **H1** or **H2**. Assuming that the computing of $\mathbf{W}_R$ and the computing of $\mathbf{W}_C$ are independent, $\mathbf{W}_C$ and $\mathbf{W}_R$ can be computed by solving two 2DPCA problems independently (**I2**), as shown in Fig. 2(c). Experimental results show that, in facial feature extraction, **H1**, **H2**, **I1** and **I2** have similar recognition performance, and **H1**, **H2**, and **I2** require less training time.

In the following, we use the third strategy to explain the procedure of BDPCA. Given a training set $\{\mathbf{X}_1, \cdots, \mathbf{X}_N\}$, N is the number of the training images, and the size of each image matrix is $m\times n$. By representing the ith image matrix $\mathbf{X}_i$ as an m-set of *1×n* row vectors

$$\mathbf{X}_i = \begin{bmatrix} \mathbf{x}_i^1 \\ \mathbf{x}_i^2 \\ \vdots \\ \mathbf{x}_i^m \end{bmatrix}, \tag{5}$$

we adopt Yang's approach (Yang et al, 2004) to define the row total scatter matrix
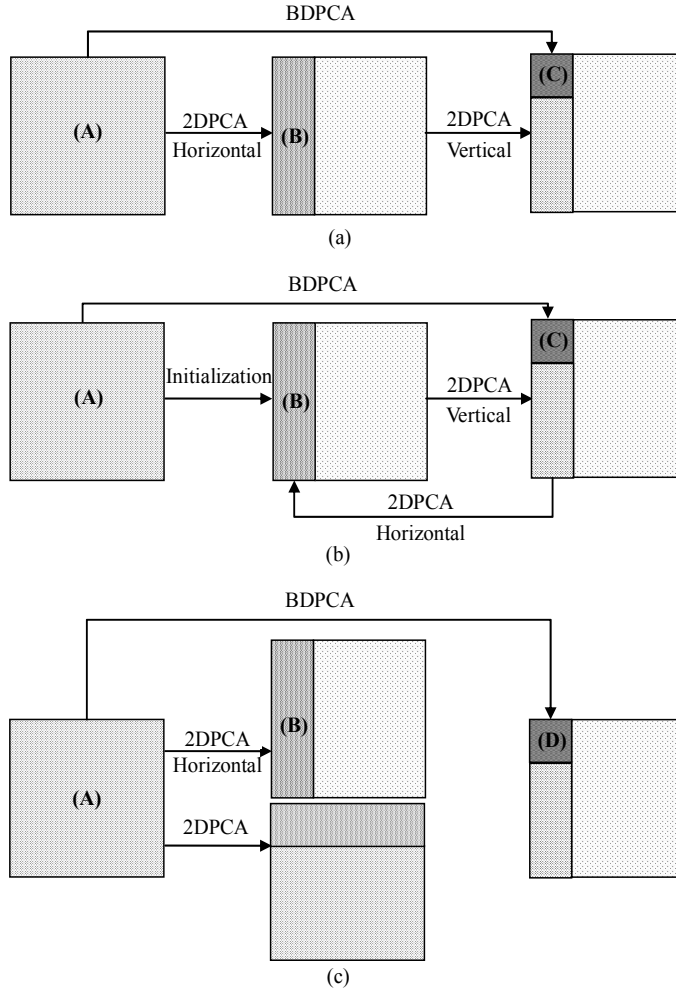
Fig. 2. Illustration of the three strategies in calculating the column and row transformation matrix

$$\mathbf{S}_t^{row} = \frac{1}{Nm}\sum_{i=1}^{N}\sum_{j=1}^{m}(\mathbf{x}_i^j - \overline{\mathbf{x}}^j)^T(\mathbf{x}_i^j - \overline{\mathbf{x}}^j) = \frac{1}{Nm}\sum_{i=1}^{N}(\mathbf{X}_i - \overline{\mathbf{X}})^T(\mathbf{X}_i - \overline{\mathbf{X}}),\qquad(6)$$

where $\mathbf{x}_i^j$ and $\overline{\mathbf{x}}_i^j$ denotes the jth row of sample $\mathbf{X}_i$ and mean matrix $\overline{\mathbf{X}}$, respectively. We choose the row eigenvectors corresponding to the first k_{row} largest eigenvalues of $\mathbf{S}_t^{row}$ to construct the row projection matrix $\mathbf{W}_r$

$$\mathbf{W}_r = [\mathbf{w}_1^{row}, \mathbf{w}_2^{row}, \cdots, \mathbf{w}_{k_{\text{row}}}^{row}],\qquad(7)$$

where $\mathbf{w}_i^{row}$ denotes the row eigenvector corresponding to the ith largest eigenvalues of $\mathbf{S}_t^{row}$. Similarly, by treating an image matrix $\mathbf{X}_i$ as an n-set of $m\times 1$ column vectors

$$\mathbf{X}_i = [\mathbf{x}_i^1 \quad \mathbf{x}_i^2 \quad \cdots \quad \mathbf{x}_i^n],\qquad(8)$$

we define the column total scatter matrix

$$\mathbf{S}_t^{col} = \frac{1}{Nn}\sum_{i=1}^{N}(\mathbf{X}_i - \bar{\mathbf{X}})(\mathbf{X}_i - \bar{\mathbf{X}})^T . \tag{9}$$

We then choose the column eigenvectors corresponding to the first k_{col} largest eigenvalues of $\mathbf{S}_t^{col}$ to construct the column projection matrix $\mathbf{W}_c$

$$\mathbf{W}_c = [\mathbf{w}_1^{col}, \mathbf{w}_2^{col}, \cdots, \mathbf{w}_{k_{col}}^{col}] , \tag{10}$$

where $\mathbf{w}_i^{col}$ is the column eigenvector corresponding to the ith largest eigenvalues of $\mathbf{S}_t^{col}$. Finally we use the transformation

$$\mathbf{Y} = \mathbf{W}_c^T \mathbf{X} \mathbf{W}_r , \tag{11}$$

to extract the feature matrix $\mathbf{Y}$ of image matrix $\mathbf{X}$.

2.2.2 BDPCA+LDA

BDPCA+LDA is an LDA approach that is applied on a low-dimensional BDPCA subspace, and thus can be used for fast facial feature extraction. Since less time is required to map an image matrix to BDPCA subspace, BDPCA+LDA is, at least, computationally faster than PCA+LDA.

BDPCA+LDA first uses BDPCA to obtain feature matrix $\mathbf{Y}$. The feature matrix $\mathbf{Y}$ is then transformed into feature vector $\mathbf{y}$ by concatenating the columns of $\mathbf{Y}$. The LDA projector $\mathbf{W}_{LDA} = [\varphi_1, \varphi_2, \cdots, \varphi_m]$ is calculated by maximizing Fisher's criterion:

$$J(\varphi) = \frac{\varphi^T \mathbf{S}_b \varphi}{\varphi^T \mathbf{S}_w \varphi} , \tag{12}$$

where φ_i is the generalized eigenvector of $\mathbf{S}_b$ and $\mathbf{S}_w$ corresponding to the ith largest eigenvalue λ_i

$$\mathbf{S}_b \varphi_i = \lambda_i \mathbf{S}_w \varphi_i , \tag{13}$$

and $\mathbf{S}_b$ is the between-class scatter matrix of $\mathbf{y}$

$$\mathbf{S}_b = \frac{1}{N}\sum_{i=1}^{C} N_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T , \tag{14}$$

and $\mathbf{S}_w$ is the within-class scatter matrix of $\mathbf{y}$,

$$\mathbf{S}_w = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{N_i}(\mathbf{y}_{i,j} - \boldsymbol{\mu}_i)(\mathbf{y}_{i,j} - \boldsymbol{\mu}_i)^T , \tag{15}$$

where N_i, $\mathbf{y}_{i,j}$ and $\boldsymbol{\mu}_i$ are the number of feature vectors, the jth feature vector and the mean vector of class i, C is the number of classes, and $\boldsymbol{\mu}$ is the mean vector of all the feature vectors.

In summary, the main steps in BDPCA+LDA feature extraction are to first transform an image matrix **X** into BDPCA feature subspace **Y** by Eq. (11), and map **Y** into its 1D representation **y** and then to obtain the final feature vector **z** by

$$\mathbf{z} = \mathbf{W}_{\mathrm{LDA}}^{\mathrm{T}}\mathbf{y} \,. \tag{16}$$

2.2.3 Advantages over the existing PCA plus LDA framework

We compare the BDPCA+LDA and the PCA+LDA face recognition frameworks in terms of their computational and memory requirements. It is worth noting that the computational requirements are considered in two phases, training and testing.

Method	Memory Requirements		Computation Requirements	
	Projector	Feature prototypes	Training	Testing
PCA+LDA	$(m{\times}n){\times}d_{\mathrm{LDA}}$ **Large**	$N{\times}d_{\mathrm{LDA}}$ **Same**	a) Calculating the projector: O($N_{\mathrm{p}}^3+d_{\mathrm{PCA}}^3$) **Large** b) Projection: $N{\times}(m{\times}n){\times}d_{\mathrm{LDA}}$ **Large**	c) Projection: $(m{\times}n){\times}d_{\mathrm{LDA}}$ **Large** d) Distance calculation: $N{\times}d_{\mathrm{LDA}}$ **Same**
BDPCA +LDA	$m{\times}k_{\mathrm{row}}+$ $n{\times}k_{\mathrm{col}}+$ $k_{\mathrm{col}}{\times}k_{\mathrm{row}}{\times}d_{\mathrm{LDA}}$ **Small**	$N{\times}d_{\mathrm{LDA}}$ **Same**	a) Calculating the projector: O($m^3+n^3+d_{\mathrm{BDPCA}}^3$) **Small** b) Projection: $N{\times}[m{\times}n{\times}\min(k_{\mathrm{row}}, k_{\mathrm{col}})$ $+k_{\mathrm{col}}{\times}k_{\mathrm{row}}{\times}\max(m+d_{\mathrm{LDA}},$ $n+d_{\mathrm{LDA}})]$ **Small**	c) Projection: $m{\times}n{\times}\min(k_{\mathrm{row}}, k_{\mathrm{col}})+$ $k_{\mathrm{col}}{\times}k_{\mathrm{row}}{\times}[\max(m,n)+d_{\mathrm{LDA}}]$ **Small** d) Distance calculation: $N{\times}d_{\mathrm{LDA}}$ **Same**

Table 1. Comparisons of computational and memory requirements of BDPCA+LDA and PCA+LDA

We first compare the computational requirement using the number of multiplications as a measurement of computational complexity. The training phase involves two computational tasks: a) calculation of the projector, and b) projection of images into feature prototypes. To calculate the projector, the PCA+LDA method must solve an $N{\times}N$ eigenvalue problem and then a $d_{\mathrm{PCA}}{\times}d_{\mathrm{PCA}}$ generalized eigenvalue problem, where N is the size if the training set and d_{PCA} is the dimension of the PCA subspace. In contrast, BDPCA+LDA must solve an $m{\times}m$, an $n{\times}n$ eigenvalue problem and a $d_{\mathrm{BDPCA}}{\times}d_{\mathrm{BDPCA}}$ generalized eigenvalue problem, where d_{BDPCA} is the dimension of BDPCA subspace. Since the complexity of an $M{\times}M$ eigenvalue problem is O(M^3), the complexity of the PCA+LDA projector-calculation operation is O($N^3+d_{\mathrm{PCA}}^3$) whereas that of BDPCA+LDA is O($m^3+n^3+d_{\mathrm{BDPCA}}^3$). Assuming that m, n, d_{PCA} and d_{BDPCA} are smaller than the number of training samples N, in calculating the projector, BDPCA+LDA requires less computation than PCA+LDA to calculate the projector.

To project images into feature prototypes, we assume that the feature dimension of BDPCA+LDA and PCA+LDA is the same, d_{LDA}. For PCA+LDA, the number of multiplications is thus $N_{\mathrm{p}}{\times}(m{\times}n){\times}d_{\mathrm{LDA}}$. For BDPCA+LDA, the number of multiplications is less than $N_{\mathrm{p}}{\times}(m{\times}n{\times}\min(k_{\mathrm{row}},k_{\mathrm{col}})+(k_{\mathrm{col}}{\times}k_{\mathrm{row}}){\times}\max(m+d_{\mathrm{LDA}}, n+d_{\mathrm{LDA}}))$, where N_{p} is the number

of prototypes. In this paper, we use all the prototypes for training, thus N_p=N. Assuming that $\min(k_{row}, k_{col})$ is much less than d_{LDA}, in the projection process, BDPCA+LDA also requires less computation than PCA+LDA.

In the test phase, there are two computational tasks: c) the projection of images into the feature vector, and d) the calculation of the distance between the feature vector and feature prototypes. In the following we compare the computational requirement of BDPCA+LDA and PCA+LDA in carrying out these two tasks. When projecting images into feature vectors, BDPCA+LDA requires less computation than PCA+LDA. Because the feature dimension of BDPCA+LDA and PCA+LDA is the same, in the similarity measure process, the computational complexity of BDPCA+LDA and PCA+LDA are equal. Taking these two tasks into account, BDPCA+LDA is also less computationally expensive than PCA+LDA in the testing phase.

The memory requirements of the PCA+LDA and BDPCA+LDA frameworks mainly depend on the size of the projector and the total size of the feature prototypes. The size of the projector of PCA+LDA is $d_{LDA} \times m \times n$. This is because the PCA+LDA projector contains d_{LDA} Fisherfaces, each of which is the same size as the original image. The BDPCA+LDA projector is in three parts, $\mathbf{W}_c$, $\mathbf{W}_r$ and $\mathbf{W}_{LDA}$. The total size of the BDPCA+LDA projector is $(k_{col} \times m) + (k_{row} \times n) + (d_{LDA} \times k_{col} \times k_{row})$, which is generally much smaller than that of PCA+LDA. Finally, because these two methods have the same feature dimensions, BDPCA+LDA and PCA+LDA have equivalent feature prototype memory requirements.

We have compared the computational and memory requirements of the BDPCA+LDA and PCA+LDA frameworks, as listed in Table 1. Generally, the BDPCA+LDA framework is superior to the PCA+LDA in both the computational and memory requirements.

2.3 BDPCA+LDA: experimental results

To evaluate the efficacy of BDPCA+LDA we make use of the FERET face database. The FERET face database is a US Department of Defense-sponsored face database and is one of the standard databases used in testing and evaluating face recognition algorithms (Phillips, 1998; Phillips et al., 2000). For our experiments, we chose a subset of the FERET database. This subset includes 1,400 images of 200 individuals (each individual contributing seven images). The seven images of each individual consist of three front images with varied facial expressions and illuminations, and four profile images ranging from ±15° to ±25° pose. The facial portion of each original image was cropped to a size of 80×80 and pre-processed using histogram equalization. Fig. 3 illustrates the seven images of one person and their corresponding cropped images.

We also compare BDPCA+LDA with other LDA-based methods, including Fisherfaces, Enhanced Fisher discriminant Model (EFM) (Liu & Wechsler, 1998), Discriminant Common Vectors (DCV) (Cevikalp et al., 2005), and D-LDA. The experimental setup is as follows. Since our aim is to evaluate the efficacy of feature extraction methods, we use a simple classifier, the nearest neighbor classifier. To reduce the variation of recognition results, we adopt the mean of 10 runs as the average recognition rate (ARR). All the experiments are carried out on an AMD 2500+ computer with 512Mb RAM and tested on the MATLAB platform (Version 6.5).

In our experiments, three images of each person are randomly chosen for training, while the remaining four images are used for testing. Thus, we obtain a training set of 600 images and a testing set of 800 images. In this way, we run the face recognition method 10 times and calculate the average recognition rate.
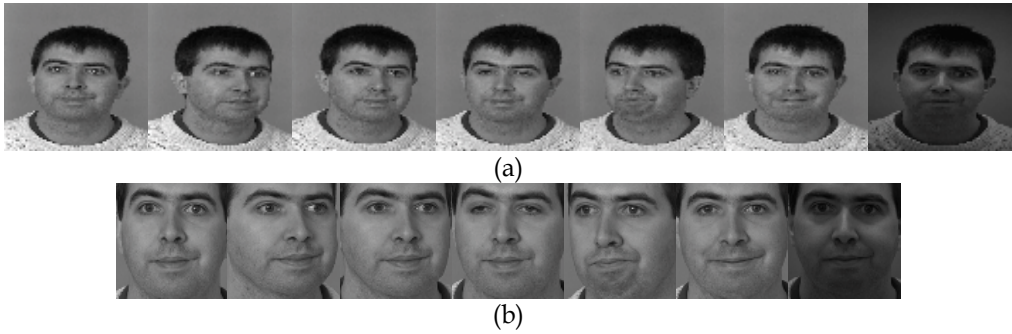
(a)

(b)

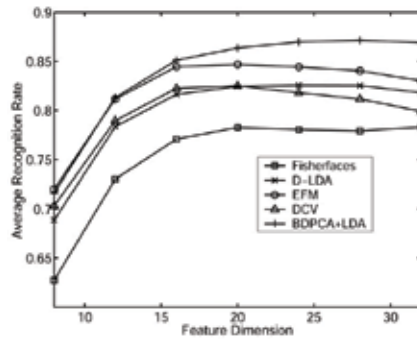Fig. 3. Images of an individual in the FERET subset: (a) the original images, and (b) the cropped images.

Fig. 4. Comparisons of the recognition rates obtained using different methods on the FERET subset.

Methods	Fisherfaces	D-LDA	EFM	DCV	BDPCA+LDA
Parameters	$[d_{PCA}, d]$	$[d_b, d]$	$[d_{PCA}, d_w, d]$	$[d]$	$[k_{col}, k_{row}, d]$
Values	[200, 20]	[60, 24]	[100, 100, 24]	[20]	[15, 5, 28]
ARR (%)	78.26%	82.56%	84.69%	82.51%	87.14%

Table 2. Recognition performance of five face recognition methods on the FERET database

Method	Time for Training (s)	Time for Testing (s)
PCA+LDA	254.2	36.2
BDPCA+LDA	57.5	26.3

Table 3. The total CPU time (s) for training and testing on the FERET database

We compare the recognition rates obtained using BDPCA+LDA, Fisherfaces, EFM, DCV and D-LDA, as shown in Fig. 4. We also list the optimal parameter values of each method and its maximum ARR in Table 2. The maximum ARR of BDPCA+LDA is 87.14%, higher than the ARRs of the other four methods.

Table 3 shows the total CPU time of PCA+LDA (EFM) and BDPCA+LDA in the training phase and the testing phase. BDPCA+LDA is much faster than EFM in both the training and testing phases.

We compare the computational and memory requirements of BDPCA+LDA and PCA+LDA (EFM). In Section 2.2.3, based on a number of assumptions, we assert that BDPCA+LDA is superior to PCA+LDA in the computational and memory requirements. We then check the

correctness of these assumptions. The size of the training set is 600, much higher than the size of row vector (80) or column vector (80). The feature dimension of EFM is 24, much higher than k_{row} (5). Thus all these assumptions are satisfied. Table 4 shows the computational and memory requirements of BDPCA+LDA and EFM. BDPCA+LDA needs less computational and memory requirements than EFM.

Method	Memory Requirements			Computation Requirements	
	Projector	Feature prototypes	Total	Training	Testing
PCA+LDA	(112×92)×39 =*401856*	200×39 =*7800*	*409656*	a) Calculating the projector: O(200³+160³) ≈*12096000* b) Projection: 200×(112×92)×39 =*80371200* Total=*92467200*	c) Projection: (112×92)×39=401856 d) Distance calculation: 200×39=*7800* Total=*409656*
BDPCA +LDA	112×12+92 ×4+(12×4) ×39 =*3584*	200×39 =*7800*	*11384*	a) Calculating the projector: O(112³+92³+(12×4)³) ≈*2294208* b) Projection: 200×[112×92×4+12×4 ×(112+39)]=*9692800* Total=*11987008*	c) Projection: 112×92×4+4×12 ×(112+39) =*48464* d) Distance calculation: 200×39 =*7800* Total=*56264*

Table 4. Comparisons of computational and memory requirements of BDPCA+LDA and PCA+LDA on the FERET subset

It should be noted that, the training complexity of BDPCA+LDA is O(N), whereas that of PCA+LDA is O(N^3), where N is the size of training set. This property implies that, when the size of the training set is high, BDPCA+LDA would be more superior to PCA+LDA in terms of computational requirement.

3. Regularization of LDA: a post-processing approach

Despite the great success of LDA in face recognition, there still exist some potential issues deserving further investigation. One is that the discriminant vectors may be over-fitted to the training set, and are very noisy and wiggly in appearance. Another disadvantage of traditional LDA is that it does not take into account the spatial relationship of pixels. Since the inaccurate location and small perturbation is unavoidable in face detection and recognition, spatial information would be helpful to improve the robustness of the recognition performance. In addressing this issue, Wang et al. (2005) proposed an image Euclidean distance (IMED) method, where a 2D-Gaussian function is used to model the effect of neighbor pixels.

In the following, we first introduce a post-processed LDA-based method, and then demonstrate the equivalence of IMED and the post-processing approach. Finally, the FERET face database is used to evaluate the performance of post-processed LDA.

3.1 Post-processed LDA

Post-processing on discriminant vectors is effective in the improvement of the recognition performance of LDA-based face recognition methods. In this section, we first briefly summarize the post-processing approach, and then present an example of the post-processing approach, post-processed enhanced Fisher's model (PEFM).

3.1.1 Post-processing approach

A post-processing approach, 2D-Gaussian filtering, has been introduced to perform on the discriminant vectors (Wang et al., 2005; Zuo et al., 2005b). 2D-Gaussian filter is an ideal filter in the sense that it reduces the magnitude of high spatial frequency in an image and has been widely applied in image smoothing and denoising. In face recognition, where the discriminant vector can be mapped to a 2D image, Gaussian filtering is used to post-process the discriminant images and reduce noise. 2D-Gaussian function is defined as

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \,, \tag{17}$$

where σ is the standard deviation. First a 2D-Gaussian model M is defined according to the standard deviation $\sigma > 0$. The window size $[w, w]$ can then be determined as $w \approx 5 \times \sigma$, and the Gaussian model M is defined as the $w \times w$ truncation from the Gaussian kernel $G(x, y)$. We then calculate the norm of the discriminant vector $\|v_i\|_2 = \sqrt{v_i^T v_i}$, and map it into the corresponding discriminant image I_i. The filter M is used to smooth the discriminant image

$$I_i'(x, y) = I(x, y) * M(x, y) \,. \tag{18}$$

$I_i'(x, y)$ is transformed into a high dimensional vector v_i' by concatenating the rows of $I_i'(x, y)$ together. Finally we normalize v_i' using the norm of v_i

$$v_i'' = \frac{\|v_i\|_2}{\sqrt{v_i'^T v_i'}} v_i' \,, \tag{19}$$

and obtain the post-processed discriminant vector v_2''.

Compared with other LDA techniques, the post-processed LDA method has some potential advantages, such as directness, two dimensionality, and complementarity. First, post-processed LDA is designed to directly modify the discriminant vectors. Other LDA techniques, such as EFM, usually adopt the strategy to define the within-class scatter matrix in PCA subspace. Second, when applied to image recognition task, post-processed LDA maps a discriminant vector into a two-dimensional image, and thus can use two-dimensional image processing techniques to alter the appearance of the discriminant vector. Third, post-processing can be used as a complementary approach to combine with other LDA techniques, such as enhanced Fisher model, and completer Fisher discriminant framework.

3.1.2 Post-processed Enhanced Fisher Model

The Enhanced Fisher Model (EFM) method is based on the PCA plus LDA framework where PCA is used to alleviate the over-fitting problem and to improve the generalization

performance (Liu & Wechsler, 1998; Liu & Wechsler, 2002). In (Wang & Tang, 2004), Wang and Tang present another insight to understand EFM by modeling face difference with intrinsic difference, transform difference, and noise, where the PCA transform is used to significantly reduce noise, and the subsequent LDA step is used to separate intrinsic difference from transform difference.

In EFM, each image should be previously mapped into a one-dimensional vector by concatenating the rows of the original image. Let $\mathbf{X} = \{\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)}, \cdots, \mathbf{x}_{N_1}^{(1)}, \cdots, \mathbf{x}_j^{(i)}, \cdots, \mathbf{x}_{N_C}^{(C)}\}$ be a training set with N_i image vectors for class i. The number of class is C, and $\mathbf{x}_j^{(i)}$ denotes the jth image vector of class i. The total covariance matrix $\mathbf{S}_t$ of PCA is then defined as

$$\mathbf{S}_t = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{N_i} (\mathbf{x}_j^{(i)} - \bar{\mathbf{x}})(\mathbf{x}_j^{(i)} - \bar{\mathbf{x}})^T \ , \tag{20}$$

where $\bar{\mathbf{x}}$ is the mean vectors of all training images, and $N = \sum_{i=1}^{C} N_i$ is the total number of training images. The PCA projector $\mathbf{T}_{pca} = [\varphi_1, \varphi_2, \cdots, \varphi_{d_{PCA}}]$ can be obtained by calculating the eigenvalues and vectors of the total scatter matrix $\mathbf{S}_t$, where φ_k is the kth eigenvector corresponding to the kth largest eigenvalue of $\mathbf{S}_t$, and d_{PCA} denotes the PCA dimension for the EFM method.

The between-class scatter matrix $\mathbf{S}_b$ and the within-class scatter matrix $\mathbf{S}_w$ are defined as

$$\mathbf{S}_b = \frac{1}{N} \sum_{i=1}^{C} N_i (\overline{\mathbf{x}^{(i)}} - \bar{\mathbf{x}})(\overline{\mathbf{x}^{(i)}} - \bar{\mathbf{x}})^T \ , \tag{21}$$

$$\mathbf{S}_w = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{N_i} (\mathbf{x}_j^{(i)} - \overline{\mathbf{x}^{(i)}})(\mathbf{x}_j^{(i)} - \overline{\mathbf{x}^{(i)}})^T \ , \tag{22}$$

where $\bar{\mathbf{x}}^{(i)}$ is mean vector of class i. With PCA projector $\mathbf{T}_{pca}$, we map $\mathbf{S}_b$ and $\mathbf{S}_w$ to the PCA subspace,

$$\breve{\mathbf{S}}_b = \mathbf{T}_{pca}^T \mathbf{S}_b \mathbf{T}_{pca} \text{ and } \breve{\mathbf{S}}_w = \mathbf{T}_{pca}^T \mathbf{S}_w \mathbf{T}_{pca} \ . \tag{23}$$

PCA projection can eliminate the singularity of the within-class scatter matrix. Thus the optimal discriminant vectors can be calculated by maximizing the Fisher's criterion

$$J_F(w) = \frac{w^T \breve{\mathbf{S}}_b w}{w^T \breve{\mathbf{S}}_w w} \ . \tag{24}$$

The discriminant vectors can be obtained by calculate the first d_{LDA} generalized eigenvectors $[w_1, w_2, \cdots, w_{d_{LDA}}]$ and the corresponding eigenvalues $[\lambda_1, \lambda_2, \cdots, \lambda_{d_{LDA}}]$ of $\breve{\mathbf{S}}_b$ and $\breve{\mathbf{S}}_w$. Given an image vector $\mathbf{x}$, the discriminant feature vector $\mathbf{z}^S$ is defined as

$$\mathbf{z}^S = \mathbf{U}_S^T \mathbf{T}_{SPCA}^T \mathbf{x} \ , \tag{25}$$

where $\mathbf{U}_S = [w_1, w_2, \cdots, w_{d_{LDA}}]$ is the subspace LDA projector.

> **PEFM Algorithm**
>
> **Step 1.** Compute $\mathbf{S}_t$, $\mathbf{S}_b$, and $\mathbf{S}_w$, and calculate the first d_{PCA} eigenvectors
> $\mathbf{T}_{pca} = [\varphi_1, \varphi_2, \cdots, \varphi_{d_{PCA}}]$ of $\mathbf{S}_t$, then modify $\mathbf{S}_b$ and $\mathbf{S}_w$ by $\bar{\mathbf{S}}_b = \mathbf{T}_{pca}^T \mathbf{S}_b \mathbf{T}_{pca}$
> and $\bar{\mathbf{S}}_w = \mathbf{T}_{pca}^T \mathbf{S}_w \mathbf{T}_{pca}$.
>
> **Step 2.** Calculate the first d_{LDA} generalized eigenvectors
> $\mathbf{U}_S = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_{d_{LDA}}]$ of $\bar{\mathbf{S}}_b$ and $\bar{\mathbf{S}}_w$, and compute
> $\mathbf{T}_{LDA} = \mathbf{T}_{PCA} \mathbf{U}_S = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{d_{LDA}}]$.
>
> **Step 3.** Using 2D-Gaussian filter $\mathbf{M}_g$, regularize each discriminant
> vector $\mathbf{v}_i$ to $\mathbf{v}_i''$, and construct the LDA projector
> $\mathbf{T}_{PEFM} = [\mathbf{v}_1'', \mathbf{v}_2'', \cdots, \mathbf{v}_{d_{LDA}}'']$.

Fig. 5. PEFM Algorithm

With the post-processing approach, we present the implementation of the post-processed RFM method (PEFM), where the main steps are illustrated in Fig. 5.

3.2 Relation between the post-processing approach and image Euclidean distance

In (Wang et al., 2005), Wang et al presented an image Euclidean distance (IMED) method, where a 2D-Gaussian function is used to model the effect of neighbor pixels. Compared with traditional Euclidean distance, IMED can be easily embedded with some popular image feature extraction and classification methods and reported a consistent performance improvement. In this section, we will demonstrate that the IMED method actually is equivalent to the post-processing approach.

Different from traditional Euclidean distance, the computation of image Euclidean distance take into account the spatial relationships of pixels

$$d_{IME}^2(\mathbf{X}_1, \mathbf{X}_2) = \frac{1}{2\pi} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k}^{n} \sum_{l}^{n} [\mathbf{X}_1(i,j) - \mathbf{X}_2(i,j)][\mathbf{X}_1(k,l) - \mathbf{X}_2(k,l)] \exp\{-\frac{(i-k)^2 + (j-l)^2}{2}\}, \quad (26)$$

where $\mathbf{X}_1$ and $\mathbf{X}_2$ are two are two $m \times n$ images, and $\mathbf{X}_1(i, j)$ represent the gray value of the (i, j) pixel of image $\mathbf{X}_1$.

Traditional Euclidean distance can be easily rewritten using its inner product representation

$$d_E^2(\mathbf{X}_1, \mathbf{X}_2) = \langle \mathbf{X}_1, \mathbf{X}_1 \rangle + \langle \mathbf{X}_2, \mathbf{X}_2 \rangle - 2 \langle \mathbf{X}_1, \mathbf{X}_2 \rangle. \quad (27)$$

Similarly, by defining image inner product (IMIP) as

$$\langle \mathbf{X}_1, \mathbf{X}_2 \rangle_{IMIP} = \frac{1}{2\pi} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k}^{n} \sum_{l}^{n} \exp\{-\frac{(i-k)^2 + (j-l)^2}{2}\} \mathbf{X}_1(i,j) \mathbf{X}_2(k,l), \quad (28)$$

image Euclidean distance can be re-formalized as

$$d_{IME}^2(\mathbf{X}_1, \mathbf{X}_2) = \langle \mathbf{X}_1, \mathbf{X}_1 \rangle_{IMIP} + \langle \mathbf{X}_2, \mathbf{X}_2 \rangle_{IMIP} - 2 \langle \mathbf{X}_1, \mathbf{X}_2 \rangle_{IMIP}. \quad (29)$$

Different from traditional inner product, IMIP not only consider the product of two corresponding pixels, but also consider the effect of the spatial relationship between neighbor pixels, and thus is more robust against small degree of variations in translation, rotation and deformation. With the introduction of IMED and IMIP, we can conveniently embed them into many popular image feature extraction and classification approaches, such as PCA, LDA, k-nearest neighbor, and support vector machine.

According to the separability property, the definition of IMIP can be formalized to

$$\langle \mathbf{X}_1, \mathbf{X}_2 \rangle_{IMIP} = \frac{1}{2\pi} \sum_{i=1}^{m} \sum_{k=1}^{m} \exp\{-\frac{(i-k)^2}{2}\} \sum_{j=1}^{n} \sum_{l=1}^{n} \exp\{-\frac{(j-l)^2}{2}\} \mathbf{X}_1(i,j) \mathbf{X}_2(k,l) . \tag{30}$$

Let

$$\mathbf{X}_1'(i,l) = \frac{1}{\sqrt{2\pi}} \sum_{j=1}^{n} \exp\{-\frac{(j-l)^2}{2}\} \mathbf{X}_1(i,j) , \tag{31}$$

$$\mathbf{X}_1''(k,l) = \frac{1}{\sqrt{2\pi}} \sum_{i=1}^{m} \exp\{-\frac{(i-k)^2}{2}\} \mathbf{X}_1'(i,l) , \tag{32}$$

IMIP can then be represented as

$$\langle \mathbf{X}_1, \mathbf{X}_2 \rangle_{IMIP} = \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{X}_1''(i,j) \mathbf{X}_2(i,j) = \langle \mathbf{X}_1'', \mathbf{X}_2 \rangle . \tag{33}$$

From Eq. (31) and (32),

$$\mathbf{X}_1''(k,l) = \frac{1}{2\pi} \sum_{j=1}^{n} \sum_{i=1}^{m} \exp\{-\frac{(i-k)^2 + (j-l)^2}{2}\} \mathbf{X}_1(i,j) . \tag{34}$$

In the spatial domain, the definition of two-dimensional linear convolution is

$$g(i,j) = f(i,j) * h(i,j) = \sum_{k=1}^{m} \sum_{l=1}^{n} f(k,l) h(i-k, j-l) . \tag{35}$$

where $f(k,l)$ denotes the original image, $h(i,j)$ denotes the convolution kernel, and $g(i,j)$ denotes the convolution result. Clearly, by defining the convolution kernel

$$h(i,j) = \frac{1}{2\pi} \exp\{-\frac{i^2 + j^2}{2}\} , \tag{36}$$

we can show the equivalence of IMIP and the post-processing approach. IMIP actually is the post-processing approach with the standard deviation $\sigma = 1$ and without the normalization step. The post-processing approach, in fact, can be regarded as a generalization of the normalized IMIP method without the constraint on the value of the standard deviation.

3.3 Performance evaluation of PEFM
In this section, we use the FERET face database to evaluate the efficiency of the PEFM method over the original EFM method, to verify the equivalence of IMED and the post-processing approach, and to evaluate the influence of the normalization step.

In our experiment, we adopt the same experimental setup as described in Section 2.3. The nearest neighbor classifier is used to match probe images and gallery images, and the averaged recognition rate (ARR) is adopted by calculating the mean value of recognition rates across 10 runs.

For PEFM, there are three parameters, the PCA dimension d_{PCA}, the LDA dimension d_{LDA}, and the standard deviation σ, to be determined. However, it is very difficult to determine these three parameters at the same time. Previous work on the FERET subset has shown that the maximum recognition accuracy could be obtained with the LDA dimension $d_{LDA} \approx 20$. With standard deviation $\sigma \approx 1.5$, the noise in the discriminant vector would be significantly reduced. So we investigate the effect of the PCA dimension d_{PCA} with $d_{LDA} = 20$ and $\sigma = 1.5$. As the PCA dimension d_{PCA} (>100) increases, PEFM will be distinctly superior to EFM in terms of recognition accuracy. Besides, the PCA dimension has a much less effect on the recognition accuracy of PEFM, whereas that of EFM deteriorates greatly with increasing of d_{PCA}. From Fig. 6(a), we can determine the PCA dimension of PEFM, $d_{PCA} = 100$.

After determining d_{PCA}, we study the recognition accuracy over the variation of the LDA dimension d_{LDA} with $d_{PCA} = 100$ and $\sigma = 1.5$, as depicted in Fig. 6(b). The maximum average recognition rate of PEFM is obtained with the LDA dimension $d_{LDA} = 24$. Then we explore the recognition rate vs. the variation of σ with $d_{PCA} = 100$ and $d_{LDA} = 24$, as shown in Fig. 6(c). The maximum average recognition rate, 87.34%, is obtained using PEFM with $d_{PCA} = 100$, $d_{LDA} = 24$ and $\sigma = 1.5$, which is higher than 84.54%, that of EFM.
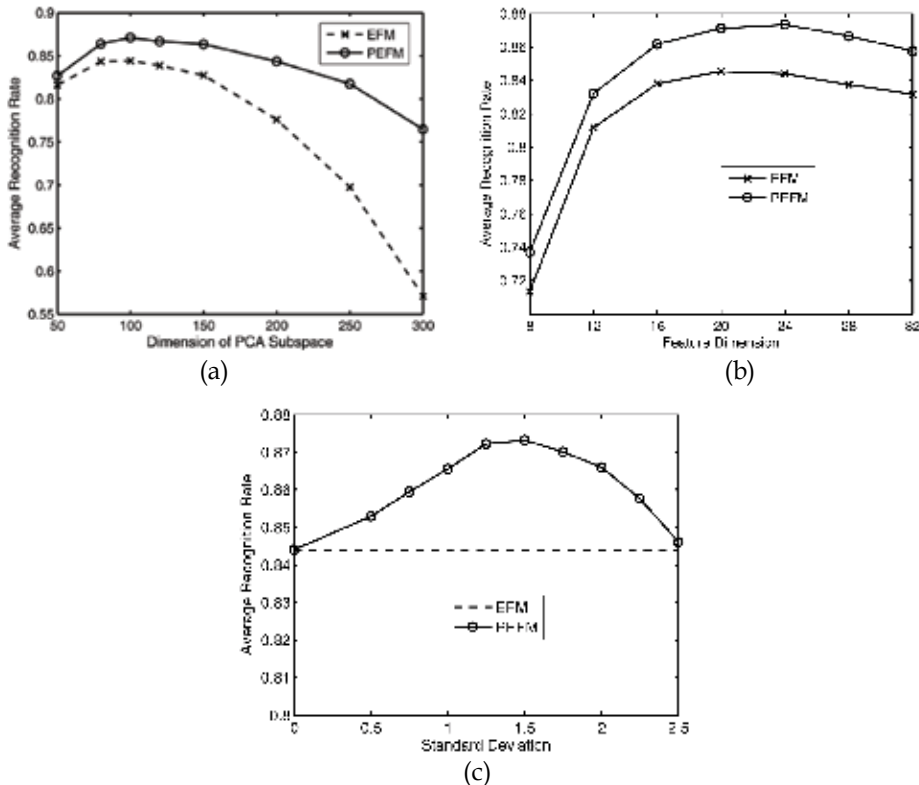

Fig. 6. Illustration of the recognition accuracy over the variation of the PEFM parameters on the FERET subset. (a) Recognition accuracy vs. the PCA dimension. (b) Recognition accuracy vs. the LDA dimension. (c) Recognition accuracy vs. the standard variation

Next we compare the recognition performance of IMED-embedded EFM and PEFM without normalization. Fig. 7 shows the recognition rates of these two methods over different feature

dimensions. From Fig. 7, we can see that, the performance difference between these two approaches is very small, and PEFM without normalization only achieve a little higher recognition rate than IMED-embedded EFM. The small performance difference, however, may be explained by that, for PEFM, IMED is only embedded in the testing stage. This indicates that, when IMED is embedded in enhanced Fisher model (EFM), it would be better to embed IMED only in the testing stage rather than to embed IMED in both the training and the testing stage.
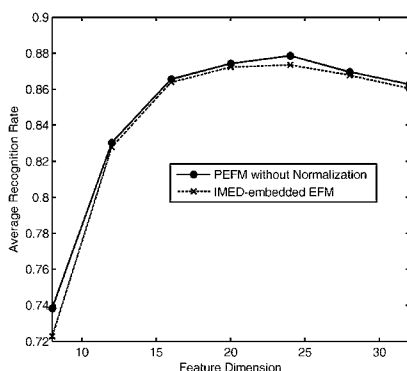

Fig. 7. Recognition rates of IMED-embedded EFM and PEFM without normalization
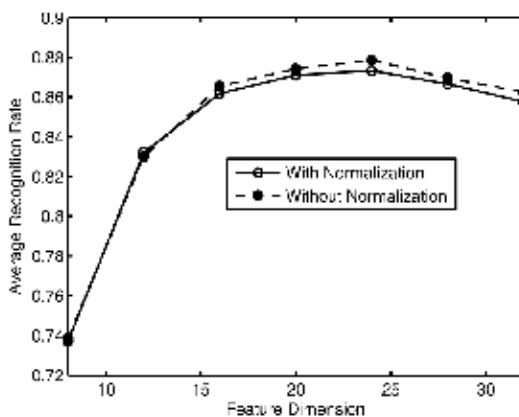

Fig. 8. Illustration of the recognition rates of PEFM with and without the normalization step over different feature dimensions.

Finally, we investigate the influence of the normalization step. Fig. 8 shows the recognition rates of PEFM with and without the normalization step over different LDA dimensions. From Fig. 8, we can see that, the normalization step in PEFM actually has a little effect on the recognition performance when applied to face recognition.

4. Robust recognition by iterated reweighted fitting of eigenfaces

A real face recognition system should capture, detect and recognize facial image automatically, making it inevitable that facial images will sometimes be noisy, partially occluded, or inaccurately located. The capture and communication of facial image itself may

introduce noise; some accessories will cause the occlusion of a facial image, for example a scarf may occlude a facial image; and facial images usually should be normalized by locations of landmarks but these locations may be inaccurate and inconsistent. Because all these three factors are inevitable, the development of face recognition system should always address the robust recognition of noisy, partially occluded, or inaccurate located image.

4.1 Iterated reweighted fitting of eigenfaces

Zhao et al. (2003) showed that, in face recognition, the appearance-based methods (e.g., PCA and LDA) are robust in the presence of low levels of small noise or occlusion. However, if the degree of occlusion further increases, the recognition performance would deteriorate severely (Martinez, 2002). Analogous to partial occlusion, the further increase of noise would also cause an immediate decrease in recognition performance.

To address the partial occlusion problem, Martinez proposed a local probabilistic approach where each face image is divided into six local areas, and each local area is then projected into its eigenspace in the recognition stage (Martinez, 2002). Local probabilistic approach, however, cannot be used to weaken the unfavorable effect of noise because noise is always globally distributed. Besides, local probabilistic approach, which divides an image into a number of parts, also neglects the global correlation of the face image.

Robust estimation (McLachlan & Krishnan, 1997; Isao & Eguchi, 2004) and robust appearance-based methods can be used to solve the noise and partial occlusion problems. For example, iterated reweighted fitting of Eigenfaces (IRF-Eigenfaces), a robust estimation of the coefficients of Eigenfaces, can address this by first defining an objective function $J(\mathbf{y})$ and then using the Expectation Maximization (EM) algorithm to compute the feature vector $\mathbf{y}$ by minimizing $J(\mathbf{y})$. The following presents the main steps in IRF-Eigenfaces:

1. Define the objective function. Given a set of Eigenfaces $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d]$, IRF-Eigenfaces calculate the feature vector $\mathbf{y}$ of image $\mathbf{x}$ by minimizing the objective function

$$J(\mathbf{y}) = \sum_{i=1}^{mn} \Psi\left((x_i - \mathbf{W}_i \mathbf{y})^2 \right), \tag{37}$$

where the function $\Psi(z)$ could be defined as (Isao & Eguchi, 2004)

$$\Psi(z) = \log \frac{1}{1 + \exp(-\beta(z - \eta))}, \tag{38}$$

where the inverse temperature β and saturation value η are two tuning parameters.

2. Calculate feature vector $\mathbf{y}$ by iteratively performing the next two steps until the value of $\mathbf{y}^{(t)}$ converges or t arrives at the pre-determined threshold $t_{\max}$.:

   E-Step. Given $\mathbf{y}^{(t)}$, update the weighted vector $\boldsymbol{\omega}^{(t)} = [\omega_1^{(t)}, \omega_2^{(t)}, \cdots, \omega_{mn}^{(t)}]$ by

$$\omega_i^{(t)} = \varphi\left(z_i^{(t)} \right) = \frac{\exp\left\{ -\beta(z_i^{(t)} - \eta) \right\}}{1 + \exp\left\{ -\beta(z_i^{(t)} - \eta) \right\}}$$

$$z_i^{(t)} = (x_i - \mathbf{W}_i \mathbf{y}^{(t)})^2$$

M-Step. Given the weighted vector $\boldsymbol{\omega}^{(t)}$, update $\mathbf{y}^{(t+1)}$ by

$$\mathbf{y}^{(t+1)} = \left(\sum_{i=1}^{mn} \omega_i^{(t)} \mathbf{W}_i^T \mathbf{W}_i \right)^{-1} \sum_i^{mn} \omega_i^{(t)} \mathbf{W}_i^T x_i$$

If we define the function $\Psi(z)$ a strictly concave function in z, $\Psi''(z) < 0$, we can theoretically guarantee that $\mathbf{y}^{(t)}$ will converge to a local optimal feature vector $\mathbf{y}$.

4.2 Evaluation on IRF-eigenfaces

We use the AR face database to evaluate the performance of the IRF-Eigenfaces method against noise and partial occlusion, and compare the recognition rate of IRF-Eigenfaces with those of Eigenfaces and the local probabilistic approach. The AR face database contains over 4,000 color frontal facial images corresponding to 126 people (70 men and 56 women) (Martinez & Benavente, 1998).There are 26 different images of each person and these were captured in two sessions separated by two weeks. In our experiments, we only use a subset of 720 images of 120 persons. There are six images of each person. Three images were captured in the first session (one neutral, one with sunglasses, and one with a scarf) and the remaining three images were captured in the second session (one neutral, one with sunglasses, and one with a scarf), as shown in Fig. 9. In our experiments, all the images were cropped according to the location of their eyes and mouths, and the 120 neutral images in the first session were used for training Eigenfaces. We set the number of principal components d_{PCA}=100 and use the whitened cosine distance measure (Phillips et al., 2005).

(a)   (b)   (c)

(d)   (e)   (f)

Fig. 9. Six images of one person in the AR database. The images (a) through (c) were captured during one session and the images (d) through (f) at a different session.

IRF-Eigenfaces is more robust when it comes to reconstruct noisy and occluded facial images. The quality of reconstructed facial images using IRF-Eigenfaces is also consistently better than those using Eigenfaces, as shown in Fig. 10. IRF-Eigenfaces also has a robust reconstruction performance for the partially occluded facial images, as shown in Fig. 11.
Using all the neutral images in the second session as a test set, we investigate the recognition performance of IRF-Eigenfaces against different degree of noise. Fig. 12 shows an original facial image and the same image after the addition of various amounts of salt and pepper

noise. The largest degree of noise in our test is 50%, which is a seriously contaminated example. We then present the recognition rates of Eigenfaces and IRF-Eigenfaces against different degrees of noise contamination, as shown in Fig. 13. The addition of 50% salt and pepper noise caused the recognition rate of Eigenfaces to fall from 81.67% to 36.67%, but the recognition rate of IRF-Eigenfaces remained unchanged (95.83%). Because the recognition rate is robust against variation of noise, we can validate the robustness of IRF-Eigenfaces against noise.
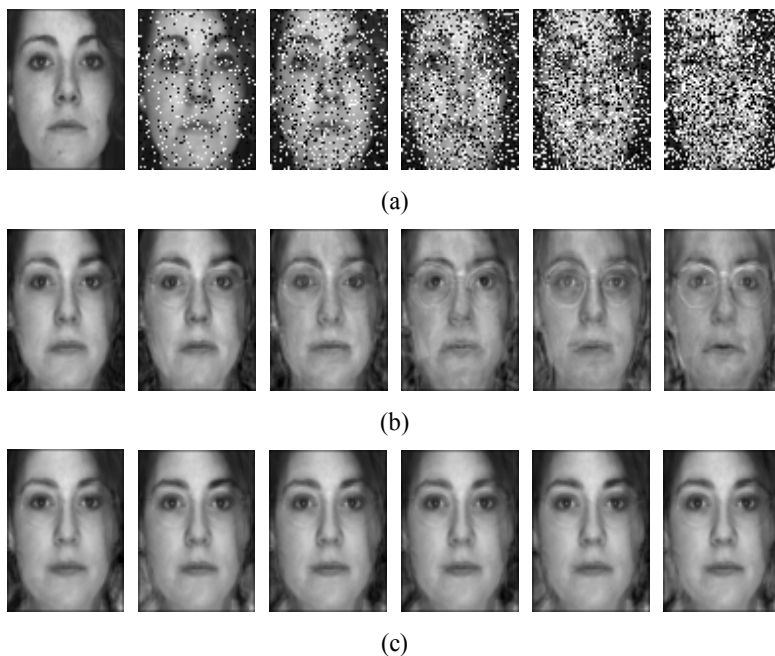

(a)

(b)

(c)

Fig. 10. Reconstruction of images with different degree of salt and pepper noise: (a) original image; (b) reconstructed images by Eigenfaces; (c) reconstructed images by IRF-Eigenfaces.

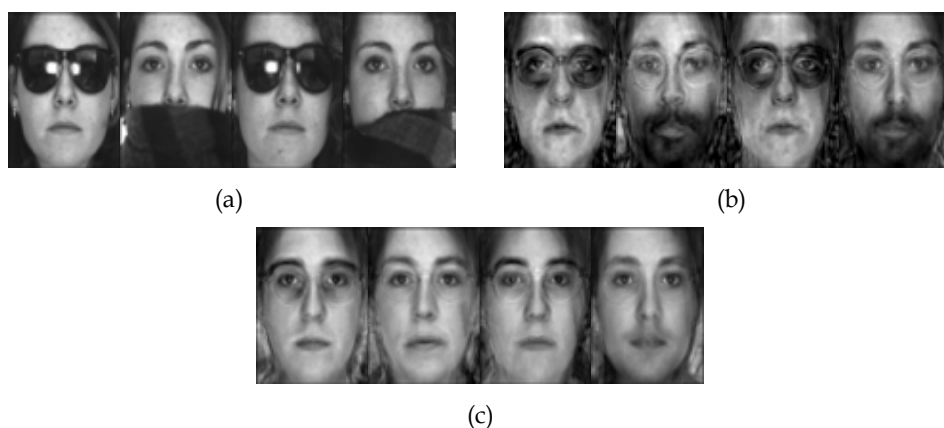(a)   (b)

(c)

Fig. 11. Reconstructed partially occluded images: (a) original images; (b) reconstructed images by Eigenfaces; (c) reconstructed images by IRF-Eigenfaces.
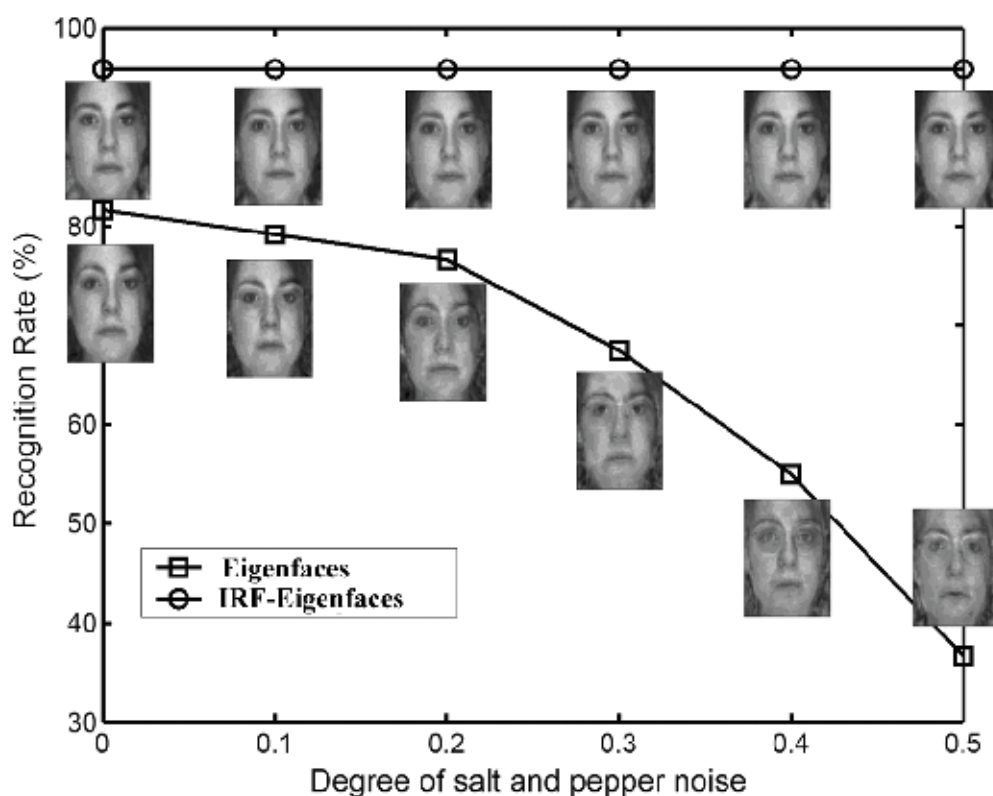
Fig. 13. The effect of salt and pepper noise on the recognition performance

Using the images with sunglasses or scarves, we tested the influence of partial occlusion on the recognition performance of IRF-Eigenfaces. Table 5 lists the recognition rates of Eigenfaces, IRF-Eigenfaces and the local probabilistic approach in recognizing faces partially occluded with either sunglasses or a scarf. The recognition rate of IRF-Eigenfaces in both the first and second sessions is much higher than that of the other two methods, Eigenfaces and the local probabilistic approach.

Methods	Session 1		Session 2	
	Sunglasses	Scarf	Sunglasses	Scarf
Eigenfaces (%)	40	35	26.67	25
LocProb (%)	80	82	54	48
IRF-Eigenfaces (%)	87.50	91.67	82.50	84.17

Table 5. Recognition performance of three face recognition methods on the AR database

Another interesting point to be noted from Table 5 is that IRF-Eigenfaces is also more robust against the variation of ageing time. Eigenfaces' recognition rate in the second session (25.83%) is much lower than in the first session (37.5%). The local probabilistic approach's recognition rate in the second session (51%) is much lower than in the first session (81%). But IRF-Eigenfaces' recognition rate in the second session (83.34%) is only slightly lower than in the first session (89.58%). There are two reasons for the robustness of IRF-Eigenfaces

against ageing time. First, IRF-Eigenfaces uses the whitened cosine distance, which can reduce the adverse effect of global illumination change of the facial image. Second, IRF-Eigenfaces, which is robust to partial occlusion, is also robust to some facial change, such as the presence of a beard. Compare Fig. 14(a), showing a neutral face captured in the first session, with Fig. 14(b), showing a face with sunglasses captured in the second session. The image in Fig. 14(b), captured in the second session, has a heavier beard. Fig. 14(c) shows the image in Fig. 14(b) reconstructed using IRF-Eigenfaces. IRF-Eigenfaces can detect parts of the beard as a partial occlusion and thus its reconstructed image is more consistent with Fig. 14(a).
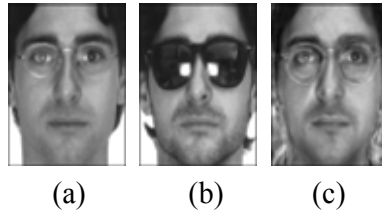

(a)   (b)   (c)

Fig. 14. The reconstruction performance of IRF-Eigenfaces for partially occluded face in the second session. (a) neutral face in the first session; (b) face with sunglasses in the second session; (c) the reconstructed image of (b) using IRF-Eigenfaces.

7. Summary

In this chapter, we introduce several recently developed subspace-based face recognition methods in addressing three problems, singularity, regularization, and robustness. To address the singularity problem, we present a fast feature extraction technique, Bi-Directional PCA plus LDA (BDPCA+LDA), which performs LDA in the BDPCA subspace. Compared with the PCA+LDA framework, BDPCA+LDA has a number of significant advantages. First, BDPCA+LDA needs less computational requirement in both the training and the testing phases. Second, BDPCA+LDA needs less memory requirement because its projector is much smaller than that of PCA+LDA. Third, BDPCA+LDA has a higher recognition accuracy over PCA+LDA.

To alleviate the over-fitting to the training set, this chapter suggests a post-processing approach on discriminant vectors, and demonstrates the internal relationship between the post-processing approach and IMED. Experimental results indicate that, the post-processing approach is effective in improving the recognition rate of the LDA-based approaches. When IMED is embedded in enhanced Fisher model, it would be better to embed IMED only in the testing stage.

To improve the robustness of subspace method over noise and partial occlusion, this chapter further presents an iteratively reweighted fitting of the Eigenfaces method (IRF-Eigenfaces). Despite the success of IRF-Eigenfaces in recognizing noisy and partially occluded facial images, it is still very necessary to further study this issue by investigating the robustness against inaccurate fiducial point location, illumination, and ageing in one uniform framework.

8. Acknowledgement

9. References

Baeka, J. & Kimb, M. (2004) Face recognition using partial least squares components. *Pattern Recognition*, Vol. 37, 1303-1306

Bartlett, M.S.; Movellan, J.R. & Sejnowski, T.J. (2002) Face Recognition by Independent Component Analysis. *IEEE Trans. Neural Network*, Vol. 13, No. 6, 1450-1464

Belhumeur, P. N.; Hespanha, J. P. & Kriegman, D. J. (1997) Eigenfaces vs Fisherfaces: recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.20, No.7, 711-720

Cevikalp, H.; Neamtu, M.; Wilkes, M. & Barkana, A. (2005) Discriminative common vectors for face recognition. *IEEE Trans Pattern Analysis and Machine Intelligence*, Vol. 27, No. 1, 4-13

Fukunaga, K. (1990) *Introduction to Statistical Pattern Recognition, Second Edition*, Academic Press, New York

He, X.; Yan, S.; Hu, Y.; Niyogi, P. & Zhang, H.-J. (2005) Face Recognition Using Laplacianfaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 27, No. 3, 328-340

Isao, I. & Eguchi, S. (2004) Robust principal component analysis with adaptive selection for tuning parameters. *Journal of Machine Learning Research*, Vol. 5, 453-471

Jolliffe, I.T. (2002) *Principal Component Analysis, Second Edition*, Springer, New York

Kanade, T. (1973) *Computer recognition of human faces*. Birkhauser, Basel, Switzerland, and Stuttgart, Germany

Lathauwer, L.; Moor, B. & Vndewalle, J. (2000) A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Application*, Vol. 21, No. 4, 1233-1278

Liu, C. (2004) Gabor-based kernel PCA with fractional power polynomial models for face recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 26, No. 5, 572-581

Liu, C. & Wechsler, H. (1998) Enhanced Fisher linear discriminant models for face recognition. *Proc. 14th Int'l Conf. Pattern Recognition*, Vol. 2, pp. 1368-1372

Liu, C. & Wechsler, H. (2002) Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Trans. Image Processing*, Vol. 10, No. 4, 467-476

Liu, K.; Cheng, Y.-Q. & Yang, J.-Y. (1993) Algebraic feature extraction for image recognition based on an optimal discriminant criterion. *Pattern Recognition*, Vol. 26, No. 6, 903-911

Lu, J.; Plataniotis, K.N. & Venetsanopoulos, A.N. (2003) Face recognition using kernel direct discriminant analysis algorithms. *IEEE Trans. Neural Networks*, Vol. 14, No. 1, 117-126

Martinez, A.M. (2002) Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 24, No. 6, 748-763

Martinez, A.M. & Benavente, R. (1998) *The AR Face Database*. CVC Technical Report #24

McLachlan, G.J. & Krishnan, T. (1997) *The EM algorithm and extensions*, John Wiley & Sons, New York

Philips, P.J.; (1998) The Facial Recognition Technology (FERET) Database, http://www.itl. nist.gov/iad /humanid/feret.

Phillips, P.J.; Flynn, P.J.; Scruggs, T.; Bowyer, K.W.; Chang, J.; Hoffman, K.; Marques, J.; Min, J. & Worek, W. (2005) Overview of the face recognition grand challenge. *Proc IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 947-954

Philips, P.J.; Moon, H.; Rizvi, S.A. & Rauss, P.J. (2000) The FERET evaluation methodology for face-recognition algorithms,"*IEEE Trans Pattern Analysis and Machine Intelligence*, Vol. 22, No. 10, 1090-1104

Roweis, S.T. & Saul, L.K. (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science*, Vol. 290, 2323-2326

Swets, D.L. & Weng, J. (1996) Using discriminant eigenfeatures for image retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 18, No. 8, 831-836

Tao, D.; Li, X.; Hu, W.; Maybank, S.J. & Wu, X. (2005) Supervised tensor learning. *IEEE Int'l Conf. on Data Mining*, pp. 450-457

Tenenbaum, J.B.; de Silva, V.; & Langford, J.C. (2000) A global geometric framework for nonlinear dimensionality reduction. *Science*, Vol. 290, 2319-2323

Timo, A.; Abdenour, H. & Matti, P. (2004) Face Recognition with Local Binary Patterns. *Proceeding of European Conference on Computer Vision*, pp. 469-481

Torkkola, K. (2001) Linear Discriminant Analysis in Document Classification. *Proc. IEEE ICDM Workshop Text Mining*

Turk, M. & Pentland, A. (1991) Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, Vol.3, No.1, 71-86

Wang, L.; Zhang, Y. & Feng, J. (2005) On the Euclidean distance of images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 27, No. 8, 1334-1339

Wang, K.; Zuo, W. & Zhang, D. (2005) Post-processing on LDA's discriminant vectors for facial feature extraction. *Proc. 5th Int'l Conf. Audio- and Video-based Biometric Person Authentication*, pp. 346-354

Wang, X. & Tang, X. (2004) A unified framework for subspace face recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 26, No. 9, 1222-1228

Wiskott, L.; Fellous, J. M.; Kruger, N. & Malsburg, C. (1997) Face Recognition by Elastic Bunch Graph Matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.19, No. 7, 775-779

Yan, S.; Xu, D.; Yang, Q.; Zhang, L.; Tang, X. & Zhang, H.-J. (2007) Multilinear discriminant analysis for face recognition. *IEEE Trans. Image Processing*, Vol. 16, No. 1, 212-220

Yang, J. & Yang, J.Y. (2003) Why can LDA be performed in PCA transformed space. *Pattern Recognition*, Vol. 36, No. 2, 563-566

Yang, J.; Zhang, D.; Frangi, A.F. & Yang, J.-Y. (2004) Two-Dimensional PCA: a New Approach to Face Representation and Recognition. *IEEE Trans Pattern Analysis and Machine Intelligence*, Vol. 26, No. 1, 131-137

Yang, J.; Zhang, D.; Xu, Y. & Yang, J.Y.(2005a) Two-dimensional discriminant transform for face recognition. *Pattern Recognition*, Vol. 38, No. 7, 1125-1129

Yang, J.; Zhang, D.; Yang, J.-Y.; Zhong, J. & Frangi, A.F. (2005b) KPCA plus LDA: a Complete Kernel Fisher Discriminant Framework for Feature Extraction and Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 27, No. 2, 230-244

Yang, M.H. (2002) Kernel Eigenfaces vs. kernel Fisherfaces: face recognition using kernel methods. *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (RGR'02)*, pp. 215-220

Ye, J. (2004) Generalized Low Rank Approximations of Matrices. *The Twenty-First International Conference on Machine Learning*

Ye, J. (2005) Generalized low rank approximations of matrices. *Machine Learning*, Vol. 61, No. 1-3, 167-191

Yu, H. & Yang, J.(2001) A direct LDA algorithm for high-dimensional data—with application to face recognition. *Pattern Recognition*, Vol. 34, No. 10, 2067-2070

Zafeiriou, S.; Tefas, A.; Buciu, I. & Pitas, I. (2006) Exploiting Discriminant Information in Nonnegative Matrix Factorization With Application to Frontal Face Verification. *IEEE Trans. Neural Networks*, Vol. 17, No. 3, 683-695

Zhao, W.; Chellappa, R.; Phillips, P.J. & Rosenfeld, A.(2003) Face recognition: a literature survey. *ACM Computing Surveys*, Vol. 35, No. 4, 399-458

Zuo, W.; Wang, K. & Zhang, D.(2005a) Bi-directional PCA with assembled matrix distance metric. *International Conference on Image Processing*, pp. 958-961

Zuo, W.; Wang, K.; Zhang, D. & Yang, J. (2005b) Regularization of LDA for face recognition: a post-processing approach. *Proc. 2th Int'l Workshop on Analysis and Modeling of Face and Gesture*, pp. 377-391

Zuo, W.; Zhang, D.; Yang, J. & Wang, K. (2006) BDPCA plus LDA: a novel fast feature extraction technique for face recognition. *IEEE Trans. Systems, Man, and Cybernetics, Part B*, Vol. 36, No. 4, 946-953

A Multi-Stage Classifier for Face Recognition Undertaken by Coarse-to-fine Strategy

Jiann-Der Lee[1] and Chen-Hui Kuo[2]

[1]*Department of Electrical Engineering, Chang Gung University, Tao-Yuan, 333*
[2]*Department of Electrical Engineering, Chung Chou Institute of Technology,*
Chang-Hua, 510,
Taiwan , R.O.C

1. Introduction

Face recognition has been a very active research area for past two decades due to its widely applications such as identity authentication, airport security and access control, surveillance, and video retrieval systems, etc. Numerous approaches have been proposed for face recognition and considerable successes have been reported [1]. A successful face recognition system should be robust under a variety of conditions, such as varying illuminations, pose, expression, and backgrounds. Although many researches [1-3] already found solutions to alleviate those problems, it is still a great challenge to accurately recognize faces that are non-frontal facial images, and disguises such as facial hair, glasses or cosmetics.

In the past, two categories of feature extraction for face recognition have been developed, including of geometric features and statistical features derived from face images [1-4]. In geometric features method, the facial features are retrieved from either the shapes of eyes, nose, mouth, and chin, or the facial geometrical relationships such as areas, distances, and angles [4]. This kind of approaches has proven to be difficult for practical application because it requires a fine segmentation of facial features. In statistical features method, the facial features are usually extracted from important facial features based on the high-dimensional intensity values of face images. For example, the principal component analysis (PCA) is the well-known statistical method [5]. This approach is simple, but does not reflect the details of facial local features.

Generally speaking, researches on face recognition system can be grouped into two categories of classifier system, one is single-classifier system and the other is multi-classifier system. The single-classifier systems, including neural network (NN) [6], Eigenface [5], Fisher linear discriminant (FLD) [7], SVM [8], HMM [9], and AdaBoost [10], are developed well in theory and experiment. On the other hand, the multi-classifier systems such as local and global face information fusion [11-13], neural networks committee (NNC) [14], multi-classifier system (MCS) [15], are proposed in parallel process of different features or classifiers.

As one knows, neural networks (NN) are a nonlinear classifier and based on the parallel architecture of human brains. Its output can be binary or multiple classes. In NN, the margin between two classes of sample point nearby the decision function is often not a maximum.

Because of its performance surface depending on the mean square error, and if the train error reaches to zero too quickly, its performance surface will be minimum and the weight values will have to be stopped to get adjusted [16]. For each subject, the numbers of face image sample are always small on public face databases, so the early training break in NN frequently occurs. The AdaBoost also has the same drawback as the NN does. That is, the situation of the early training break occurs in AdaBoost if the error of weak learner quickly reaches to zero. Besides, the application of HMM is a content-dependent classification, especially on speech recognition. It is based on an assumption that the class variations are closely related. The research of HMM on face recognition includes 1D-HMM [17], 2D-HMM [18], embedded HMM [19], and pseudo 2D-HMM [20]. The disadvantage of using HMM on face recognition is on the part of huge computation cost under the condition that the numbers of dimension and state are high.

The SVM were originally designed for binary classification and it is based on the structural risk minimization (SRM) principle. Although several methods to effectively extend the SVM for multi-class classification have been reported on technical literatures [21,22], it is still an on-going research issue. The category methods of SVM for multi-class classification are one-against-all (OAA) [21,30], one-against-one (OAO) [21], directed acyclic graph support vector machine (DAGSVM) [23], and binary tree SVM [8]. In DAGSVM and binary tree SVM, their training phases are the same as those of the OAO method by solving $N(N\text{-}1)/2$ binary SVMs, where N is the numbers of class. However, in the testing phase of DAGSVM, it uses a rooted binary directed acyclic graph which has $N(N\text{-}1)/2$ internal nodes and N leaves. Each node is a binary SVM of ith and jth classes. On the other hand, the testing phase of binary tree SVM constructs a bottom-up binary tree for classification. The advantage of using a DAGSVM and binary tree SVM is less testing time than that of OAO (maximum vote) method. If one employs the same feature vector for SVM, NN, and AdaBoost, he will find the performance of SVM is better than that of NN and AdaBoost because the SVM will result in the maximum separating margin to the hyperplane of the two classes. And if the feature vector includes noisy data, and the noisy data possesses at least one of the following properties: (a) overlapping class probability distributions, (b) outliers and (c) mislabeled patterns [24], the hyperplane of SVM will turn out to be hard margin and overfitting. Additionally, the SVM allows noise or imperfect separation, provided that a non-negative slack variable is added to the objective function as a penalizing term.

The Eigenface is a successful example using template matching for face recognition [5]. The method of Eigenface uses principle component analysis (PCA) or Karhunen-Loeve transform that constructs a number of Eigenfaces derived from a set of training face images. Every prototype face image in the database is represented as a feature point, i.e. a vector of weights, in the space and is the query face image. The limitations of Eigenface are their computation cost and memory requirement burden if the face recognition system is to scale up.

As for multi-classifier systems, Zhou et al. [13] presented a combined feature Fisher classifier (CF²C) approach, whose combined facial features are derived from facial global and local information extracted by DCT, for face recognition. And it performs better than the traditional methods such as Eigenfaces and Fisherfaces. One problem of this method is that it is hard to accurately detect the landmark of images, and another is that the localization errors might sustain to the classification step to disturb the final result. Rajagopalan et al. [12] proposed a face recognition method that fuses information acquired

from global and local features of the face for improving performance. Their method is very similar to that proposed by Zhou et al. [13], it brings about the same problem accordingly as mentioned above. Kwak et al. [11] proposed two approaches to fuzzy information fusion for face recognition involving aggregation of local and global face information and a wavelet decomposition approach. Zhao et al. [14] studied a face recognition method based on multi-features using a neural networks committee (NNC) machine. Either of their methods is based on parallel features and classifiers, and is with a combination strategy at the end. Although their experimental results show a more accurate classification rate than that of single feature and classifier, its architecture is totally different from our proposed cascade stages, which are proceeded with a coarse-to-fine strategy. Our novel coarse-to-fine strategy ends up with a much more successful performance in facial recognition.

In our system, the extracted feature for SVM is discrete cosine transform (DCT) coefficients that are common used for image pre-processing. In the past, Chen et al. [6] extracted DCT feature from the entire face image for face recognition, because they believed that if the DCT is obtained from individual sub-images, certain relationship information between sub-images are not existed any more. In [25], Jing and Zhang selected the useful DCT frequency bands and obtained a 1D training sample set, then they proposed an improved Fisherface method to extract the image discrimination features, at last they applied the nearest neighbor classifier to the feature classification.

To combine the image feature of frequency, intensity, and space information, we propose a novel face recognition approach, which combines SVM, Eigenface, and RANSAC [26] methods with the multi-stage classifier system. The whole decision process is developed through consecutive stages, i. e., "one-against-all (OAA) of SVM", "one-against-one (OAO) of SVM", "Eigenface", and "RANSAC", respectively. The stage 1 "OAA of SVM" and stage 2 "OAO of SVM" uses the DCT features extracted from the entire face image. The stage 3 "Eigenface", face images are projected onto a feature space (face space). The face space is defined by the "Eigenfaces", which are the eigenvectors of the set of faces and based on intensity information of the face image. "RANSAC" is applied in the last stage, in which the epipolar geometry method with space information of the testing image is matched with the two training images, and then the image with the greatest match numbers of and the shortest distance to corresponding feature points is selected. The face databases used for performance evaluation are retrieved from Olivetti Research Laboratory (ORL) [44], Yale database [45], and IIS (Institute of Information Science, Academia Sinica) face databases [46]. For each database, we use four different evaluation methods, which are OAA-SVM, OAO-SVM, Eigenface, and our proposed multi-stage classifier. In this way, the experimental results can be compared with other face recognition approaches fairly.

The remainder of this paper is organized as following: In section 2, the feature extraction methods, i.e., the DCT and PCA are briefly described. In section 3, the proposed coarse-to-fine stages, OAA, OAO, and multi-stage classifiers are presented in detail. Experimental results using this method and the comparison to other approaches with several famous face databases are given in section 4. Conclusions are included in section 5.

2. Feature extraction

The goal of feature extraction is to transform a given set of sample data to a new set of features. If the transform method is suitably chosen, the features after transformation can

exhibit high "information packing" properties compared with the original input data. This means that most of the classification-related information is "squeezed" in a relatively small number of features, leading to a reduction of the necessary feature space dimension. 2D image that usually transforms space information to frequency information can be accomplished by one of the following methods, such as Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and Fast Fourier Transform (FFT). These methods are applicable to fixed original images and optimal information packing properties. In addition, the computation requirement of these methods is lower than that of Karhunen-Loeve (KL, also known as PCA) Transform [6]. PCA is a useful statistical technique that can be applied to fields such as face recognition [5] and image compression, and it is a common technique for finding patterns in data of high dimension. PCA also is for feature extraction under the unsupervised learning setting. This section will give a brief introduction of the two methods, DCT and Eigenface, for feature extraction.

2.1 Discrete Cosine Transform (DCT)

There are three procedures for feature extraction of frequency transformation: (1). data sampling, (2) data transform, (3) feature vector extraction. For data sampling, two kinds of approaches namely block-based and non block-based are common used; overlapped block-based with top-bottom scan [27] or overlapped block-based with raster scan [28,29], and non block-based Full image [6]. That is, in this approach, we employ the DCT to the entire face image for data sampling, as shown in Fig. 1. Because some related information between sub-images cannot be obtained if the DCT is applied to the sub-image independently. With entire-face data sampling by the DCT, all frequency information then can be accomplished. The DCT coefficients with large magnitude are mainly located in the upper-left corner of the DCT matrix as shown in Fig. 1(c). In our system, we scan the DCT coefficient matrix in a zig-zag manner starting from the upper-left corner and subsequently convert it to a one-dimensional vector as shown in Fig. 1(d).

The DCT is a technique used for converting space signals into frequency components. The one-dimensional DCT is useful in processing one-dimensional (1D) signals such as speech waveforms. For analysis of two-dimensional (2D) signals such as images, a 2D version of the DCT (2D-DCT) is required. In short, for an $N \times M$ matrix f, the 2D-DCT is computed in a simple way: The 1D-DCT is applied to each row of f and then to each column of the result. If an image $f(x,y)$ of size $N \times M$, whose discrete cosine transform is $C(u,v)$, it can be expressed as Eq. (1).

$$C(u,v) = \frac{2}{\sqrt{NM}} \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2x+1)v\pi}{2M}\right) \ , \qquad (1)$$

$$u = 0,\dots,N, \quad v = 0,\dots,M$$

where
$$\alpha(u) = \begin{cases} 2^{-\frac{1}{2}} & for \quad u = 0 \\ 1 & otherwise \end{cases}$$

Since 2D-DCT can be computed by applying 1D transforms separately to the rows and columns, it can be said that 2D-DCT is separable in two dimensions. In the 1D case, each element $C(u,v)$ of the transform is the inner product of the input and a basis function, but in the 2D case, the basis functions are $N \times M$ matrices. Each 2D basis matrix is the outer product of two of the one-dimensional basis vectors.

2.2 Eigenface method

PCA is a well-known technique commonly exploited in multivariate linear data analysis. The main underlying concept is to reduce the dimensionality of a data set while retaining as much variation as possible in a data set. If a face image $f(x,y)$ is a two-dimensional $N \times N$ array of intensity values, the corresponding image i_k is viewed as a vector with N^2 coordinates that result from a concatenation of successive rows of the image. Moreover, if the training sets have M face images and every image is of the same size, the training set of M faces can be denoted by $I = \{i_1, i_2, \ldots, i_M\}$. Its average face $\bar{i}$ is defined by:

$$\bar{i} = \frac{1}{M} \sum_{j=1}^{M} i_j$$

Each face image differs from the average by the vector $\phi_j = i_j - \bar{i}$ for $j=1,\ldots, M$. This set of huge vectors is then subject to principal component analysis, which seeks a set of M orthonormal vectors u_k and their associated eigenvalues λ_k, which best describes the distribution of the data. The vectors u_k and scalars λ_k are the eigenvectors and eigenvalues, respectively, of the covariance matrix.

$$C = \frac{1}{M} \sum_{j=1}^{M} \phi_j \phi_j^T$$
$$= AA^T$$

(2)

where $A=[\phi_1 \; \phi_2 \; \ldots \phi_M]$. The matrix C of Eq. (2), however, is $N^2 \times N^2$, and determining the N^2 eigenvectors and eigenvalues is an intractable task for typical image sizes. This problem can be alleviated by referring to some basic finding known in linear algebra. Fortunately we can determine the eigenvectors by first solving a much smaller $M \times M$ matrix problem, and taking linear combinations of resulting vectors [5]. The cost of computation is greatly decreased from the order of the number of pixels in the images (N^2) to the order of the number of images in the training set (M). In practice, the numbers of face images M are smaller than numbers of pixels N^2 ($N^2 >> M$), and the calculations of the covariance matrix C become quite manageable.

A testing face image (i_x) is transformed into its eigenface components (projected into "face space") by a simple operation, $w_k = u_k^T (i_x - \bar{i})$ for $k=1,\ldots, M$. The weights form a vector $\psi^T = [w_1 \; w_2 \; \ldots \; w_M]$ that describes the contribution of each eigenface in representing the input face image, treating the eigenfaces as a basis set for face images. The vector is used to find which number of pre-defined face class best describes the face. The simplest method for determining which face class provides the best description of an input face image is to find the face class k that minimizes the Euclidian distance $\varepsilon_k = \left\| (\psi - \psi_k) \right\|$, where ψ_k is a vector describing the kth face class.

3. The proposed method

For face recognition, based on coarse-to-fine strategy, we design a multi-stage recognition system which combines SVM, Eigenface, and RANSAC methods to increase the recognition accuracy. The detail of this system is demonstrated as following.

3.1 Support vector machine for binary classifier

Support vector machine (SVM) is a method based on statistical learning theory, which is applicable as pattern recognition, developed by V Vapnik [22]. The main purpose of the SVM is to separate two classes and maximize the margin between Hyper-plane and the nearly data point, as shown in Fig. 2.

In the case of linearly separating two classes, the two classes of hyper-planes, $(w·x)+b=0$, where $w \in R^d$ and $b \in R$, is considered, its corresponding to the decision function Eq. (3) is:

$$f(x) = \text{sgn}((w \cdot x) + b) = \pm 1 \tag{3}$$

The decision function $f(x)$ is described by weight vector w, threshold b and input patterns x. The solution to the optimization problem of SVM is given by the saddle point of Lagrange functional as Eq. (4):

$$\begin{cases} \min_{w,b} L_p = \dfrac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i y_i \cdot ((x_i \cdot w) + b) + \sum_{i=1}^{n} \alpha_i \\ subject \ \ to \quad \alpha_i \geq 0 \quad i = 1, 2, \ldots, n \\ L_p : primal \quad problem \end{cases} \tag{4}$$

where α_i : Lagrange multipliers.

By using Lagrange multiplier techniques, the minimization of Eq. (4) leads to the following dual optimization problem as Eq. (5).

$$\begin{cases} \max_{\alpha_i} \quad L_D = \sum_{i=1}^{n} \alpha_i - \dfrac{1}{2} \sum_{i=1; j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j \\ subject \ \ to \quad \alpha_i \geq 0 \quad i = 1, 2, \ldots, n \\ \qquad\qquad \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{5}$$

where L_D: dual problem

When the training data are nonlinear, we usually change the feature space dimension using the transfer function $\Phi( )$. Thus the decision functions of the more general form [31] are obtained as Eq. (6).

$$\begin{aligned} f(x) &= \text{sgn}\left[\sum_{i=1}^{m} y_i \alpha_i \cdot \left(\Phi(x) \cdot \Phi(x')\right) + b \right] \\ &= \text{sgn}\left[\sum_{i=1}^{m} y_i \alpha_i \cdot k(x, x') + b \right], \\ y_i &: \ \pm 1 \\ \alpha_i &: \ \text{Lagrange multiplier} \\ k(x, x') &: \ \text{kernel, similarity of two examples } x \text{ and } x' \end{aligned} \tag{6}$$

Where k is the kernel that evaluated on input patterns x, x'. Usually there are two kinds of kernel function, i.e., polynomial and radial basis functions (rbf), expressed as Eq. (7) and Eq. (8), respectively.

$$\text{poly:} \quad k(x, x') = \left(\langle x \cdot x' \rangle + 1 \right)^d \tag{7}$$

$$\text{rbf:} \quad k(x, x') = \exp\left(-\frac{\|x - x'\|}{2\sigma^2} \right) \tag{8}$$

Each given face image in the face database with known labels (classes) is modeled by SVM. Several images of the same person under different poses and illuminations are used for training. Recognition is accomplished by matching a test face image with unknown labels against all trained image in the face database.

3.2 One-against-all (OAA) of SVM for multi-class recognition

Basically, the OAA strategy uses a system of N_s binary SVM, where N is the class numbers. More specifically, it involves a parallel architecture made up of N numbers of SVM, and each binary SVM solves a two-class problem defined by one class against all the others. One of the N_s SVM is trained with all of the examples in the *ith* class with positive labels, and all the examples in the other classes except *ith* with negative labels. Thus given k training data $(x_1,y_1),\ldots,(x_k,y_k)$, where $x_i \in R^d$, $i=1,\ldots,k$ and $y_i \in \{1,\ldots,N\}$ is the class of x_i, the *ith* SVM solves the following problem:

$$\begin{aligned}
\min_{w^i, b^i, \xi^i} \quad & \frac{1}{2}(w^i)^T w^i + C \sum_{j=1}^{k} \xi_j^i \\
& (w^i)^T \varphi(x_j) + b^i \geq 1 - \xi_j^i, \quad \text{if } y_j = i, \\
& (w^i)^T \varphi(x_j) + b^i \leq -1 + \xi_j^i, \quad \text{if } y_j \neq i, \\
& \xi_j^i \geq 0, j = 1, \ldots, k
\end{aligned} \tag{9}$$

Where the $\phi(x_j)$ is a kernel function that mapped the training data x_j to a higher dimensional space and C is the penalty parameter. When data with noise causes hard margin, there is a penalty term $C\sum_{j=1}^{k} \xi_j^i$ which can relax the hard margin and allow a possibility of mistrusting the data. It can reduce the number of training errors.

After solving Eq. (9), there are N decision functions as shown in Eq. (10)

$$f_1 = (w^1)^T \varphi(x) + b^1,$$
$$\vdots$$
$$f_N = (w^N)^T \varphi(x) + b^N,$$

$$m(x, f_1, f_2, \ldots, f_N) = \arg \max_{i=1,\ldots,N}(f_i) \tag{10}$$

where f_i is a confidential rate of the output of *ith* SVM, x is the input data, and m is the final decision that found which class has the largest value of the decision function.

3.3 One-against-one (OAO) of SVM for multi-class recognition

In the OAO strategy, several binary SVM are constructed, but each one is constructed by training data from only two different classes. Thus, this method is sometimes called a "pair-

wise" [32,33] approach. For a data set with N different classes, this method constructs $C_2^N=$ $N(N$-$1)/2$ models of two-class SVM. The method also is very popular among researchers in neural networks, Adaboost, decision trees, etc. For training data from the ith and jth classes, we solve the following binary classification problem:

$$
\begin{aligned}
&\min_{w^{ij},b^{ij},\xi^{ij}} && \frac{1}{2}(w^{ij})^T w^{ij} + C\sum_t \xi_t^{ij} \\
&&& (w^{ij})^T \varphi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, \ \ \text{if } y_t = i, \\
&&& (w^{ij})^T \varphi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, \ \ \text{if } y_t = j, \\
&&& \xi_t^{ij} \geq 0,
\end{aligned}
\tag{11}
$$

The simplest decision function is a majority vote or max-win scheme. The decision function counts the votes for each class based on the output from the $N(N$-$1)/2$ SVM. The class with the most votes is the system output.

In the majority vote process, it is necessary to compute each discriminate function $f_{ij}(x)$ of the input data x for the $N(N$-$1)/2$ SVMs model. The score function $R_i(x)$ is sums of the correct votes. The final decision is taken on the basis of the "winner takes all" rule, which corresponds to the following maximization. The expression for final decision is given as Eq. (12).

$$
\begin{aligned}
& f_{ij}(x) = (x * w^n) + b_n, \quad n = 1,\ldots,N \\
& R_i(x) = \sum_{\substack{j=1 \\ j \neq i}}^{N} \text{sgn}\left\{ f_{ij}(x) \right\} \\
& m(x, R_1, R_2, \ldots, R_N) = \arg\max_{i=1,\ldots,N} \left\{ R_i(x) \right\}.
\end{aligned}
\tag{12}
$$

where $f_{ij}(x)$ is the output of $ijth$ SVM, x is the input data and m is the final decision that found which class has the largest voting from the decision function f_{ij}.

3.4 Our multi-stage classifier system for multi-class recognition

Based on coarse-to-fine strategy, the proposed novel scheme for face recognition is a consecutive multi-stage recognition system, in which each stage is devoted to remove a lot of false classes more or less during the recognition process. The flowcharts of the proposed system including the training phase and recognition phase are shown in Fig. 3(a) and (b), respectively. In the first two stages (OAA and OAO of SVM), the obtained DCT features from feature extraction process are used. In the first stage (OAA), we selected the classes whose confidential rate is greater than the threshold, $f_i > T$, which are the subset for the second stage. In the second stage (OAO), the voting strategy is employed to select the top two classes with maximum votes, because the selected top two classes can reach a recognition rate nearly up to 100%. In the third stage, the Euclidian distance of each image of the two classes is calculated, and then the lowest distance of the image of each class is determined for the last stage, i.e., stage four. "RANSAC" method is applied in the last stage, in which the epipolar geometry method with space information of the testing image is matched with the two training images, and then the classified image with the greatest matching numbers of corresponding feature points is selected as the correct one ; in other

words, the training image with the most geometric similarity to the testing image is thus presented.

As mentioned above, the purpose of using DCT is for dimensionality reduction, and the selected low-frequency DCT coefficient vectors are fed into the first two stages for training and testing phase (see Fig. 3(a), (b)). There are two advantages of using the same feature vectors for the first two stages, one is to save the computation cost and the other is to simplify the framework of the recognition system. Essentially, the major difference between the DCT, PCA, and RANSAC are the methodologies of feature extraction while considering what kind of features is the best for classification. For examples, DCT is for frequency transformation, PCA is a statistical technique, and RANSAC is for space geometric information. Three methods can complement each other in our recognition system. The outputs of each stage are collected by a cascade manner as shown in Fig. 3(b), and the final output of last stage accomplished by "RANSAC" method catches a striking and attractive classification end.

In the first stage, OAA testing phase, (see Fig. 3(b)), a scheme with a system of N_s binary SVM where N is the numbers of class is used. As indicated by Eq. (10), there are N decision functions, and f is a confidential rate which is an output from binary SVM, in which $f \in R$ presents. And the formula, $m(x, f_1, f_2, ..., f_N) = \arg\max_{i=1,...,N}(f_i)$, means the most confidential rate. We select a subset part of classes A_j whose confidential rate is larger than the threshold T_j, $A_j = \{f_i \mid f_i \geq T_j\}$, where $i = 1, 2, ..., N$; $j = 1, 2, ..., k$; N are class numbers; k are testing data numbers, $A_j \subset \{C_1, C_2, ..., C_N\}$, C_i = ClassLabels. Then the subset A_j of the class numbers is delivered into second stage in order to shorten the computation time. For example, only the top 10 classes whose confidential rates excess a preset threshold are preserved if there are 40 classes in all; in other words, only the top 10 of the total 40 classes are selected for use in the second stage.

In the second stage OAO model as shown in Fig. 3(b), there are $C_2^N = N(N-1)/2$ models of "pair-wise" SVM. From Eq. (12), $m(x, R_1, R_2, ..., R_L) = \arg\max_{i=1,...,L}\{R_i(x)\}$ is the largest voting from the decision function f_{ij}, where L is the class numbers of subset A, which is screened from first stage. In order to reduce testing error, top two classes with the first and second high voting value are selected, and the difference between their voting values is computed accordingly. If the numerical difference falls less than or equals e, where e is a setting number, these selected two classes are delivered into the third stage for binary classification; $R_i(x) - R_j(x) \leq e$, where $R_i(x)$ and $R_j(x)$ are the classes with the first and second voting value. However, if $R_i(x) - R_j(x) > e$, the *class i* is then be decided as the only correct answer. While the voting value difference falls less than or equals e, it represents a very close difference between *classes i* and *j*, it also tells that there is definitely a need to proceed to the next stage to identify the decision.

In the third stage, the eigenvectors binary model is shaped by the training phase. As shown in the Fig. 4(a), the input image is projected into "face space" and the weights form vectors $\psi^T = [w_1 \ w_2 \ ... \ w_9]$. In our case, only one image of each class is selected by the process of finding the minimum Euclidian distance. Fig. 4(b) shows the Euclidian distance between input image and the ten training images. As indicated in Fig. 4(b), it represents two classes composed of ten training images in all, class 1 includes the first five images, which are images 1, 2, 3, 4, and 5, respectively; and class 2 includes the other five images, which are

images 6, 7, 8, 9, and 10, respectively. Subsequently, the image with the minimum Euclidian distance from each class is picked out for the last stage. As a result, in our case, image 5 out of class 1 and image 9 out of class 2 are decided in stage three.

In the last stage, "RANSAC" method is used to match one testing image with two training images, trying to find which training image best matches with the testing image. It shows that the one with the maximum numbers of corresponding points fits best. The procedure of "RANSAC" is described as following.

a. Find Harris corners [34] in testing and training images: Shifting a window in any direction should give a large change in intensity as shown in Fig. 5(a). The change $E_{x,y}$ produced by a shift (x,y) is given by:

$$E_{x,y} = \sum_{u,v} w_{u,v} [I_{x+u,y+v} - I_{u,v}]^2 \tag{13}$$

where w specifies the image window, for example a rectangular function: it is unity within a specified rectangular region, and zero elsewhere. A Gaussian functions: smooth circular window $w_{u,v} = \exp-(u^2+v^2)/2\sigma^2$.

$I_{u,v}$: image intensity

b. Find putative matches: Among previously detected corner feature points in given image pairs, putative matches are generated by looking for match points that are maximally correlated with each other within given windows. Undoubtedly, only points that robustly correlate with each other in both directions are returned. Even though the correlation matching results in many wrong matches, which is about 10 to 50 percent, it is well enough to compute the fundamental matrix F as shown in Fig. 5(b).

c. Use RANSAC method to locate the corresponding points between the testing and training images: As shown in Fig. 6, the map x → l' between two images defined by fundamental matrix F is considered. And the most basic properties of F is x'Fx =0 [35] for any pair of corresponding points x ↔ x' in the given image pairs. Following steps was used by RANSAC method to consolidate fundamental matrix F estimation:

Repeat

i.   Select random samples of 8 correspondence points.

ii. Compute F.

iii. Measure support (number of inliers within threshold distance of epipolar line).

Choose the F with the largest number of inliers and obtain the corresponding points x_i ↔ x'$_i$ (as shown in Fig. 5(c)).

d. Count numbers of matched and unmatched feature points: The threshold distance between two corresponding points x_i ↔ x'$_l$ is set. Match counts if the distance between two corresponding points is smaller than that of the threshold; on the contrary, no match does. For any given image pairs, the successful match pairs should be the training images with the largest matching number as shown in Fig. 5 (d).

4. Experimental results

The multi-stage-based face recognition is evaluated on three face databases, i.e., the ORL, Yale and IIS face databases as shown in Fig. 7. The ORL face database contains abundant variability in expression, pose and facial details (as shown in Fig. 7 (a)), which is used as a baseline study. The Yale face database is used to evaluate face recognition methods under varying lighting conditions (as shown in Fig. 7(b)) and the IIS face database (as shown in

Fig. 7(c)) is evaluated on a great number of images of 100 subjects, each subject has 30 different images.

We conducted experiments to compare our cascade multi-stage classifier strategy with some other well-known single classifier, e.g., the SVM with OAA, OAO, and Eigenface. The experimental platforms are Intel Celeron 2.93GHz processor, 1GB DDRAM, Windows XP, and Matlab 7.01.

4.1 Face recognition on the ORL database

The first experiment is performed on the ORL database. There are 400 images of 40 distinct subjects. Each subject has ten different images taken at different times. Each image was digitized a 112×92 pixel array whose gray levels ranged between 0 and 255. One sample subject of the ORL face images is shown in Fig. 7 (a). There are variations in facial expressions such as open/closed eyes, smiling/non-smiling, and glasses/no glasses. In our experiments, five images are randomly selected as training samples, the other five images and then serve as testing images. Therefore, for 40 subjects in the database, a total of 200 images are used for training and another 200 for testing and there are no overlaps between the training and testing sets. Here, we verify our system based on the average error rate. Such procedures are repeated for four times, i.e. four runs, which result in four groups of data. For each group, we calculated the average of the error rates versus the number of feature dimensions (from 15 to 100). Fig. 8 shows the results of the average of four runs and the output of each stage from the multi-stage classifier, which are SVM with OAA, OAO, Eigenface, and final stage. As shown in Fig. 8, the error rates of the output of the final stage is lower than the other three types of single classifier, our proposed method obtains the lowest error rate. The average minimum error rate of our method is 1.37% on the 30 feature numbers, while the OAA-SVM is 10.50%, OAO-SVM is 2.87%, and Eigenface is 8.50%. If we choose the best results among the four groups of the randomly selected data, the lowest error rate of the final stage can achieve 0%.

4.2 Comparison with previous reported results on ORL

Several approaches have been conducted for face recognition using the ORL database [5,6,8,9,11,13,36-41,43]. The methods of using single classifier systems for face recognition are Eigenface [5,37,38,40], DCT-RBFNN [6], binary tree SVM [8], 2D-HMM [9], LDA [39], and NFS [43]. The methods of using multi-classifiers for ORL face recognition are fuzzy fisherface [11,41], and CF²C [13]. Here, we present a comparison under similar conditions between our proposed method and the other methods described on the ORL database. Approaches are evaluated on error rate, and feature vector dimension. Comparative results of different approaches are shown in Table 1. It is hard to compare the speed of different methods performed on different computing platforms, so we ignore the training and recognition time in each different approach. It is evident as indicated in the table that the proposed approach achieves best recognition rate in comparison with the other three approaches. In other words, our approach outperforms the other three approaches in respect of recognition rate.

4.3 Face recognition on the Yale database

The second experiment was conducted on the Yale face database, which contains 165 centered face images of 15 individuals and 11 images per person with major variations,

including changes in illumination conditions (center-light, left-light, right-light), glasses/ no glasses, and different facial expressions (happy, sad, winking, sleepy, and surprised). The original images are 256 grayscale levels with a resolution of 195×231. The training and testing set are selected randomly with five training and six testing samples per person at four times. Similarly, we verify the proposed system based on the average error rate obtained from our four experimental results and calculate the error rates versus the number of feature dimensions (from 15 to 100). Fig. 9 shows the results of the average of four runs and the output of each stage from the multi-stage recognition system, which are SVM with OAA, OAO, Eigenface, and final stage. As shown in Fig. 9, the error rates of the output of the final stage is lower than the other three types of single classifier. The average minimum error rate of our method is 0.27% in the 65 feature numbers, while the OAA-SVM is 2.50%, OAO-SVM is 0.82%, and Eigenface is 10.27%. If we choose the best results among the four groups of verified data, the lowest error rate of the final stage can even reach 0%.

4.4 Comparison with previous reported results on Yale
Several approaches have been conducted for face recognition by using Yale database [6,11,13,25,40]. The methods of using Yale database in single classifier systems for face recognition are 2D-PCA [40], DCT-RBFNN [6], and DCT-NNC [25]. The methods of using multi-classifier systems for face recognition are fuzzy fisherface [41], and CF2C [13]. Here, the comparison of the classification performance of all the methods is provided in Table 2. Again, Table 2 clearly indicates that the proposed approach outperforms the other five approaches.

4.5 Face recognition on the IIS database
The IIS face database contains 3000 face images of 100 individuals. There are 30 images per subject, the images per subject include tens for frontal face, tens for left profile, and tens for right profile. Each image was digitized and presented by 175×155 pixel array whose gray levels ranged between 0 and 255 as shown in Fig. 7(c). The training and testing set are selected randomly. This split procedure has been repeated four times in each case. Six images of each subject are randomly selected for training, and the remaining 24 images are for testing. The result is shown in Fig. 10 and Table 3 gives evidence that the proposed method outperformed other classification techniques.

5. Conclusions

This paper presents a multi-stage classifier method for face recognition based on the techniques of SVM, Eigenface, and RANSAC. The proposed multi-stage method is based on a coarse-to-fine strategy, which can reduce the computation cost. The facial features are first extracted by the DCT for the first two stages, i.e., OAA-SVM and OAO-SVM. Through all our experiments, OAO-SVM obtained a higher recognition rate than the OAA-SVM, so in our research, we put the OAO after the OAA. Although the last stage (RANSAC) led to more accuracy in comparison with the other three stages, its computation cost more in the geometric fundamental matrix F estimation. In order to decrease the computation time, we need to reduce the classes and images to only two training images to match with testing image in the last stage. The key of this method is to consolidate OAO-SVM for the output of the top two maximum votes so that the decision of the correct class could be made later by RANSAC in the last stage. The feasibility of the proposed approach has been successfully tested on ORL, Yale, and IIS face databases, which are acquired under varying pose,

illumination, expression, and a great quantity of samples. Comparative experiments on the three face databases also show that the proposed approach is superior to single classifier and multi-parallel classifier.

6. References

[1] R. Chellappa, C. L. Wilson, S.Sirohey, Human and machine recognition of faces: a survey, in Proc. IEEE, 83 (5) (1995) 705-740.

[2] A. Samal, P. A. Iyengar, Automatic recognition and analysis of human faces and facial expressions: a survey, Pattern Recognition, 25 (1992) 65-77.

[3] D. Valentin, H. Abdi, A. J. O'Toole, G. W. Cottrell, Connectionist models of face processing: a survey, Pattern Recognition, 27 (1994) 1209-1230.

[4] R. Brunelli, T. Poggio, Face recognition: Features versus templates, IEEE Trans. Pattern Anal. Mach. Intell., 15 (10) (1993) 1042-1053.

[5] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience, 3 (1991) 71-86.

[6] M. J. Er, W. Chen, S. Wu, High-Speed Face Recognition Based on Discrete Cosine Transform and RBF Neural Networks, IEEE Trans. Neural Networks, 16 (3) (2005) 679-691.

[7] C. Xiang, X. A. Fan, T. H. Lee, Face Recognition Using Recursive Fisher Linear Discriminant, IEEE Trans. on Image Processing, 15 (8) (2006) 2097-2105.

[8] G. Guo, S. Z. Li, K. L. Chan, Support vector machines for face recognition, Image and Vision Computing, 19 (2001) 631-638.

[9] H. Othman, T. Aboulnasr, A Separable Low Complexity 2D HMM with Application to Face Recognition, IEEE Trans. on Pattern Analysis and Machine Intelligence, 25 (10) (2003) 1229-1238.

[10] J. Lu, K. N. Plataniotis, A. N. Venetsanopoulos, S. Z. Li, Ensemble-Based Discriminant Learning With Boosting for Face Recognition, IEEE Trans. on Neural Networks, 17 (1) (2006) 166-178.

[11] K. C. Kwak, W. Pedrycz, Face recognition: A study in information fusion using fuzzy integral, Pattern Recognition Letters, 26 (2005) 719-733.

[12] A. N. Rajagopalan, K. S. Rao, Y. A. Kumar, Face recognition using multiple facial features, Pattern Recognition Letters, 28 (2007) 335-341.

[13] D. Zhou, X. Yang, N. Peng, Y. Wang, Improved-LDA based face recognition using both facial global and local information, Pattern Recognition Letters, 27 (2006) 536-543.

[14] Z. Q. Zhao, D. S. Huang, B. Y. Sun, Human face recognition based on multi-features using neural networks committee, Pattern Recognition Letters, 25 (2004) 1351-1358.

[15] A. Lemieux, M. Parizeau, Flexible multi-classifier architecture for face recognition systems, in 16th Int. Conf. on Vision Interface, (2003).

[16] J. C. Peincipe, N. R. Euliano, W. C. Lefebvre, Neural and Adaptive Systems, John Wiley & Sons, Inc., 1999.

[17] A. V. Nefian, M. H. Hayes, Face Detection and Recognition using Hidden Markov Models, in Proc. IEEE Int. Conf. Image Processing, 1 (1998) 141-145.

[18] H. Othman, T. Aboulnasr, A Separable Low Complexity 2D HMM with Application to Face Recognition, IEEE Trans. Pattern Anal. Mach. Intell. , 25 (10) (2003) 1229-1238.

[19] M. S. Kim, D. Kim, S. Y. Lee, Face recognition using the embedded HMM with second-order block-specific observations, Pattern Recognition, 36 (2003) 2723-2735.

[20] S. Eickeler, S. Birlinghoven, Face Database Retrieval Using Pseudo 2D Hidden Markov Models, in Proc. of Fifth IEEE Int. Conf. on Automatic Face and Gesture Recognition, (2002) 58-63.

[21] C. W. Hsu, and C. J. Lin, A comparison of methods for multi-class support vector machines, IEEE Trans. Neural Network, 13 (2) (2002) 415-425.

[22] V. Vapnik, Statistical Learning Theory, John Wiley & Sons, Inc., 1998.

[23] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, Large margin DAGs for multiclass classification, In Advances in Neural Information Processing Systems, MIT Press, 12 (2000) 547-553.

[24] G. Ratsch, T. Onoda, K. R. Muller, Soft Margins for AdaBoost, Machine Learning, 42 (2001) 287-320.

[25] X. Y. Jing, D. Zhang, A Face and Palmprint Recognition Approach Based on Discriminant DCT Feature Extraction, IEEE Trans. on Systems, Man, and Cyb., 34 (6) (2004) 2405-2415.

[26] M. A. Fischler, R. C. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Communications of the ACM, 24 (6) (1981) 381-395.

[27] Ara V. Nefian and Monson H. Hayes III, Hidden Markov Models for Face Recognition, in Proc. IEEE Int. Conf. Acoustic, Speech, and Signal Processing, 5 (1998) 2721-2724.

[28] V. V. Kohir, U. B. Desai, Face recognition using a DCT-HMM approach, in Proc. 4th IEEE Int. Conf. Application of Computer Vision, (1998) 226-231.

[29] M. Bicego, U. Castellani, V. Murino, Using Hidden Markov Models and Wavelets for Face Recognition, in Proc. 12th IEEE Int. Conf. Image Analysis and Processing, (2003) 52-56.

[30] L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, L. Jackel, Y. LeCun, U. Muller, E. Sackinger, P. Simard, and V. Vapnik. Comparison of classifier methods: a case study in handwriting digit recognition. In International Conference on Pattern Recognition, IEEE Computer Society Press, (1994) 77-87.

[31] Ben-Hur Asa, David Horn, T.H. Siegelmann, Vladimir Vapnik, Support Vector Clustering, J. of Machine Learning Research, (2001) 125-137.

[32] D. Price, S. Knerr, Pairwise neural network classifiers with probabilistic outputs, Neural Information Processing Systems, 7 (1994)

[33] T. Hastie, R. Tibshirani, Classification by pairwise coupling, in Conf. on Advances in Neural Information Processing Syatems, MIT Press, Cambridge, MA, (1998) 507-513.

[34] C. Harris, M. Stephens, A Combined Corner and Edge Detector, In 4th Alvey Vision Conference, (1988) 147-151.

[35] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, second ed., Cambridge University Press, 2003.

[36] C. H. Chen, C. T. Chu, High efficiency feature extraction based on 1-D wavelet transform for real-time face recognition, WSEAS Trans. on Information Science and Application, 1 (2004) 411-417.

[37] B. Li, Y. Liu, When eigenfaces are combined with wavelets, Knowledge-Based Systems, 15 (2002) 343-347.

[38] T. Phiasai, S. Arunrungrusmi, K. Chamnongthai, Face recognition system with PCA and moment invariant method, in Proc. of the IEEE International Symposium on Circuits and Systems, 2 (2001) 165-168.

[39] J. Lu, K. N. Plataniotis, A. N. Venetsanopoulos, Face recognition using LDA-based algorithms, IEEE Trans. on Neural Networks, 14 (2003) 195-200.

[40] J. Yang, D. Zhang, A. F. Frangi, J. Y. Yang, Two-dimensional PCA: a new approach to appearance-based face representation and recognition, IEEE Trans. on Pattern Analysis and Machine Intelligence, 26 (2004) 131-137.

[41] K. C. Kwak, W. Pedrycz, Face recognition using a fuzzy fisferface classifier, Pattern Recognition, 38 (2005) 1717-1732.

[42] C. T. Chu, C. H. Chen, J. H Dai, Multiple Facial Features Representation for Real-Time Face Recognition, Journal of Information Science and Engineering, 22 (2006) 1601-1610.
[43] J. T. Chien, C. C. Wu, Discriminant waveletfaces and nearest feature classifiers for face recognition, IEEE Trans. on Pattern Analysis and Machine Intelligence, 24 (2002) 1644-1649.
[44] ORL face database, (http://www.uk.research.att.com/facedatabase.html).
[45] Yale face database, (http://cvc.yale.edu/projects/yalefaces/yalefaces.html).
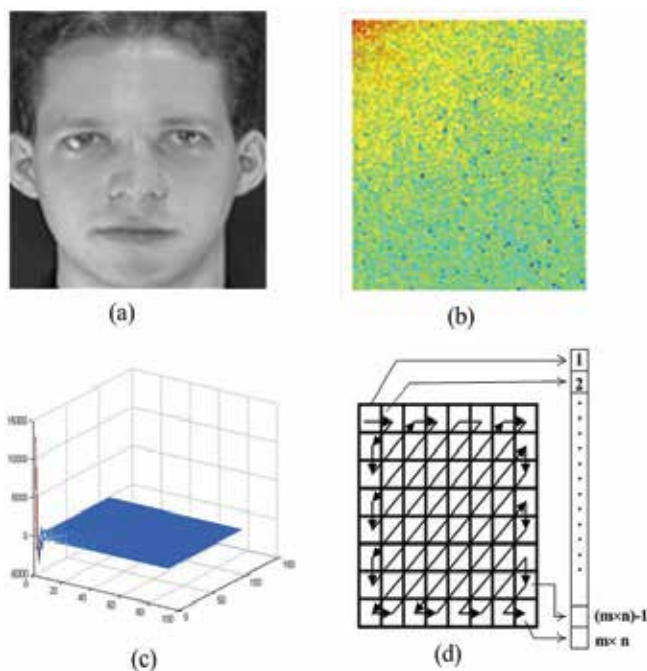[46] IIS face database, (http://smart.iis.sinica.edu.tw/).

Fig. 1. Facial image and DCT transform image (a) original face image (b) 2D plot after 2D-DCT (c) 3D plot after 2D-DCT (d) Scheme of zig-zag method to extract 2D-DCT coefficients to a 1D vector
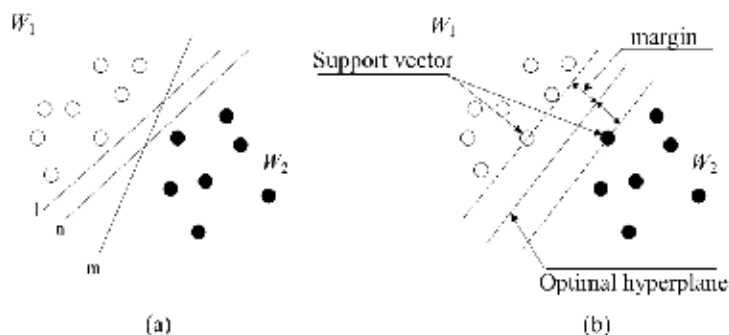

Fig. 2. Classification between two classes W_1 and W_2 using hyperplanes: (a) arbitrary hyperplanes l, m, and n. (b) the optimal separating hyperplane with the largest margin identified by the dashed line, passing the two support vectors.
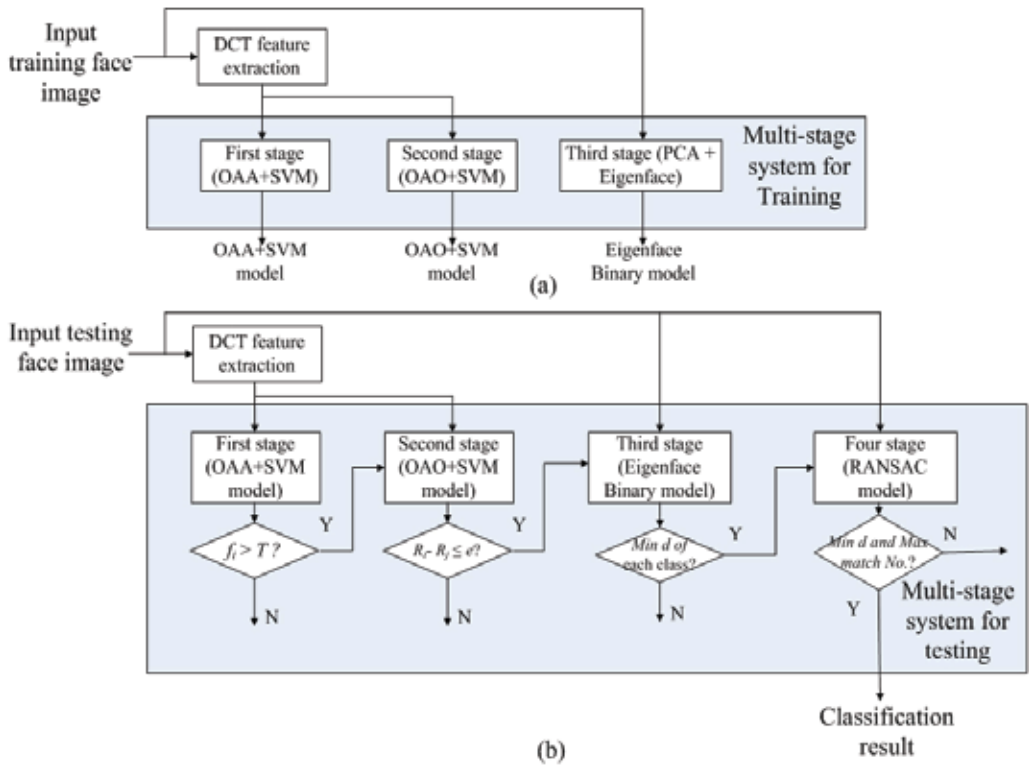
(a)

(b)

Fig. 3. Flowchart of the face recognition system. (a) training phase (b) testing phase.
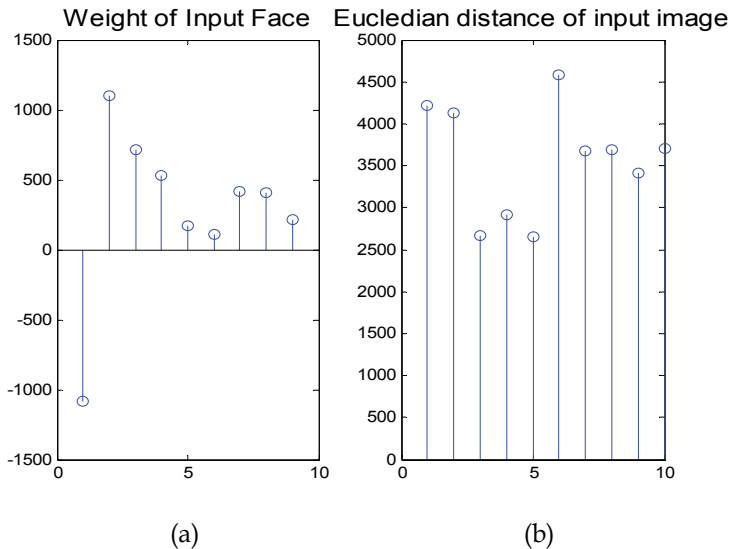

(a)   (b)

Fig. 4. (a) The weight vector $\psi^T = [w_1 \, w_2 \ldots w_9]$ of input image. (b) The Euclidian distance between the input image and ten training samples, respectively. The sample 1, 2, 3, 4, 5 are the same classes and sample 6, 7, 8, 9, 10 are another classes.
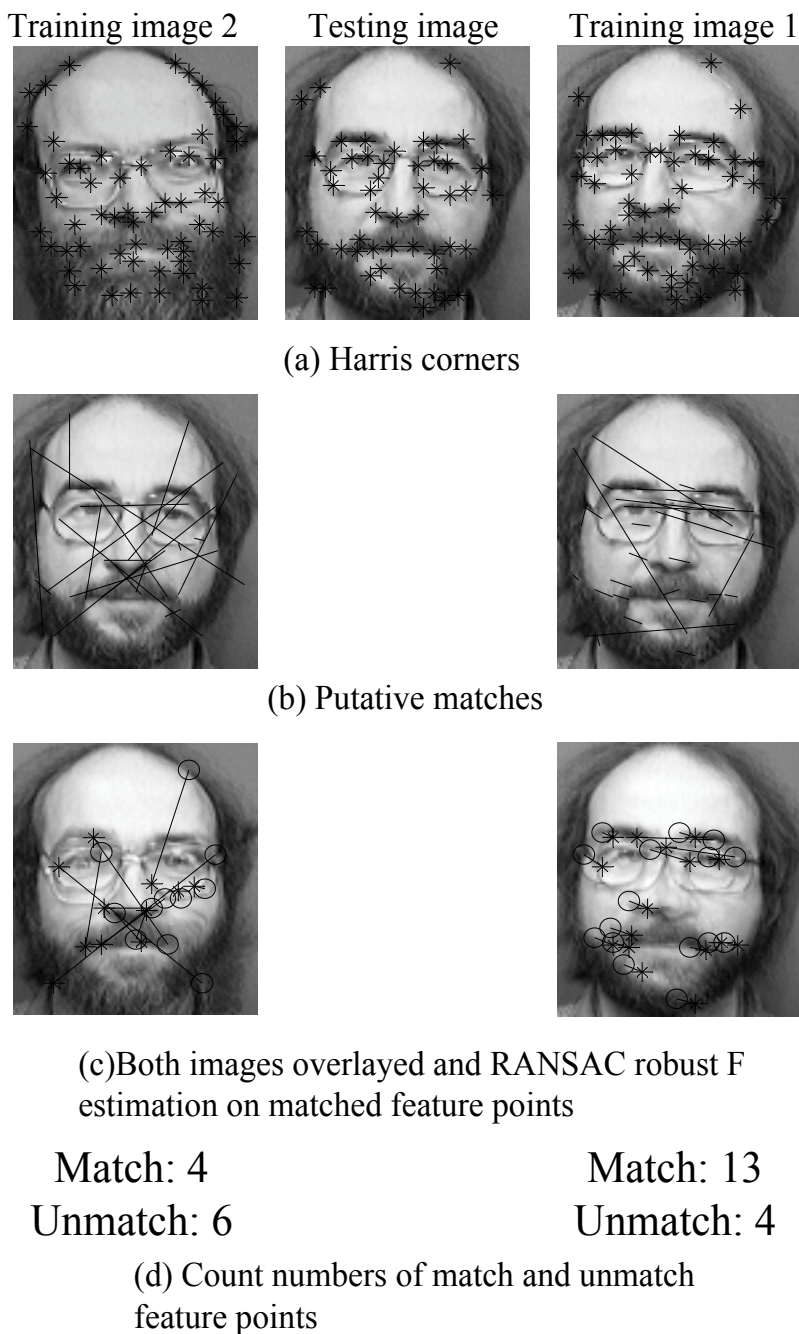
Training image 2   Testing image   Training image 1

(a) Harris corners

(b) Putative matches

(c)Both images overlayed and RANSAC robust F
estimation on matched feature points

Match: 4   Match: 13

Unmatch: 6   Unmatch: 4

(d) Count numbers of match and unmatch
feature points

Fig. 5. Four procedures of space information using RANSAC method to find the match and
unmatch feature points. (a) Find Harris corners feature points in one testing and two
training images. (b) Find putative matches of testing and training images. (c) Using
RANSAC method to find testing and training images of match feature points. (d) Count
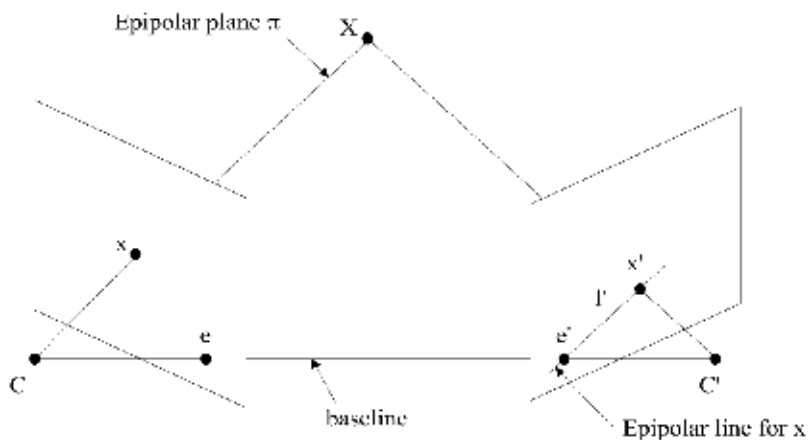numbers of match and unmatch feature points.

Fig. 6. Point correspondence geometry. The two cameras are indicated by their centers C and C' and image planes. The camera centers, 3-space point X, and its images x and x' lie in a common plane π. An image point x back-projects to a ray in 3-space defined by the first camera center, C, and x. This ray is imaged as a line l' in the second view. The 3-space point X which projects to x must lie on this ray, so the image of X in the second view must lie on l'.
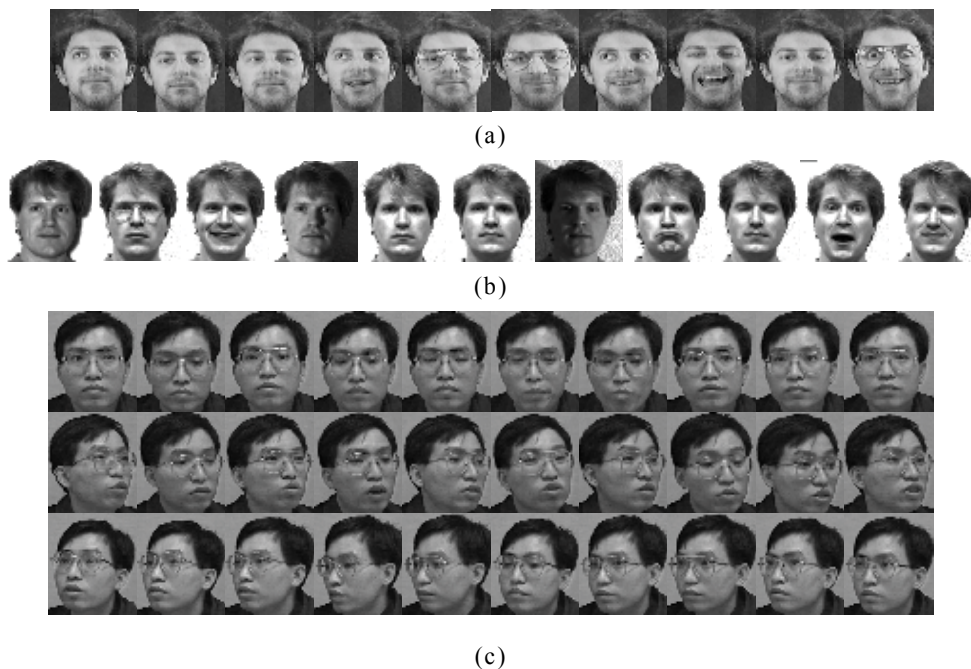

(a)

(b)

(c)

Fig. 7. Some sample images from publicly available face database used in the experiments: (a) ORL face database, (b) Yale face database, (c) IIS face database.
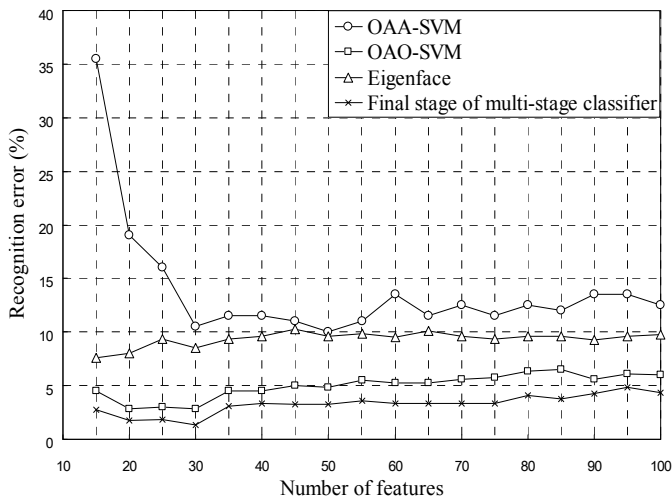
Fig. 8. Comparison of recognition error versus the number of features of the OAA-SVM, OAO-SVM, Eigenface, and final stage of the Multi-stage classifier system on the ORL face database.

Fig. 9. Comparison of recognition error versus the number of features of the OAA-SVM, OAO-SVM, Eigenface, and final stage of the Multi-stage classifier system on the Yale face database.
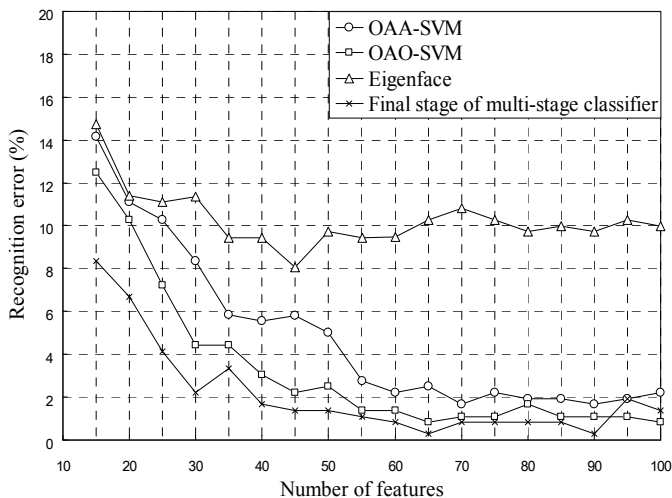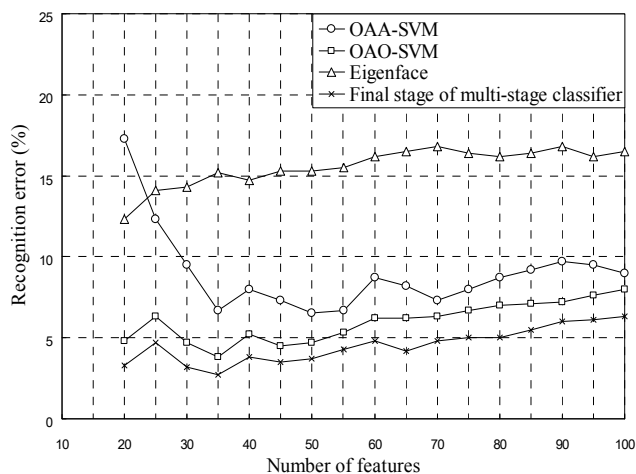
Fig. 10. Comparison of recognition error versus the number of features of the OAA-SVM, OAO-SVM, Eigenface, and final stage of the Multi-stage classifier system on the IIS face database.

Methods	Error rate (%)		Feature vector dimension
	Best	Mean	
Wavelet + Eigenface [37]	2	4	140
2D-PCA [40]	4	5	112×3
Binary tree SVM [8]	N/A	3	48
DCT-RBFNN [6]	0	2.45	30
CF²C [13]	3	4	30
Fuzzy Fisherface [41]	2.5	4.5	60
Our proposed approach	0	1.375	30

Table 1. Recognition performance comparison of different approaches (ORL)

Methods	Error rate (%)		Feature vector dimension
	Best	Mean	
2D-PCA [40]	N/A	15.76	139/165
DCT-RBFNN [6]	N/A	1.8	N/A
DCT-NNC [25]	2.22	2.22	59
CF²C [13]	N/A	3.1	14
Fuzzy Fisherface [41]	2.16	5.2	40
Our proposed approach	0	0.27	65

Table 2. Recognition performance comparison of different approaches (Yale)

Methods	Error rate (%)		Feature vector dimension
	Best	Mean	
Discriminant waveletface + NFS [43]	3.6	N/A	60
DWT + PNN [36]	8.83	N/A	24
Multi-feature + DWT + PNN [42]	3.08	N/A	70
Our proposed approach	1.8	2.7	35

Table 3. Recognition performance comparison of different approaches (IIS)

PCA-ANN Face Recognition System based on Photometric Normalization Techniques

Shahrin Azuan Nazeer and Marzuki Khalid
Telekom Research & Development Sdn. Bhd., Universiti Teknologi Malaysia

1. Introduction

The human face is the main focus of attention in social interaction, and is also the major key in conveying identity and emotion of a person. It has the appealing characteristic of not being intrusive as compared with other biometric techniques. The research works on face recognition started in the 1960s with the pioneering work of Bledsoe and Kanade, who introduced the first automated face recognition system (Zhao *et al*, 2003). From that onwards, the research on face recognition has widespread and become one of the most interesting research area in vision system, image analysis, pattern recognition and biometric technology.

Recently, research on face recognition has received attention and interest from the scientific community as well as from the general public. Face recognition has become a major issue in many security, credit card verification, and criminal identification applications due to its applicability as a biometric system in commercial and security applications to prevent unauthorized access or fraudulent use of Automated Teller Machines (ATMs), cellular phones, smart cards, desktop personal computers, workstations and computer networks. Face recognition has been used by law enforcement agencies for finding criminals, by government agencies for fraud and homeland security, and by financial institutions for ATM and check-cashing security to protect customers against identity theft and fraudulent transactions. By using the face recognition, a picture identity, bankcard or Personal Identification Number (PIN) is no longer needed to verify a customer's identity. Face recognition is also applicable in areas other than security oriented applications such as computer entertainment and customized computer-human interaction applications that can be found in products such as cars, aids for disabled people, or buildings. The interest for face recognition will most likely increase even more in the future due to the increased penetration of technologies, such as digital cameras and the internet, and a larger demand for different security schemes.

Face recognition systems (FRS) are still in their infancy. The current FRS still experiencing low accuracy rates due to factors such as illumination, orientation and other disturbances. The quality of a face image has also a big impact on the performance of the FRS. If the illumination on the face image is too high, the face image will be too bright; however, if the illumination is too low, the face image will be too dark. The variation on the illumination will greatly affect the quality of the face image and reduce the performance of the FRS. Thus, it is crucial to improve the quality of the face image in order to have a better

performance. In this work, we proposed to improve the face image quality by using photometric normalization techniques which will normalize the illumination variation of the face image. The techniques used are based on Histogram Equalization and Homomorphic Filtering. A face recognition system based on Principal Component Analysis (PCA) followed by Artificial Neural Networks (ANN) called PCA-ANN is proposed. PCA is used during the feature extraction phase since it is found to be the simple and popular technique used for feature extraction (Martinez et al, 2001; Li & Jain, 2005).

Sirovich and Kirby (1987) proposed the use of PCA to obtain a reduced representation of face images. Kirby and Sirovich (1991) then proposed the use of PCA for face analysis and representation. The work was followed by Turk and Pentland (1991) who first applied PCA for face recognition and named as "Eigenfaces" technique since the basis vectors constructed by PCA had the same dimension as the input face images. They used PCA as the projection scheme to obtain the feature vectors or Eigenfaces, and Euclidean distance as similarity function to solve face recognition problem. Lee and Lu (1998) proposed a method for generalizing the representational capacity of available face database. Lee et al. (1999) later proposed a method using PCA which detects the head of an individual in a complex background and then recognize the person by comparing the characteristics of the face to those of known individuals.

Meanwhile, the ANN based on feed-forward neural networks is used during classification phase because it is one of the machine learning algorithms which is widely used for classification. ANN have been commonly used as the classifier for FRS generally in a geometrical local feature based manner, but there are also some methods where ANN are applied holistically(Nazeer et al, 2007a, 2007b). Lades et al. (1993) presented an object recognition system based on Dynamic Link Architectures (DLA) which is an extension of the ANN. The DLA uses correlations in the fine-scale cellular signals to group neurons dynamically into higher order entities. These entities can be used to code high-level objects, such as a 2-D face image. The face images are represented by sparse graphs, whose vertices are labeled by a multi-resolution description in terms of local power spectrum, and whose edges are labeled by geometrical distance vectors. Lawrence et al. (1997) presented a hybrid neural-network solution that combines local image sampling, a self-organizing map (SOM) neural network, and a convolutional neural network for face recognition. The SOM provides a quantization of the image samples into a topological space where inputs that are nearby in the original space are also nearby in the output space, thereby providing dimensionality reduction and invariance to minor changes in the image sample, and the convolutional neural network provides for partial invariance to translation, rotation, scale, and deformation. The convolutional network extracts successively larger features in a hierarchical set of layers. Thomaz et al. (1998) also studied on ANN by combining PCA and RBF neural network. Their system is a face recognition system consisting of a PCA stage which inputs the projections of a face image over the principal components into a RBF network acting as a classifier. Their main concern is to analyze how different network designs perform in a PCA+RBF face recognition system. They used a forward selection algorithm, and a Gaussian mixture model. According to the results of their experiments, the Gaussian mixture model optimization achieves the best performance even using less neuron than the forward selection algorithm. Their results also show that the Gaussian mixture model design is less sensitive to the choice of the training set.

Temdee et al. (1999) presented a frontal view face recognition method by using fractal codes which are determined by a fractal encoding method from the edge pattern of the face region

(covering eyebrows, eyes and nose). In their recognition system, the obtained fractal codes are fed as inputs to back-propagation neural networks for identifying an individual. They tested their system performance on the ORL face database, and reported a recognition rate of 85 %. Er *et al.* (2002) suggested the use of Radial Basis Function (RBF) neural networks on the data extracted by discriminant eigenfeatures. They used a hybrid learning algorithm to decrease the dimension of the search space in the gradient method, which is crucial on optimization of high dimension problem. First, they tried to extract the face features by both the PCA and Linear Discriminant Analysis methods. Next, they presented a hybrid learning algorithm to train the RBF neural networks, where the dimension of the search space is significantly decreased in the gradient method. Tolba and Abu-Rezq (2000) presented an invariant face recognition using the generalization capabilities of both Learning Vector Quantization (LVQ) and RBF neural networks to build a representative model of a face from a variety of training patterns with different poses, details and facial expressions. The combined generalization error of the classifier was found to be lower than that of each individual classifier. A new face synthesis method was implemented for reducing the false acceptance rate and enhancing the rejection capability of the classifier. The system was capable of recognizing a face in less than one second using ORL database for testing the combined classifier.

The chapter is organized as follows. The first part of the chapter provides the system overview of an automated face recognition system. The second part of the chapter elaborates on the methodology used for the proposed face recognition system which includes the photometric normalization techniques, Histogram Equalization and Homomorphic Filtering, the feature extraction technique using PCA and classification using ANN. The final part of the chapter explains the performances of the proposed face verification system using both original face image and the face image with photometric normalization.

2. System overview

An automated face recognition system consists of two main parts which are face detection and face recognition as depicted in Fig. 1. Each of these parts will be described in the following sections.

2.1 Face detection

Face detection is essentially the first fundamental step or front-end of any online face recognition system. It is used to determine whether there is human face in a scene either obtained from camera or still image. It then identifies where the human face is. If the human face is identified, it outputs a human face image consisting of the eyes, the nose and the mouth. The human face is identified using direct image processing techniques which determines the locations and sizes of the face in scene image, and separates them from other non-face objects and distracting background information. The face alignment which involves translation, rotation, and scaling is carried out using the center or edges of the eyes as a reference point since the eyes are an important feature that can be consistently identified.

In Fig. 1, the scene image is captured using a web camera. The captured color image is transformed into a grayscale image. Face localization is applied to determine the image position of a face in the scene image. When the face image is detected, eyes detection is applied to detect the presence and location of the eyes in an image. The eyes location is used

for face alignment to correct the orientation of the face image into an upright frontal face image using affine transformation. The geometrical normalization is used to crop the upright frontal face image and scale it to a desired resolution. The cropped face image is used as the input image for face recognition.
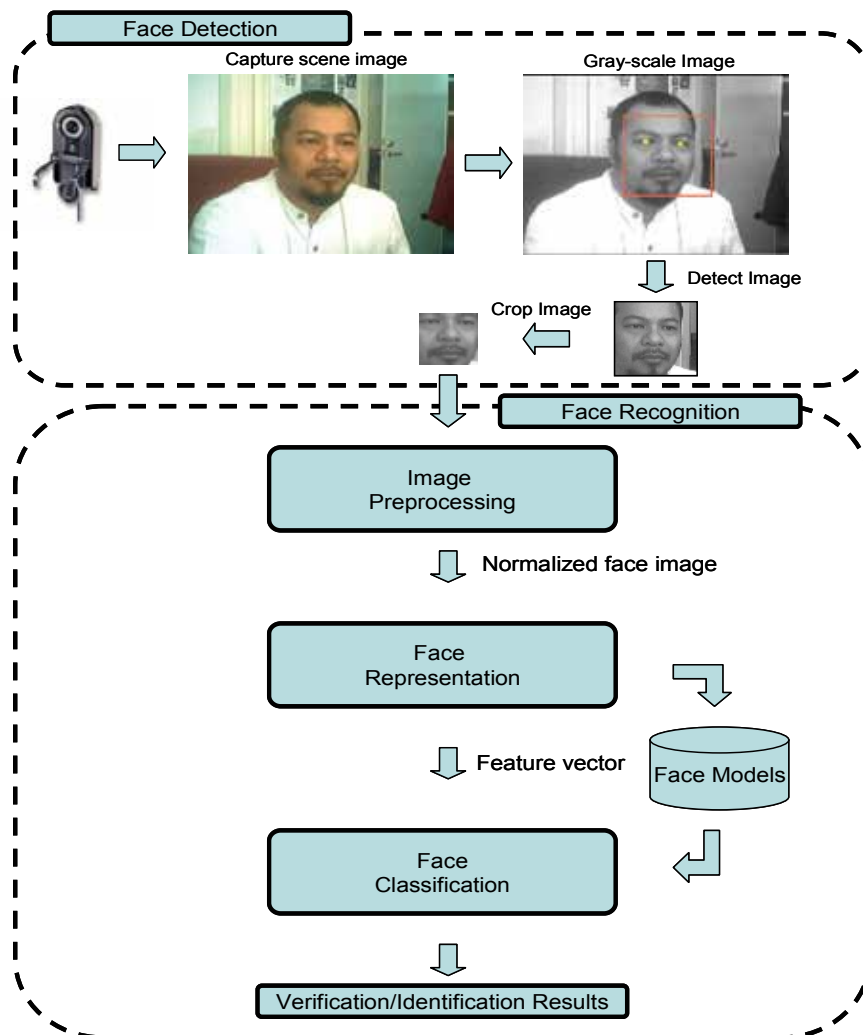

Fig. 1. The main parts of an automated face recognition system

2.2 Face recognition

Face recognition establishes the identity of a person based on the person's face with reference to a database of known faces. In Fig. 1, face recognition comprises of three main stages which are the image preprocessing, face representation, and face classification. Each of these stages is explained in the following subsections.

(a) Image Preprocessing

The aim of the image preprocessing is to preprocess a face image to enhance the data, remove noise and segment out the crucial data. The image preprocessing involves face

normalization which is used to compensate or normalize a face for position and illumination so that the variance due to these is minimized. A histogram of the face alone is computed to compensate for lighting changes in the image. Consequently, the small variation in the image due to identity, or muscle actuation will become the dominant source of intensity variance in the image and can thus be analyzed for recognition purposes.

(b) Face Representation

Face representation is used to generate a low-dimensional feature representation intrinsic to face objects with good discriminatory power for pattern classification using feature extraction. Feature extraction refers to applying a mapping or transformation of the multidimensional space into a space of fewer dimensions. The aim of feature extraction is to extract a compact set of interpersonal discriminating geometrical or photometrical features of the face. Feature extraction analyzes the variances left in the image using linear statistical techniques to generate a low-dimensional feature representation intrinsic to face objects with good discriminatory power for pattern classification. These techniques are used to classify and characterize the variances that remain in the image since they are now constrained and limited. After been preprocessed to such conditioned data, the unique features are then extracted and a template is then generated to represent the face image. The template will be the basis form to associate the uniqueness of the data with the identity of the user.

(c) Face Classification

Face classification or feature matching is the actual recognition process. Given the feature representation of face objects, a classifier is required to learn a complex decision function to implement the final classification. The feature representation is optimized for the best discrimination which would help reduce the complexity of the decision function, and ease the classifier design. Thus, a good classifier would be able to further learn the different between the subjects. The feature vector obtained from the feature extraction is classified or matched to classes (persons) of facial images already enrolled in a database. The classification or feature matching algorithms used vary from the nearest neighbor classifier such as the Euclidean distance to advanced schemes like Artificial Neural Network (ANN).

If the user is using the face recognition system for the first time, the user will be registered and the user template will be stored for future references. Face recognition involves comparing the generated template against the stored reference template. For verification, the matching is made against a claimed identity, where the matching process will be a one to one comparison between the generated template and the stored reference template. For identification, the matching is made by comparing the generated template against a list of reference templates of legitimate users which will be a one to many comparisons.

3. Methodology

3.1 Photometric normalization

The purpose of the image preprocessing module is to reduce or eliminate some of the variations in face due to illumination. The image preprocessing is crucial as the robustness of a face recognition system greatly depends on it. By performing explicit normalization processes, system robustness against scaling, posture, facial expression and illumination is increased. The image preprocessing includes photometric normalization which removes the mean of the geometrically normalized image and scales the pixel values by their standard deviation, estimated over the whole cropped image. The photometric normalization techniques applied are based on Histogram Equalization and Homomorphic Filtering.

a) Histogram Equalization

Gray level transformation is an image processing system that looks at every input pixel gray level and generates a corresponding output gray level according to a fixed gray level map. Histogram Equalization is the most common histogram normalization or image-specific gray level transformation used for contrast enhancement with the objective to obtain a new enhanced image with a uniform histogram or to produce an image with equally distributed brightness levels over the whole brightness scale. Histogram equalization is usually achieved by equalizing the histogram of the image pixel gray-levels in the spatial domain so as to redistribute them uniformly. It is usually done on too dark or too bright images in order to enhance image quality and to improve face recognition performance. It modifies the dynamic range (contrast range) of the image and as a result, some important facial features become more apparent.

Histogram Equalization arranges the gray-levels of the image by using the histogram form information. Histogram, an array of 256 elements containing the counts or number of pixels of all gray levels, is applied by Histogram Equalization to generate a special gray level mapping suited for a particular image. The accumulated density function of the histogram for the processed image histogram would approximate a straight line. The redistribution of pixel brightness to approximate the uniform distribution improves the contrast of the image. The result of this process is that the histogram becomes approximately constant for all gray values. The steps for Histogram Equalization algorithm are depicted in Fig. 2.

1 Model *H(i)* as the histogram function of an image and *G(i)* as the desired histogram to be mapped via a transformation *fHG(i)*

2 Compute the transformation functions for both *H(i)* and *G(i)* , $f_{HU}(i)$ and $f_{GU}(i)$ to map the histogram to a uniform distribution, *U(i)*.

3 Map to a uniform distribution (Histogram Equalization) based on the following equation:

$$f_{HU}(i) = \frac{\sum_{j=0}^{i} H(i)}{\sum_{j=0}^{i-1} H(i)} \qquad f_{GU}(i) = \frac{\sum_{j=0}^{i} G(i)}{\sum_{j=0}^{n-1} G(i)}$$

where: *n* is the number of discrete intensity levels and is set to 256 for 8-bit gray-scale image.
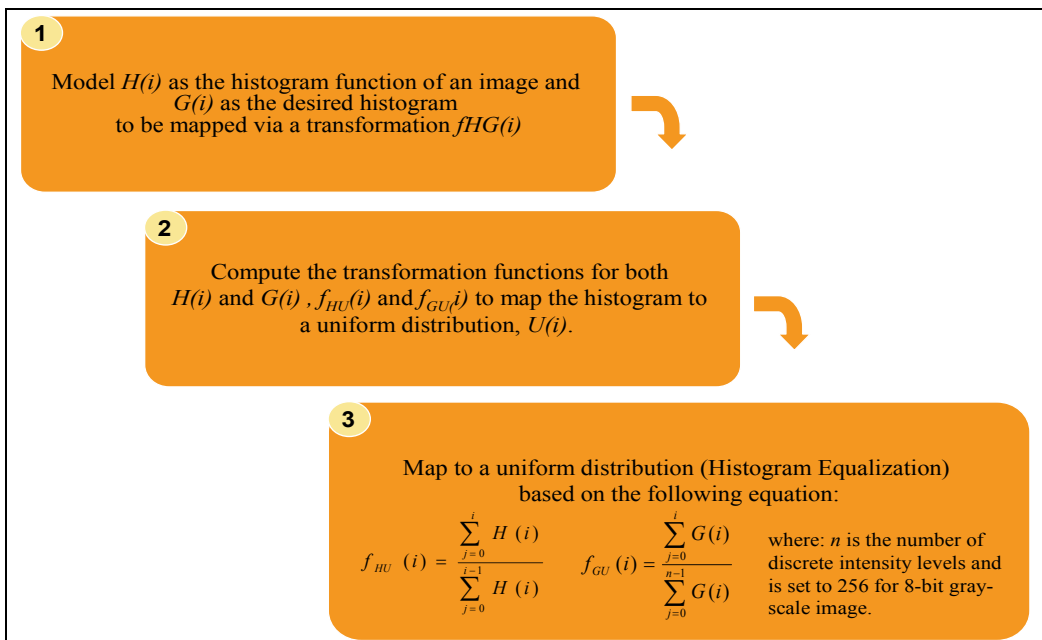
Fig. 2. Histogram Equalization algorithm

b) Homomorphic Filtering

Homomorphic Filtering algorithm is similar to that of Horn's algorithm except the low spatial frequency illumination is separated from the high frequency reflectance by Fourier high-pass filtering. In general, a high-pass filter is used to separate and suppress low

frequency components while still passing the high frequency components in the signal. If the two types of signal are additive then the actual signal is the sum of the two types of signals. However, in this illumination or reflection problem, low-frequency illumination is multiplied instead of added to the high-frequency reflectance. To still be able to use the usual high-pass filter, the logarithm operation is needed to convert the multiplication to addition. After the Homomorphic Filtering process, the processed illumination should be drastically reduced due to the high-pass filtering effect, while the reflectance, after this procedure should still be very close to the original reflectance. The steps for Homomorphic Filtering algorithm are depicted in Fig. 3.
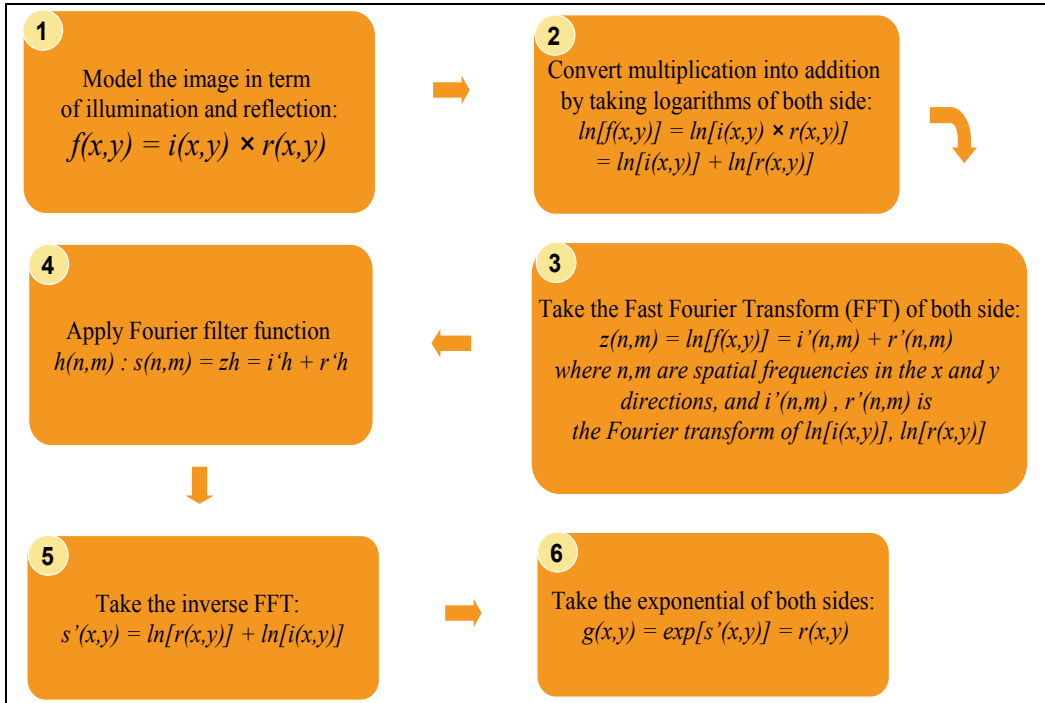

Fig. 3. Homomorphic Filtering algorithm

3.2 Feature extraction

Face recognition is a high dimensional pattern recognition problem which requires large computation time and memory storage. The purpose of feature extraction is to extract the feature vectors or information which represents the face to reduce computation time and memory storage. Nowadays, face recognition research has witnessed a growing interest based on techniques that capitalized, and apply algebraic and statistical tools for extraction and analysis of the underlying manifold. Face images are represented as a high dimensional pixel arrays that belongs to a manifold of intrinsically low dimension. Computer analysis of gray-scale face images deals with visual signal or light reflected off the surface of a face that is registered by a digital sensor as an array of pixel values. The pixel array is represented as a point or vector in an *m-by-n* dimensional image space after image preprocessing which involved normalization and rescaling to a fixed *m-by-n* size. Since faces are similar in

appearance and contain significant statistical regularities, the intrinsic dimensionality of the face space is much lower than the dimensionality of the input face image. For that reason, dimensionality reduction techniques based on linear subspace feature extraction, PCA is used to reduce the dimensionality of the input face space.

PCA is based on the information theory approach which is a dimensionality reduction technique based on extraction of the desired number of principal components of the multi-dimensional data. It is related to the Karhunen-Loeve Transform, which is derived in the signal processing context as the orthogonal transform. It is a statistical and computational method used to identify the linear directions in which a set of vectors are best represented in a least-square sense, to reduce multi-dimensional dataset into two dimensions for analysis by performing a covariance analysis between factors, and to identify new meaningful underlying variables or principal component.

PCA extracts the relevant information in a face image and encodes it as efficient as possible. It identifies the subspace of the image space spanned by the training face image data and de-correlates the pixel values. This involves the computation of the eigenvalue decomposition of a dataset, after mean centering the data for each attribute that transforms a number of correlated variables into a (smaller) number of uncorrelated variables called principal components. The principal components are obtained by projecting the multivariate data vectors on the space spanned by the eigenvectors. The first principal component is the linear combination of the original dimensions that has the maximum variance; the *n*-th principal component is the linear combination with the highest variance, subject to being orthogonal to the *n-1* of the first principal component (Shakhnarovich and Moghaddam, 2004). The sum of the eigenvalues equals the trace of the square matrix and the maximum number of eigenvectors equals the number of rows (or columns) of the matrix.

The basis vectors constructed by PCA have the same dimension as the input face images. The results of a PCA method are discussed in terms of scores and loadings. The classical representation of a face image is obtained by projecting it to the coordinate system defined by the principal components. The projection of face images into the principal component subspace achieves information compression, de-correlation and dimensionality reduction to facilitate decision making. In mathematical terms, the principal components of the distribution of faces or the eigenvectors of the covariance matrix of the set of face images is sought by treating an image as a vector in a very high dimensional face space. The steps for the PCA algorithm are as follow:

Step 1: The normalized training image in the *N*-dimensional space is stored in a vector of size *N*. Let the normalized training face image set,

$$T = \{X_1, X_2, ..., X_N\} \text{ where } X = \{x_1, x_2, x_3, ..., x_n\}^T \tag{1}$$

Step 2: Create Eigenspace

    a. Center data: Each of the normalized training face images is mean centered. This is done by subtracting the mean face image from each of the normalized training images. The mean image is represented as a column vector where each scalar is the mean of all corresponding pixels of the training images,

$$\overline{X_i} = X_i - \overline{X} \tag{2}$$

where the average of the training face image set is defined as:

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{3}$$

b. Create data matrix: Once the training face images are centered, the next process is to create the eigenspace which is the reduced vectors of the mean normalized training face images. The training images are combined into a data matrix of size N by P, where P is the number of training images and each column is a single image.

$$\overline{X} = \{\overline{X_1}, \overline{X_2}, \ ... \ ,\overline{X_P}\} \tag{4}$$

c. Create covariant matrix: The column vectors are combined into a data matrix which is multiplied by its transpose to create a covariance matrix. The covariance is defined as:

$$\Omega = \overline{X}.\overline{X}^T \tag{5}$$

d. Compute the eigenvalues and eigenvectors: The eigenvalues and corresponding eigenvectors are computed for the covariance matrix using Jacobian transformation,

$$\Omega V = \Lambda V \tag{6}$$

where V is the set of eigenvectors associated with the eigenvalues Λ.

e. Order eigenvectors: Order the eigenvectors $V_i \in V$ according to their corresponding eigenvalues $\lambda_i \in \Lambda$ from high to low with non-zero eigenvalues. This matrix of eigenvectors is the eigenspace V, where each column of V is an eigenvector. The principal components are the eigenspace V.

$$V = \{V_1, V_2, \ ... \ ,V_P\} \tag{7}$$

Step 3: Project training images

Each of the centered training images ($\overline{X}_i$) is projected into the eigenspace. The projected training images (T_P) are computed based on the dot product of the centered training images with each of the ordered eigenvectors denoted as,

$$T_P = V^T \overline{X}_i \tag{8}$$

The new vectors of the projected images are the feature vectors of the training face images. Let $T_P = \{X_{p1}, X_{p2}, ..., X_{pn}\}$ as feature vectors from the projection of the training set onto the principal components. The feature vector is defined as $X_p = \{p_1, p_2, p_3, ..., p_m\}^T$.

Step 4: Project testing image

The vector of the testing face image (Y) is initially mean centered by subtracting the mean image.

$$\overline{Y} = Y - \overline{X} \tag{9}$$

The feature vector of the testing image (Y_P) is obtained by projecting the vector of the mean testing face image ($\overline{Y}$) into the eigenspace or the principal components,

$$Y_P = V^T \overline{Y} \tag{10}$$

3.3 Face classification

The purpose of the classification stage is to map the feature space of a test data to a discrete set of label data that serves as template. For the proposed face recognition systems, the method used for classification is based on ANN. The ANN paradigm used is based on the Multi-layer Perceptrons (MLP) neural network. MLP is the most widely used ANN model today, which has also been extensively analyzed and for which many learning algorithms has been developed. MLP applies the back-propagation learning algorithm. The back propagation learning algorithm consists of forward pass and backward pass. The parameters for the back-propagation learning algorithm includes the number of hidden nodes, learning rate, momentum coefficient, number of training cycles, size of training subset, and size of test subsets. The back-propagation training algorithm requires a good selection of values for these parameters where they should not be set too high (large) or too low (small), and thus should be optimized or carefully selected commonly through trial and error. Training a network by back-propagation involves three stages: the feed-forward of the input training pattern, the back-propagation of the associated error, and adjustment of the weights. Back-propagation is a training process where the input data is repeatedly presented to the neural network. With each presentation the output of the network is compared to the desired output and an error is computed. The error is then fed back to the network and is used to adjust the weights such that the error decreases with each of the iteration and the neural model gets closer and closer to producing the desired output. The whole process of the back propagation algorithm is depicted in Fig. 4.

4. Experimental results

The performance of the proposed face recognition systems using photometric normalization, linear subspace feature extraction, and Artificial Neural Network (ANN) classification are evaluated using two (2) face datasets, which are AT&T face dataset, and local face dataset. In addition to the proposed face recognition system, we also implement the conventional face recognition systems using classifiers such as similarity distance techniques based on Euclidean Distance (ED), Normalized Correlation (NC), and Bayesian classifier as baselines for experimental comparison.

In these experiments, a comparison of face verification performance, namely FRR and FAR using different set of frontal face images, namely the original cropped face images, and face images which have undergone the photometric normalization techniques, Histogram Equalization, Homomorphic Filtering, combination of Histogram Equalization and Homomorphic Filtering, and combination of Homomorphic Filtering and Histogram Equalization. These sets of face images are evaluated using the proposed face recognition systems based on hybrid of PCA feature extraction and ANN classification (PCA+ANN). The experiments are conducted based on five (5) types of photometric normalization techniques which are:
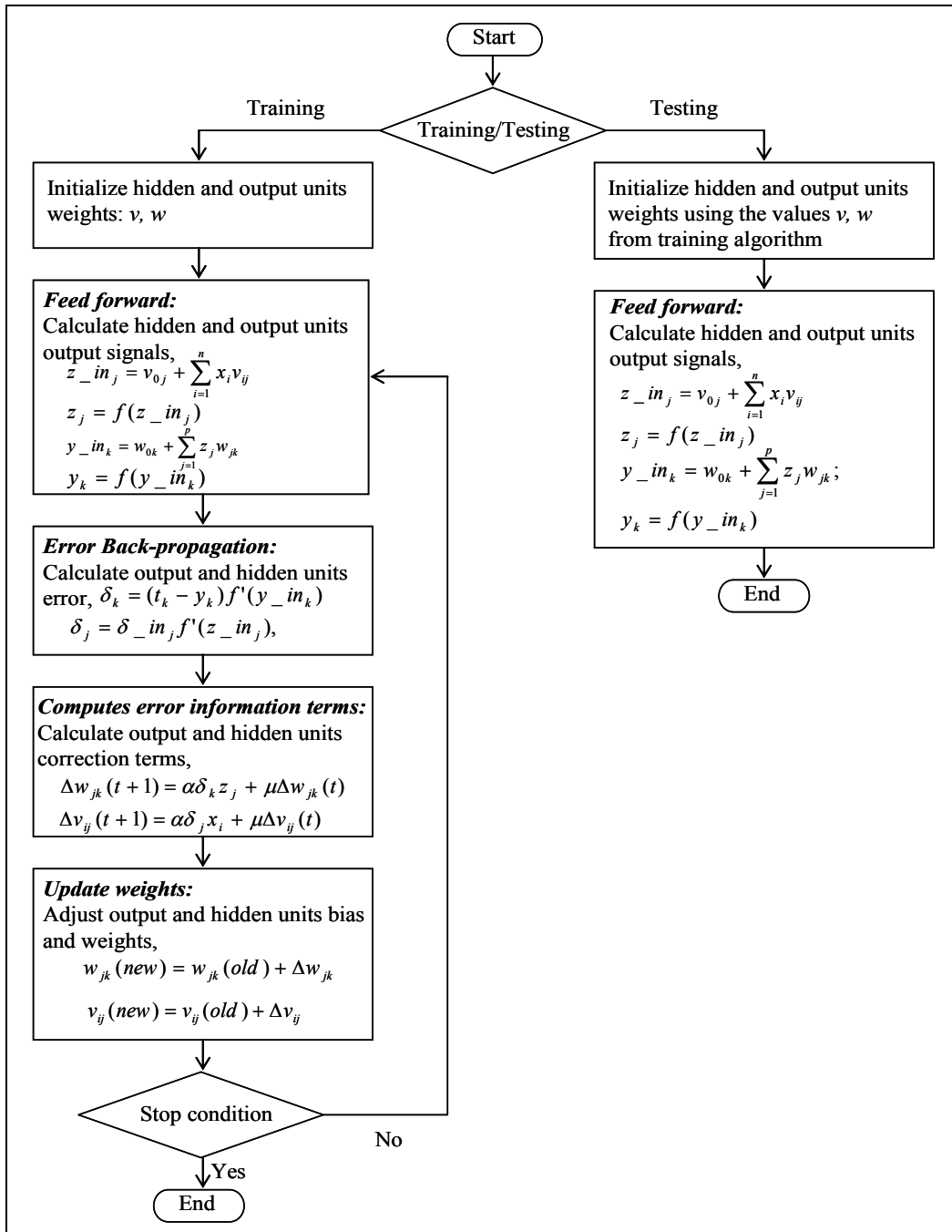
Fig. 4. The process flow of the back propagation algorithm

a) Type 1: without photometric normalization preprocessing
b) Type 2: photometric normalization preprocessing using Histogram Equalization
c) Type 3: photometric normalization preprocessing using Homomorphic Filtering

d) Type 4: photometric normalization preprocessing using combination of Histogram Equalization and Homomorphic Filtering

e) Type 5: photometric normalization preprocessing using combination of Homomorphic Filtering and Histogram Equalization.

Sample of the face images which have been preprocessed using Type 1, Type 2, Type 3, Type 4, and Type 5 photometric normalization preprocessing techniques are shown in Fig. 5, Fig. 6, Fig. 7, Fig. 8, and Fig. 9, respectively.
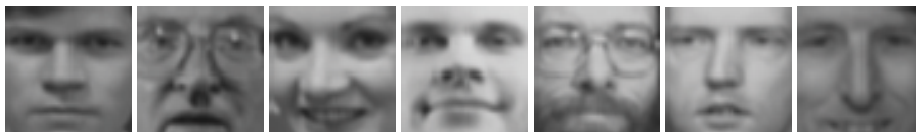

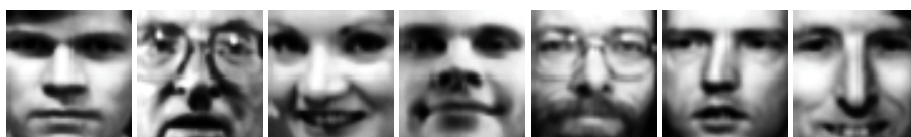Fig. 5. Sample of images based on Type 1 photometric normalization

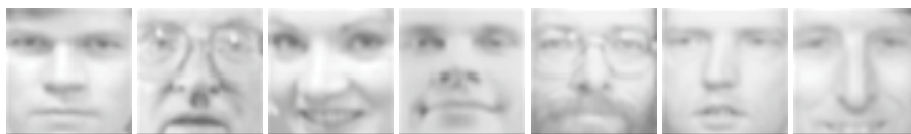Fig. 6. Sample of images based on Type 2 photometric normalization

Fig. 7. Sample of images based on Type 3 photometric normalization
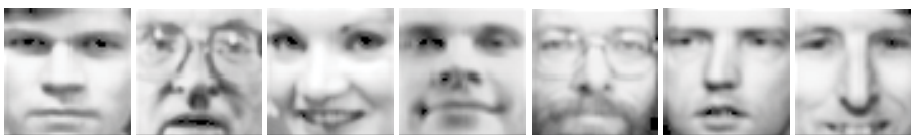

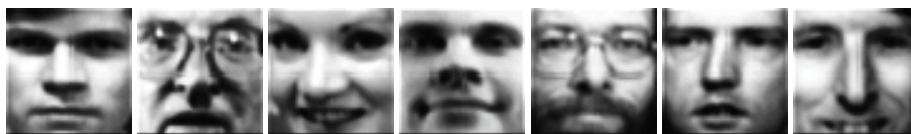Fig. 8. Sample of images based on Type 4 photometric normalization

Fig. 9. Sample of images based on Type 5 photometric normalization

The proposed face recognition system main decision making tool investigated is based on ANN classification. The parameters used for ANN classification are number of hidden neurons, learning rate, and momentum constant. For the experiments, the configuration of the optimal value for the number of hidden neurons is 100, for learning rate is 0.2, and for momentum constant is 0.7.

The experiments were conducted using the prescribed types of photometric normalization face image sets based on AT&T and local face datasets. For both of these datasets, five (5) images per subject were chosen as training images and the remaining images of each subject are used as the testing images. There were two hundreds (200) training images from AT&T and one hundred (100) training images from the local face datasets, respectively. Meanwhile, there were two hundreds (200) testing images from AT&T, and three hundreds

(300) test images from the local face dataset, respectively. The experiments were conducted based on the proposed face recognition system, PCA+ANN, and compared with the conventional face recognition systems, PCAA+ED, PCA+NC and Bayesian PCA.

The face verification experiments were performed with different photometric normalization techniques (without normalization, Histogram Equalization, Homomorphic Filtering, combination of Histogram Equalization and Homomorphic Filtering, combination of Homomorphic Filtering and Histogram Equalization), linear subspaces feature extraction based on PCA, and decision rule or classification based on ANN classifier which was compared with Euclidean Distance, Normalized Correlation, and Bayesian classifiers. The results using local and AT&T face datasets are summarized in Table 5.1 and Table 5.2, respectively. For local face dataset, the best verification performance was achieved using ANN classifier and Homomorphic Filtering with Half Total Error Rate (HTER) of 2.58%, and the worst performance of 21.15% was resulted using Bayesian classifier and Histogram Equalization. Meanwhile, the best verification performance for AT&T face dataset was achieved using ANN classifier, and combination of Histogram Equalization and Homomorphic Filtering with HTER of 5.02%, and the worst performance of 20.78% was resulted using ED classifier and Homomorphic Filtering.

The results presented in Table 5.1 and Table 5.2, show the system performance evaluation based on the proposed and conventional face recognition systems, namely PCA+ANN, PCA+ED, PCA+NC, and Bayesian PCA. In both AT&T face dataset and local face dataset experiments, it is interesting to note that the best HTER was achieved using PCA feature extraction and ANN classification, but using different photometric normalization technique. The differences in the test set performance are indicative of the different generalization capabilities of the respective methods. When the representation space already captured and emphasized, the discriminatory information content as in the case of PCA bases, ANN was superior to the simple Euclidean distance or correlation decision rules. ANN also shows a superior capability to cope with illumination changes, provided these were adequately represented in the training data. This was the main reason for the large difference between the observed performance of the Euclidean Distance, Normalized Correlation and Bayesian classification methods used as a benchmark and ANN.

The performance of the face recognition systems that use two-dimensional (2D) images is dependent on the consistent conditions such as illumination, pose and facial expression. The experimental results demonstrate that the proposed face recognition system improves the verification performance in the presence of illumination variations along with pose changes. One of the reasons for low verification performance is that the current optimization process is still subject to local minimum. Furthermore, the decision theoretic problems are best handled by classification methods based on an ANN classifier that yield the required decision functions directly via training rather than by methods based on distance metrics such as Euclidean distance, Normalized Correlation and Bayesian classifiers which make assumptions regarding the underlying probability density functions or other probabilistic information about pattern classes under consideration. The study proved that the enhancement of face recognition systems based on photometric normalization, linear subspace feature extraction, and ANN classification is able to improve the verification performance, and can be recommended for identity verification system.

Photometric Normalization	Feature Extractor	Classifier	Threshold	FAR	FRR	HTER
Without normalization	PCA	ED	6.10	7.32	7.41	7.37
		NC	0.54	14.44	16.56	16.00
		Bayesian	1.22	14.88	16.19	16.04
		ANN	0.37	6.62	6.56	6.59
Histogram Equalization		ED	9.63	12.96	12.96	12.96
		NC	0.23	6.21	6.93	6.07
		Bayesian	1.15	21.19	21.11	21.15
		ANN	0.20	4.42	4.44	4.43
Homomorphic Filtering		ED	4.91	6.58	6.30	6.44
		NC	0.42	7.80	7.41	7.61
		Bayesian	1.22	16.51	16.93	16.72
		ANN	0.43	2.57	2.59	**2.58**
Histogram Equalization + Homomorphic Filtering		ED	6.85	11.87	11.85	11.86
		NC	0.23	6.75	6.30	6.03
		Bayesian	0.92	20.70	20.74	20.72
		ANN	0.17	6.99	7.41	7.20
Homomorphic Filtering + Histogram Equalization		ED	9.67	12.57	12.59	12.58
		NC	0.24	6.69	6.56	6.63
		Bayesian	1.18	19.98	20.37	20.18
		ANN	0.17	3.79	3.70	3.75

Table 5.1. Verification performance as function of photometric normalization, PCA feature extractor, and classifiers based on local face dataset

5. Conclusions

The chapter has presented an enhancement of face recognition using photometric normalization, linear subspace feature extraction, and Artificial Neural Network (ANN) classification to improve the verification performance. The proposed face recognition systems based on the enhancement of FRS using photometric normalization, linear subspace feature extraction, and ANN classification has improved the performance of the proposed FRS. The proposed FRS using the combination of photometric normalization based on Homomorphic Filtering, feature extraction based on PCA, and ANN classification clearly outperform comparable conventional face recognition systems.

Based on the experimental results, the proposed face recognition using photometric normalization based on Homomorphic Filtering, feature extraction based on PCA, and ANN classification produces the best verification performance rate for local face dataset by yielding the lowest verification performance rate, HTER of 2.58%. On the other hand, the proposed face recognition using photometric normalization based on the combination of Histogram Equalization and Homomorphic Filtering, feature extraction based on PCA, and ANN classification produces the best verification performance rate for AT&T face dataset by yielding the lowest verification performance rate, HTER of 5.02%. Furthermore, the ANN classification based on Multilayer Perceptrons proved to be superior compared to that of distance metric techniques such as Euclidean distance, Normalized Correlation and Bayesian classifier. The reason for better verification performance using ANN classification

is because the classifier is trained through supervised learning known as error back-propagation algorithm. In supervised learning, a machine chooses the best function that relates between the inputs and outputs. This function is judged by its ability to generalize on new inputs which were not given in the training data. Therefore, the experimental results point out the overall robustness of the proposed face recognition system in comparison with the conventional face recognition systems. Thus, it proves the feasibility of using photometric normalization in the early stage of a face recognition system, which gave much impact to the verification performance.

Photometric Normalization	Feature Extractor	Classifier	Threshold	FAR	FRR	HTER
Without normalization		ED	6.30	13.56	13.50	13.53
		NC	0.30	11.42	12.50	11.96
		Bayesian	1.05	16.65	16.00	16.83
		ANN	0.11	6.67	6.50	6.59
Histogram Equalization		ED	12.28	16.29	16.50	16.40
		NC	0.24	7.53	7.50	7.52
		Bayesian	1.10	17.25	17.00	17.13
		ANN	0.12	6.47	6.50	6.49
Homomorphic Filtering	PCA	ED	4.19	20.56	21.00	20.78
		NC	0.25	12.50	11.50	12.00
		Bayesian	0.89	17.96	19.00	18.48
		ANN	0.07	9.23	9.50	9.37
Histogram Equalization + Homomorphic Filtering		ED	16.71	9.96	10.00	9.98
		NC	0.32	8.32	7.50	7.91
		Bayesian	1.12	18.02	18.00	18.01
		ANN	0.17	5.04	5.00	**5.02**
Homomorphic Filtering + Histogram Equalization		ED	11.89	14.99	16.00	16.00
		NC	0.25	6.94	7.50	7.22
		Bayesian	1.11	17.46	17.50	17.48
		ANN	0.11	6.41	6.50	6.46

Table 5.2. Verification performance as function of photometric normalization, PCA feature extractor, and classifiers based on AT&T face dataset

6. References

Er, M. J., Wu, J., Lu, S. & Toh, H. L. (2002). "Face recognition with radial basis function (RBF) neural networks". *IEEE Trans. Neural Networks*. Vol.13:697–710.

Kirby,M. & Sirovich, L. (1991). "Application of the Karhunen-Loève procedure for the characterization of human faces". *IEEE Trans. Pattern Anal. Mach. Intell*. Vol.12, No.1:103-108.

Lades M., Vorbruggen, J. C., Buhmann, J., Lange, J., Von der Malsburg , C., Wurtz, R. P. & Konen, W. (1993). "Distortion Invariant Object Recognition in the Dynamic Link Architecture", *IEEE Transactions on Computers*.Vol.42, March 1993: 300- 310.

Lawrence, S., Giles, C. L., Tsoi, A. C. & Back, A. D. (1997). "Face Recognition: A Convolutional Neural Networks Approach". *IEEE Trans. on Neural Networks*, Special Issue on Neural Networks and Pattern Recognition. Vol. 8, No. 1:98-113.

Lee, S. Z. & Lu, J. (1998). "Generalizing Capacity of Face Database for Face Recognition". *IEEE*: 402-406.

Lee, S. J., Yung, S. B., Kwon, J. W. & Hong, S. H. (1999). "Face Detection and Recognition Using PCA". *IEEE TENCON*: 84-87.

Li, S.Z. & Jain, A.K. (2005). *Handbook of Face Recognition*. Springer.

Martinez, A.M. & Kak, A.C. (2001). PCA versus LDA. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vo.l. 23, No. 2: 228-233.

Nazeer, S.A.; Omar, N. & Khalid, M. (2007). Face Recognition System using Artificial Neural Networks Approach. *International Conference on Signal Processing, Communications and Networking (ICSCN '07)*, 22-24 Feb. 2007: 420-425.

Nazeer, Shahrin Azuan; Omar, Nazaruddin; Khalid, Marzuki & Yusof, Rubiyah (2007). Face Verification and Recognition based on Learning Algorithm. *The 4th International Symposium on Management Engineering (ISME 2007)*, 10-12 March, 2007, Kitakyushu.

Shakhnarovich, G. & Moghaddam, B. (2004). Face Recognition in Subspaces. *In Handbook of Face Recognition. Eds. Stan Z. Li and Anil K. Jain.* Springer-Verlag.

Sirovich, L. & Kirby, M. (1987)."Low-dimensional procedure for the characterization of human face". *J. Opt. Soc. Am. A* , Vol. 4, No. 3:519-524.

Temdee, P., Khawparisuth, D. & Chamnongthai, K. (1999). Face Recognition by Using Fractal Encoding and Backpropagation Neural Network, *Fifth International Symposium on Signal Processing and its Applications*(ISSPA '99),159-161 , August 1991, Brisbane, Australia.

Thomaz, C. E., Feitosa, R. Q. & Veiga, A. (1998). "Design of Radial Basis Function Network as Classifier in Face Recognition Using Eigenfaces", *IEEE*: 118-123.

Tolba, A.S. & Abu-Rezq, A.N. (2000). "Combined Classifiers for Invariant Face Recognition". *Pattern Analysis and Applications. Springer London.* Vol. 3, No. 4: 289-302.

Turk, M. & Pentland, A. (1991)."Eigenfaces for Recognition". *Journal of Cognitive Neurosicence.* Vol. 3, No. 1:71-86.

Zhao, W.; Chellappa, R.; Rosenfeld, A.; Phillips, P.J. (2003). Face Recognition: A Literature Survey. *ACM Computing Surveys*, No. 35: 399-458, Dec 2003.

Online Incremental Face Recognition System Using Eigenface Feature and Neural Classifier

Seiichi Ozawa[1], Shigeo Abe[1], Shaoning Pang[2] and Nikola Kasabov[2]
[1]Graduate School of Engineering, Kobe University,
[2]Knowledge Engineering & Discover Research Institute, Auckland University of Technology
[1]Japan,
[2]New Zealand

1. Introduction

Understanding how people process and recognize faces has been a challenging problem in the field of object recognition for a long time. Many approaches have been proposed to simulate the human process, in which various adaptive mechanisms are introduced such as neural networks, genetic algorithms, and support vector machines (Jain et al., 1999). However, an ultimate solution for this is still being pursued. One of the difficulties in the face recognition tasks is to enhance the robustness over the spatial and temporal variations of human faces. That is, even for the same person, captured images of human faces have full of variety due to lighting conditions, emotional expression, wearing glasses, make-up, and so forth. And the face features could be changed slowly and sometimes drastically over time due to some temporal factors such as growth, aging, and health conditions.

When building a face recognition system, taking all the above variations into consideration in advance is unrealistic and maybe impossible. A remedy for this is to make a recognition system evolve so as to make up its misclassification on its own. In order to construct such an adaptive face recognition system, so-called *incremental learning* should be embedded into the system because it enables the system to conduct learning and classification on an ongoing basis. One challenging problem for this type of learning is to resolve so-called "plasticity and stability dilemma" (Carpenter & Grossberg, 1988). Thus, a system is required to improve its performance without deteriorating classification accuracy for previously trained face images.

On the other hand, feature extraction plays an essential role in pattern recognition because the extraction of appropriate features results in high generalization performance and fast learning. In this sense, incremental learning should be considered not only for a classifier but also for the feature extraction part. As far as we know, however, many incremental learning algorithms are aiming for classifiers. As for the incremental learning for feature extraction, Incremental Principal Component Analysis (IPCA) (e.g., Oja & Karhunen, 1985; Sanger, 1989; Weng et al., 2003; Zhao et al., 2006) and Incremental Linear Discriminant Analysis (Pang et al., 2005; Weng & Hwang, 2007) have been proposed so far. Hall and Martin (1998) proposed a method to update eigen-features (e.g., eigen-faces) incrementally based on eigenvalue decomposition. Ozawa et al. (2004) extended this IPCA algorithm such that an eigen-axis was augmented based on the accumulation ratio to control the dimensionality of an eigenspace easily.

Recently, a prototype face recognition system was developed by the authors (Ozawa et al, 2005) based on a new learning scheme in which a classifier and the feature extraction part were simultaneously learned incrementally. In this system, IPCA was adopted as an online feature extraction algorithm, and Resource Allocating Network with Long-Term Memory (RAN-LTM) (Kobayashi et al., 2001) was adopted as a classifier model. It was verified that the classification accuracy of the above classification system was improved constantly even if a small set of training samples were provided at a starting point. To accelerate learning of IPCA, we also proposed an extended algorithm called *Chunk IPCA* (Ozawa et al., 2008) in which an eigenspace is updated for a chunk of given training examples by solving a single intermediate eigenvalue problem.

The aim of this chapter is to demonstrate the followings:

1. how the feature extraction part is evolved by IPCA and Chunk IPCA,
2. how both feature extraction part and classifier are learned incrementally on an ongoing basis,
3. how an adaptive face recognition system is constructed and how it is effective.

This chapter is organized as follows. In Section 2, IPCA is first reviewed, and then Section 3 presents the detailed algorithm of Chunk IPCA. Section 4 explains two neural classifier models: Resource Allocating Network (RAN) and its variant model called RAN-LTM. In Section 5, an online incremental face recognition system and its information processing are described in detail, and we also explain how to reconstruct RAN-LTM when an eigenspace model is dynamically updated by Chunk IPCA. In Section 6, the effectiveness of incremental learning in face recognition systems is discussed for a self-compiled face image database. Section 7 gives conclusions of this chapter.

2. Incremental Principal Component Analysis (IPCA)

2.1 Learning assumptions and outline of IPCA algorithm

Assume that N training samples $\mathbf{x}^{(i)} \in R^n$ $(i = 1, \ldots, N)$ are initially provided to a system and an eigenspace model $\Omega = (\bar{\mathbf{x}}, \mathbf{U}_k, \mathbf{\Lambda}_k, N)$ is obtained by applying Principal Component Analysis (PCA) to the training samples. In the eigenspace model Ω, $\bar{\mathbf{x}}$ is a mean vector of $\mathbf{x}^{(i)}$ $(i = 1, \ldots, N)$, $\mathbf{U}_k$ is an $n \times k$ matrix whose column vectors correspond to eigenvectors, and $\mathbf{\Lambda}_k = \text{diag}\{\lambda_1, \ldots, \lambda_k\}$ is a $k \times k$ matrix whose diagonal elements are non-zero eigenvalues. Here, k is the number of eigen-axes spanning the eigenspace (i.e., eigenspace dimensionality) and the value of k is determined based on a certain criterion (e.g., accumulation ratio). After calculating Ω, the system keeps the information on Ω and all the training samples are thrown away.

Now assume that the $(N+1)$th training sample $\mathbf{x}^{(N+1)} = \mathbf{y} \in R^n$ is given. The addition of this new sample results in the changes in the mean vector and the covariance matrix; therefore, the eigenspace model $\Omega = (\bar{\mathbf{x}}, \mathbf{U}_k, \mathbf{\Lambda}_k, N)$ should be updated. Let us define the new eigenspace model by $\Omega' = (\bar{\mathbf{x}}', \mathbf{U}'_{k'}, \mathbf{\Lambda}'_{k'}, N+1)$. Note that the eigenspace dimensions might be increased from k to $k+1$; thus, k' in Ω' is either k or $k+1$. Intuitively, if almost all energy of $\mathbf{y}$ is included in the current eigenspace spanned by the eigenvectors $\mathbf{U}'_{k'}$, there is no need to increase an eigen-axis. However, if $\mathbf{y}$ includes certain energy in the complementary eigenspace, the dimensional augmentation is inevitable; otherwise crucial information on $\mathbf{y}$ might be lost. Regardless of the necessity in eigenspace augmentation, the

eigen-axes should be rotated to adapt to the variation in the data distribution. In summary, there are three main operations in IPCA: (1) mean vector update, (2) eigenspace augmentation, and (3) eigenspace rotation. The first operation is easily carried out without past training examples based on the following equation:

$$\overline{\mathbf{x}}' = \frac{1}{N+1}\left(N\overline{\mathbf{x}} + \mathbf{y}\right) \ \in R^n.$$

(1)

Hence, the following subsections give the explanation only for the last two operations.

2.2 Eigenspace augmentation
There have been proposed two criteria for judging eigenspace augmentation. One is the norm of a residue vector defined by

$$\mathbf{h} = (\mathbf{y} - \overline{\mathbf{x}}) - \mathbf{U}_k^T \mathbf{g} \ \ \text{where} \ \ \mathbf{g} = \mathbf{U}_k^T (\mathbf{y} - \overline{\mathbf{x}}).$$

(2)

Here, T means the transposition of vectors and matrices. The other is the accumulation ratio whose definition and incremental calculation are shown as follows:

$$A(\mathbf{U}_k) = \frac{\sum_{i=1}^{k}\lambda_i}{\sum_{j=1}^{n}\lambda_j} = \frac{(N+1)\sum_{i=1}^{k}\lambda_i + \left\|\mathbf{U}_k^T(\mathbf{y}-\overline{\mathbf{x}})\right\|^2}{(N+1)\sum_{j=1}^{n}\lambda_j + \left\|\mathbf{y}-\overline{\mathbf{x}}\right\|^2}$$

(3)

where λ_i is the ith largest eigenvalues, n and k mean the dimensionality of the input space and that of the current eigenspace, respectively. The former criterion in Eq. (2) was adopted in the original IPCA (Hall & Martin, 1998), and the latter in Eq. (3) was used in the modified IPCA proposed by the authors (Ozawa et al., 2004). Based on these criteria, the condition of increasing an eigen-axis $\hat{\mathbf{h}}$ is represented by:

$$[\text{Residue Vector Norm}] \ \ \ \hat{\mathbf{h}} = \begin{cases} \mathbf{h}/\|\mathbf{h}\| & \text{if } \|\mathbf{h}\| > \eta \\ 0 & \text{otherwise} \end{cases}$$

(4)

$$[\text{Accumulation Ratio}] \ \ \ \ \hat{\mathbf{h}} = \begin{cases} \mathbf{h}/\|\mathbf{h}\| & \text{if } A(\mathbf{U}_k) < \theta \\ 0 & \text{otherwise} \end{cases}$$

(5)

where η (in the original IPCA, η is set to zero) and θ are positive constants. Note that setting a too large threshold η or too small θ would cause serious approximation errors for eigenspace models. Hence, it is important to set proper values to η in Eq. (4) and θ in Eq. (5). In general, finding a proper threshold η is not easy unless input data are appropriately normalized within a certain range. On the other hand, since the accumulation ratio is defined by the ratio of input energy in an eigenspace over the original input space, the value of θ is restricted between 0 and 1. Therefore, it would be easier for θ to get an optimal value by applying the cross-validation method. The detailed algorithm of finding θ in incremental learning settings is described in Ozawa et al. (2008).

2.3 Eigenspace rotation

If the condition of Eq. (4) or (5) satisfies, the dimensions of the current eigenspace would be increased from k to $k+1$, and a new eigen-axis $\hat{\mathbf{h}}$ is added to the eigenvector matrix $\mathbf{U}_k$. Otherwise, the dimensionality remains the same. After this operation, the eigen-axes are rotated to adapt to the new data distribution. Assume that the rotation is given by a rotation matrix $\mathbf{R}$, then the eigenspace update is represented by the following equation:

$$1) \text{ If there is a new eigen-axis to be added, } \mathbf{U}'_{k+1} = [\mathbf{U}_k, \hat{\mathbf{h}}]\mathbf{R}, \tag{6}$$

$$2) \text{ otherwise, } \mathbf{U}'_k = \mathbf{U}_k\,\mathbf{R}. \tag{7}$$

It has been shown that $\mathbf{R}$ is obtained by solving the following intermediate eigenproblem (Hall & Martin, 1998):

1. If there is a new eigen-axis to be added,

$$\left\{ \frac{N}{N+1}\begin{bmatrix} \mathbf{\Lambda}_k & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \frac{N}{(N+1)^2}\begin{bmatrix} \mathbf{g}\mathbf{g}^T & \gamma\mathbf{g} \\ \gamma\mathbf{g}^T & \gamma^2 \end{bmatrix} \right\}\mathbf{R} = \mathbf{R}\mathbf{\Lambda}'_{k+1}, \tag{8}$$

2. otherwise,

$$\left\{ \frac{N}{N+1}\mathbf{\Lambda}_k + \frac{N}{(N+1)^2}\mathbf{g}\mathbf{g}^T \right\}\mathbf{R} = \mathbf{R}\mathbf{\Lambda}'_k. \tag{9}$$

Here, $\gamma = \hat{\mathbf{h}}^T(\mathbf{y}-\bar{\mathbf{x}})$ and $\mathbf{0}$ is a k-dimensional zero vector; $\mathbf{\Lambda}'_{k+1}$ and $\mathbf{\Lambda}'_k$ are the new eigenvalue matrices whose diagonal elements correspond to k and $k+1$ eigenvalues, respectively. Using the solution $\mathbf{R}$, the new eigenvector matrix $\mathbf{U}'_k$ or $\mathbf{U}'_{k+1}$ is calculated from Eq. (6) or (7).

3. Chunk Incremental Principal Component Analysis (Chunk IPCA)

3.1 Learning assumptions and outline of chunk IPCA algorithm

IPCA can be applied to one training sample at a time, and the intermediate eigenproblem in Eq. (8) or (9) must be solved for each sample even though a chunk of samples are provided to learn at a time. Obviously this is inefficient from a computational point of view, and the learning may get stuck in a deadlock if a large chunk of training samples is given to learn in a short term; that is, the next chunk of training samples could come before the learning is completed if it takes long time for updating an eigenspace.

To overcome this problem, the original IPCA is extended so that the eigenspace model Ω can be updated with a chunk of training samples in a single operation (Ozawa et al., 2008). This extended algorithm is called *Chunk IPCA*.

Assume again that N training samples $\mathbf{X} = \left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\right\} \in R^{n \times N}$ have been given so far and an eigenspace model $\Omega = (\bar{\mathbf{x}}, \mathbf{U}_k, \mathbf{\Lambda}_k, N)$ was obtained from these samples. Now, a chunk of L training samples $\mathbf{Y} = \left\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(L)}\right\} \in R^{n \times L}$ are presented to the system. Let the updated eigenspace model be $\Omega' = (\bar{\mathbf{x}}', \mathbf{U}'_{k'}, \mathbf{\Lambda}'_{k'}, N+L)$. The mean vector $\bar{\mathbf{x}}'$ in Ω' can be updated without the past training samples $\mathbf{X}$ as follows:

$$\overline{\mathbf{x}}' = \frac{1}{N+L}\left(N\,\overline{\mathbf{x}} + L\,\overline{\mathbf{y}}\right)\!\cdot \tag{10}$$

The problem is how to update $\mathbf{U}'_k$ and $\mathbf{\Lambda}'_k$ in Ω'.

As shown in the derivation of IPCA, the update of $\mathbf{U}'_k$ and $\mathbf{\Lambda}'_k$ is reduced to solving an intermediate eigenproblem, which is derived from an eigenvalue problem using a covariance matrix. Basically, the intermediate eigenproblem for Chunk IPCA can also be derived in the same way as shown in the derivation of IPCA (Hall & Martin, 1998). However, we should note that the dimensionality k' of the updated eigenspace could range from k to $k+L$ depending on the given chunk data $\mathbf{Y}$. To avoid constructing a redundant eigenspace, the smallest k' should be selected under the condition that the accumulation ratio is over a designated threshold. Thus, an additional operation to select a smallest set of eigen-axes is newly introduced into Chunk IPCA. Once the eigen-axes to be augmented are determined, all the eigen-axes should be rotated to adapt to the variation in the data distribution. This operation is basically the same as in IPCA.

In summary, there are three main operations in Chunk IPCA: (1) mean vector update, (2) eigenspace augmentation with the selection of a smallest set of eigen-axes, and (3) eigenspace rotation. The first operation is carried out by Eq. (10). The latter two operations are explained below.

3.2 Eigenspace augmentation

In Chunk IPCA, the number of eigen-axes to be augmented is determined by finding the minimum k such that $A(\mathbf{U}_k) \geq \theta$ holds where θ is a threshold between 0 and 1. To introduce this criterion, we need to modify the update equation of Eq. (3) such that the accumulation ratio can be updated incrementally for a chunk of L samples. In Chunk IPCA, we need to consider two types of accumulation ratios. One is the accumulation ratio for a k-dimensional eigenspace spanned by $\mathbf{U}'_k = \mathbf{U}_k\mathbf{R}$ where $\mathbf{R}$ is a rotation matrix which is calculated from the intermediate eigenvalue problem described later. The other is that for a $(k+l)$-dimensional augmented eigenspace spanned by $\mathbf{U}'_{k+l} = [\mathbf{U}_k, \mathbf{H}_l]\mathbf{R}$ where $\mathbf{H}_l$ is a set of l augmented eigen-axes. The former is used for checking if the current k-dimensional eigenspace should be augmented or not. The latter one is used for checking if further eigen-axes are needed for the $(k+l)$-dimensional augmented eigenspace.

The new accumulation ratio $A'(\mathbf{U}'_k)$ and $A'(\mathbf{U}'_{k+l})$ are calculated as follows (Ozawa et al., 2008):

$$A'(\mathbf{U}'_k) = \frac{\displaystyle\sum_{i=1}^{k}\lambda_i + \frac{L}{N+L}\left\|\overline{\mathbf{g}}\right\|^2 + \frac{1}{N}\sum_{j=1}^{L}\left\|\mathbf{g}''_j\right\|^2}{\displaystyle\sum_{i=1}^{n}\lambda_i + \frac{L}{N+L}\left\|\overline{\mathbf{\mu}}\right\|^2 + \frac{1}{N}\sum_{j=1}^{L}\left\|\mathbf{\mu}''_j\right\|^2} \tag{11}$$

$$A'(\mathbf{U}'_{k+l}) \approx \frac{\displaystyle\sum_{i=1}^{k}\lambda_i + \frac{L}{N+L}\left\|\frac{\overline{\mathbf{g}}}{\overline{\mathbf{\gamma}}}\right\|^2 + \frac{1}{N}\sum_{j=1}^{L}\left\|\frac{\mathbf{g}''_j}{\mathbf{\gamma}''_j}\right\|^2}{\displaystyle\sum_{i=1}^{n}\lambda_i + \frac{L}{N+L}\left\|\overline{\mathbf{\mu}}\right\|^2 + \frac{1}{N}\sum_{j=1}^{L}\left\|\mathbf{\mu}''_j\right\|^2} \tag{12}$$

where $\bar{\mathbf{g}} = \mathbf{U}_k^T \left(\bar{\mathbf{y}} - \bar{\mathbf{x}} \right),$ $\mathbf{g}_i'' = \mathbf{U}_k^T \left(\mathbf{y}^{(i)} - \bar{\mathbf{y}} \right),$ $\bar{\boldsymbol{\mu}} = \bar{\mathbf{x}} - \bar{\mathbf{y}},$ $\boldsymbol{\mu}_j'' = \mathbf{y}^{(j)} - \bar{\mathbf{y}},$ $\bar{\boldsymbol{\gamma}} = \mathbf{H}_l^T \left(\bar{\mathbf{y}} - \bar{\mathbf{x}} \right),$ and $\boldsymbol{\gamma}_i'' = \mathbf{H}_l^T \left(\mathbf{y}^{(i)} - \bar{\mathbf{y}} \right)$. To update $A'(\mathbf{U}_k')$, the summation of eigenvalues λ_i $(i = 1, \ldots, n)$ is required, and this summation can be held by accumulating the power of training samples (Ozawa et al., 2004). Hence, the individual eigenvalues λ_i $(i = k+1, \ldots, n)$ are not necessary for this update.

As seen from Eqs. (11) and (12), we need no past sample $\mathbf{x}^{(j)}$ and no rotation matrix $\mathbf{R}$ to update the accumulation ratio. Therefore, this accumulation ratio is updated with the following information: a chunk of given training samples $\mathbf{Y} = \left\{ \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(L)} \right\}$, the current eigenspace model $\Omega = (\bar{\mathbf{x}}, \mathbf{U}_k, \boldsymbol{\Lambda}_k, N)$, the summation of eigenvalues $\sum_{j=1}^n \lambda_i$, and a set of augmented eigen-axes $\mathbf{H}_l$ which are obtained through the procedure described next.

In IPCA, a new eigen-axis is obtained to be orthogonalized to the existing eigenvectors (i.e., column vectors of $\mathbf{U}_k$). A straightforward way to obtain new eigen-axes is to apply Gram-Schmidt orthogonalization to a chunk of given training samples (Hall et al., 2000). If the training samples are represented by $\widetilde{L}$ linearly independent vectors, the maximum number of augmented eigen-axes is $\widetilde{L}$. However, the subspace spanned by all of the $\widetilde{L}$ eigen-axes is redundant in general. Therefore, we should find a smallest set of eigen-axes without losing essential information on $\mathbf{Y}$.

The problem of finding an optimal set of eigen-axes is stated below.

Find the smallest set of eigen-axes $\mathbf{H}^* = \left\{ \mathbf{h}_1, \ldots, \mathbf{h}_{l^*} \right\}$ *for the current eigenspace model* $\Omega = (\bar{\mathbf{x}}, \mathbf{U}_k, \boldsymbol{\Lambda}_k, N)$ *without keeping the past training samples* $\mathbf{X}$ *such that the accumulation ratio* $A'(\mathbf{U}_{k+l^*}')$ *of all the given training samples* $\{\mathbf{X}, \mathbf{Y}\}$ *is larger than a threshold* θ.

Assume that we have a candidate set of augmented eigen-axes $\mathbf{H}_l = \left\{ \mathbf{h}_1, \ldots, \mathbf{h}_l \right\}$. Since the denominator of Eq. (12) is constant once the mean vector $\bar{\mathbf{y}}$ is calculated, the increment of the accumulation ratio from $A'(\mathbf{U}_k')$ to $A'(\mathbf{U}_{k+l}')$ is determined by the numerator terms. Thus, let us define the following difference $\Delta \widetilde{A}'(\mathbf{U}_{k+l}')$ of the numerator terms between $A'(\mathbf{U}_k')$ and $A'(\mathbf{U}_{k+l}')$:

$$\Delta \widetilde{A}'(\mathbf{U}_{k+l}') = \frac{L}{N+L} \left\| \mathbf{H}_l^T (\bar{\mathbf{x}} - \bar{\mathbf{y}}) \right\|^2 + \frac{1}{N} \sum_{j=1}^L \left\| \mathbf{H}_l^T (\mathbf{y}^{(j)} - \bar{\mathbf{y}}) \right\|^2 \stackrel{\text{def}}{=\!=} \sum_{i=1}^l \Delta \widetilde{A}_i' \tag{13}$$

where

$$\Delta \widetilde{A}_i' = \frac{L}{N+L} \left\{ \mathbf{h}_i^T (\bar{\mathbf{x}} - \bar{\mathbf{y}}) \right\}^2 + \frac{1}{N} \sum_{j=1}^L \left\{ \mathbf{h}_i^T (\mathbf{y}^{(j)} - \bar{\mathbf{y}}) \right\}^2. \tag{14}$$

Equation (13) means that the increments of the accumulation ratio is determined by the linear sum of $\Delta \widetilde{A}_i'$. Therefore, to find the smallest set of eigen-axes, first we find $\mathbf{h}_i$ with the largest $\Delta \widetilde{A}_i'$, and put it into the set of augmented eigen-axes $\mathbf{H}_l$ (i.e., $l = 1$ and $\mathbf{H}_l = \mathbf{h}_i$). Then, check if the accumulation ratio $A'(\mathbf{U}_{k+l}')$ in Eq. (12) becomes larger than the threshold θ. If not, select $\mathbf{h}_i$ with the second largest $\Delta \widetilde{A}_i'$, and the same procedure is repeated until $A'(\mathbf{U}_{k+l}') \geq \theta$ satisfies. This type of greedy algorithm makes the selection of eigen-axes very simple. The algorithm of the eigen-axis selection is summarized in **Algorithm 1**.

Algorithm 1: *Eigen-axis Selection*

Input: Eigenspace model $\Omega = (\overline{\mathbf{x}}, \mathbf{U}_k, \mathbf{\Lambda}_k, N)$, L training samples $\mathbf{Y} = \left\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(L)}\right\}$,

the summation of eigenvalues $\sum_{j=1}^{n} \lambda_i$, and threshold θ of accumulation ratio.

Calculate the mean vector $\overline{\mathbf{y}}$ in Eq. (10).

Calculate the accumulation ratio $A'(\mathbf{U}'_k)$ in Eq. (11).

If $A'(\mathbf{U}'_k) \geq \theta$ **then**

Terminate this algorithm.

End if

For $i = 1$ to L **do**

Obtain the following residue vectors $\mathbf{h}_i$ using the ith sample $\mathbf{y}^{(i)}$:

$$\mathbf{h}_i = \frac{\mathbf{r}_i}{\|\mathbf{r}_i\|} \text{ where } \mathbf{r}_i = (\mathbf{y}^{(i)} - \overline{\mathbf{x}}) - \mathbf{U}_k \mathbf{U}_k^T (\mathbf{y}^{(i)} - \overline{\mathbf{x}}).$$

End for

Define an index set Γ of $\mathbf{h}_i$, and initialize $\mathbf{H}$ and l.

Loop

Find the residue vector $\mathbf{h}_{i'}$ that gives the largest $\Delta \widetilde{A}'_i$ in Eq. (14): $i' = \arg\max_{i \in \Gamma} \Delta \widetilde{A}'_i$.

$\mathbf{H} \leftarrow [\mathbf{H}, \mathbf{h}_{i'}]$, $l \leftarrow l + 1$, and $\Gamma \leftarrow \Gamma - i'$.

Calculate the updated accumulation ratio $A'(\mathbf{U}'_{k+l})$ in Eq. (12).

If Γ is empty or $A'(\mathbf{U}'_{k+l}) > \theta$ **then**

Terminate this algorithm.

End if

End loop

Output: Augmented eigen-axes: $\mathbf{H}_l = \left\{\mathbf{h}_1, \ldots, \mathbf{h}_l\right\}$.

3.3 Eigenspace rotation

Next, let us derive the update equations for $\mathbf{U}_k$ and $\mathbf{\Lambda}_k$. Suppose that l eigen-axes are augmented when a chunk of L training samples $\mathbf{Y}$ is provided; that is, the eigenspace dimensions are increased by l. Let us denote the augmented eigen-axes as follows:

$$\mathbf{H}_l = \left\{\mathbf{h}_1, \ldots, \mathbf{h}_l\right\} \in R^{n \times l}, \quad 0 \leq l \leq L. \tag{15}$$

Then, the updated eigenvector matrix $\mathbf{U}'_{k+l}$ is represented by

$$\mathbf{U}'_{k+l} = [\mathbf{U}_k, \mathbf{H}_l] \mathbf{R} \tag{16}$$

where $\mathbf{R}$ is a rotation matrix. It has been shown that $\mathbf{R}$ is obtained by solving the following intermediate eigenproblem (Ozawa et al., 2008):

1. If there are new eigen-axes to be added (i.e., $l \neq 0$),

$$\left\{ \frac{N}{N+L} \begin{bmatrix} \mathbf{\Lambda}_k & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \frac{NL^2}{(N+L)^3} \begin{bmatrix} \overline{\mathbf{g}}\,\overline{\mathbf{g}}^T & \overline{\mathbf{g}}\,\overline{\mathbf{\gamma}}^T \\ \overline{\mathbf{\gamma}}\,\overline{\mathbf{g}}^T & \overline{\mathbf{\gamma}}\,\overline{\mathbf{\gamma}}^T \end{bmatrix} + \frac{N^2}{(N+L)^3} \sum_{i=1}^{L} \begin{bmatrix} \mathbf{g}'_i \mathbf{g}'^T_i & \mathbf{g}'_i \mathbf{\gamma}'^T_i \\ \mathbf{\gamma}'_i \mathbf{g}'^T_i & \mathbf{\gamma}'_i \mathbf{\gamma}'^T_i \end{bmatrix} \right.$$

$$\frac{L(L+2N)}{(N+L)^3} \sum_{i=1}^{L} \begin{bmatrix} \mathbf{g}_i'' \mathbf{g}_i''^T & \mathbf{g}_i'' \boldsymbol{\gamma}_i''^T \\ \boldsymbol{\gamma}_i'' \mathbf{g}_i''^T & \boldsymbol{\gamma}_i'' \boldsymbol{\gamma}_i''^T \end{bmatrix} \Bigg\} \mathbf{R} = \mathbf{R} \boldsymbol{\Lambda}_{k+l}', \tag{17}$$

2.   otherwise,

$$\left\{ \frac{N}{N+L} \boldsymbol{\Lambda}_k + \frac{NL^2}{(N+L)^3} \overline{\mathbf{g}}\overline{\mathbf{g}}^T + \frac{N^2}{(N+L)^3} \sum_{i=1}^{L} \mathbf{g}_i' \mathbf{g}_i'^T + \frac{L(L+2N)}{(N+L)^3} \sum_{i=1}^{L} \mathbf{g}_i'' \mathbf{g}_i''^T \right\} \mathbf{R} = \mathbf{R} \boldsymbol{\Lambda}_k'. \tag{18}$$

Here, $\mathbf{g}_i' = \mathbf{U}_k^T \left(\mathbf{y}^{(i)} - \overline{\mathbf{x}} \right)$ and $\boldsymbol{\gamma}_i' = \mathbf{H}_l^T \left(\mathbf{y}^{(i)} - \overline{\mathbf{x}} \right)$. As seen from Eqs. (17) and (18), the rotation matrix $\mathbf{R}$ and the eigenvalue matrix $\boldsymbol{\Lambda}_{k+l}'$ correspond to the eigenvectors and eigenvalues of the intermediate eigenproblem, respectively. Once $\mathbf{R}$ is obtained, the corresponding new eigenvector matrix $\mathbf{U}_{k+l}'$ is given by Eq. (16).

The overall algorithm of Chunk IPCA is summarized in **Algorithm 2**.

Algorithm 2: *Chunk IPCA*

Input: Eigenspace model $\Omega = (\overline{\mathbf{x}}, \mathbf{U}_k, \boldsymbol{\Lambda}_k, N)$ and L training samples $\mathbf{Y} = \left\{ \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(L)} \right\}$.

   Perform *Eigen-axis Selection* (**Algorithm 1**) to obtain augmented eigen-axes $\mathbf{H}_l$.

   Solve the intermediate eigenproblem in Eq. (17) or (18) to obtain a rotation matrix $\mathbf{R}$ and an updated eigenvalue matrix $\boldsymbol{\Lambda}_{k+l}'$.

   Obtain the updated the eigenvector matrix $\mathbf{U}_{k+l}'$ in Eq. (16).

   Update the mean vector $\overline{\mathbf{x}}'$ in Eq. (10).

Output: Updated eigenspace model $\Omega' = (\overline{\mathbf{x}}', \mathbf{U}_{k+l}', \boldsymbol{\Lambda}_{k+l}', N+L)$.

3.4 Training of initial eigenspace

Assume that a set of initial training samples $D_0 = \left\{ (\mathbf{x}^{(i)}, \mathbf{z}^{(i)}) \mid i = 1, \ldots, N \right\}$ is given before incremental learning gets started. To obtain an initial eigenspace model $\Omega = (\overline{\mathbf{x}}, \mathbf{U}_k, \boldsymbol{\Lambda}_k, N)$, the conventional PCA is applied to D_0 and the smallest dimensionality k of the eigenspace is determined such that the accumulation ratio is larger than θ. Since a proper θ is usually unknown and often depends on training data, the cross-validation technique can be applied to determining θ (Ozawa et al., 2008). However, for the sake of simplicity, let us assume here that a proper θ is given in advance. The algorithm of the initial training is shown in **Algorithm 3**.

Algorithm 3: *Training of Initial Eigenspace*

Input: Initial training set $D_0 = \left\{ (\mathbf{x}^{(i)}, z^{(i)}) \mid i = 1, \ldots, N \right\}$ and threshold θ.

   Calculate the mean vector $\overline{\mathbf{x}}$ of $\mathbf{x}^{(i)} \in D_0$.

   Apply PCA to D_0 and obtain the eigenvectors $\mathbf{U}_n = \left\{ \mathbf{u}_1, \ldots, \mathbf{u}_n \right\}$ whose eigenvalues $\boldsymbol{\Lambda}_n = \mathrm{diag}\{\lambda_1, \ldots, \lambda_n\}$ are sorted in decreasing order.

   Find the smallest k such that the following condition holds: $A(\mathbf{U}_k) \geq \theta$.

Output: Eigenspace model $\Omega = (\overline{\mathbf{x}}, \mathbf{U}_k, \boldsymbol{\Lambda}_k, N)$.

---

4. Incremental learning of classifier

4.1 Resource Allocating Network (RAN)

Resource Allocating Network (RAN) (Platt, 1991) is an extended version of RBF networks. When the training gets started, the number of hidden units is set to one; hence, RAN has simple approximation ability at first. As the training proceeds, the approximation ability of RAN is developed with the increase of training samples by allocating additional hidden units.

Figure 1 illustrates the structure of RAN. The output of hidden units $\mathbf{y} = \{y_1, \ldots, y_J\}^T$ is calculated based on the distance between an input $\mathbf{x} = \{x_1, \ldots, x_I\}^T$ and center vector of the jth hidden unit $\mathbf{c}_j = \{c_{j1}, \ldots, c_{jI}\}^T$:

$$y_j = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_j\|^2}{\sigma_j^2} \right) \qquad \text{for } j = 1, \ldots, J \tag{19}$$

where I and J are the numbers of input units and hidden units, respectively, and σ_j^2 is a variance of the jth radial basis. The network output $\mathbf{z} = \{z_1, \ldots, z_K\}^T$ is calculated as follows:

$$z_k = \sum_{k=1}^{K} w_{kj} y_j + \gamma_k \qquad \text{for } k = 1, \ldots, K \tag{20}$$

where K is the number of output units, w_{kj} is a connection weight from the jth hidden unit to the kth output unit, and γ_k is a bias of the kth output unit.

When a training sample $(\mathbf{x}, \mathbf{d})$ is given, the network output is calculated based on Eqs. (19) and (20), and the root mean square error $E = \|\mathbf{d} - \mathbf{z}\|$ between the output $\mathbf{z}$ and target $\mathbf{d}$ for the input $\mathbf{x}$ is evaluated. Depending on E, either of the following operations is carried out:
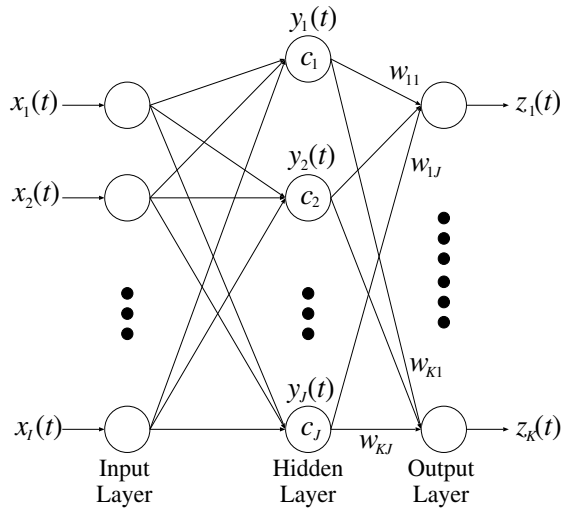
Fig. 1. Structure of Resource Allocating Network (RAN).

1. If E is larger than a positive constant ε and the distance between an input $\mathbf{x}$ and its nearest center vector $\mathbf{c}^*$ is larger than a positive value $\delta(t)$ (i.e., $E > \varepsilon$ and $\left\| \mathbf{x} - \mathbf{c}^* \right\| > \delta(t)$), a hidden unit is added (i.e., $J \leftarrow J + 1$). Then, the network parameters for the Jth hidden unit (center vector $\mathbf{c}_J$, connection weights $\mathbf{w}_J = \{ w_{1J}, \ldots, w_{KJ} \}$, and variance σ_j^2) are set to the following values: $\mathbf{c}_J = \mathbf{x}_p$, $\mathbf{w}_J = \mathbf{d} - \mathbf{z}$, and $\sigma_J = \kappa \left\| \mathbf{x} - \mathbf{c}^* \right\|$ where κ is a positive constant. $\delta(t)$ is decreased with time t as follows:

$$\delta(t) = \max \left[\delta_{\max} \exp \left(-\frac{t}{\tau} \right), \delta_{\min} \right] \tag{21}$$

where τ is a decay constant, $\delta_{\max}$ and $\delta_{\min}$ are maximum and minimum values of $\delta(t)$, respectively.

2. Otherwise, the network parameters are updated as follows:

$$w_{kj}^{NEW} = w_{kj}^{OLD} + \alpha \, e_k \, y_{pj} \tag{22}$$

$$c_{ji}^{NEW} = c_{ji}^{OLD} + \frac{\alpha}{\sigma_j^2} (x_{pi} - c_{ji}) y_{pj} \sum_{k=1}^{K} e_k \, w_{kj} \tag{23}$$

$$\gamma_k^{NEW} = \gamma_k^{OLD} + \alpha \, e_k \tag{24}$$

where $e_k = d_k - z_k$ and α is a positive learning ratio.

Although the approximation ability is developed by allocating hidden units, the interference cannot be suppressed completely only by this mechanism. In the next section, we present an extended model of RAN in which a mechanism of suppressing the interference is explicitly introduced.

4.2 Resource allocating network with long-term memory

RAN is a neural network with spatially localised basis functions; hence it is expected that the catastrophic interference (Carpenter & Grossberg, 1988) is alleviated to some extent. However, since no explicit mechanism of suppressing the interference is introduced, the insufficient suppression might cause serious unlearning over the long run.

To suppress unexpected forgetting in RAN, Resource Allocating Network with Long-Term Memory (RAN-LTM) (Kobayashi et al., 2001) has been proposed. Figure 2 shows the architecture of RAN-LTM which consists of two modules: RAN and an external memory called *Long-Term Memory* (LTM). Representative input-output pairs are extracted from the mapping function acquired in RAN and they are stored in LTM. These pairs are called *memory items* and some of them are retrieved from LTM to learn with training samples. In the learning algorithm, a memory item is created when a hidden unit is allocated; that is, an RBF center and the corresponding output are stored as a memory item.
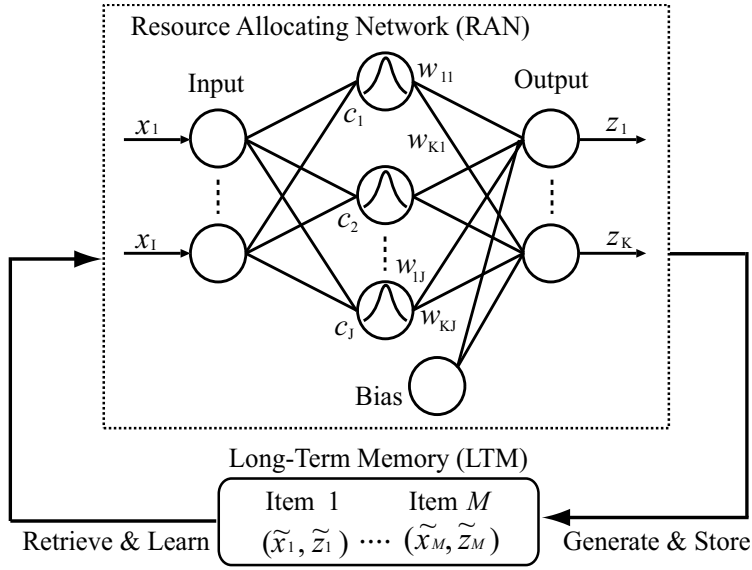
Fig. 2. Architecture of RAN-LTM.

The learning algorithm of RAN-LTM is divided into two phases: the allocation of hidden units (i.e., incremental selection of RBF centers) and the calculation of connection weights between hidden and output units. The procedure in the former phase is the same as that in the original RAN, except that memory items are created at the same time. Once hidden units are allocated, the centers are fixed afterwards. Therefore, the connection weights $\mathbf{W} = \left\{ w_{jk} \right\}$ are only parameters that are updated based on the output errors. To minimize the errors based on the least squares method, it is well known that the following linear equations should be solved (Haykin, 1999):

$$\boldsymbol{\Phi}\mathbf{W} = \mathbf{D} \tag{25}$$

where $\mathbf{D}$ is a matrix whose column vectors correspond to the target outputs and $\boldsymbol{\Phi}$ is a matrix of hidden outputs. Suppose that a new training sample $(\mathbf{x}, \mathbf{d})$ is given and M memory items $(\widetilde{\mathbf{x}}^{(m)}, \widetilde{\mathbf{z}}^{(m)})$ $(m = 1, \ldots, M)$ have already been created, then in the simplest version of RAN-LTM (Ozawa et al., 2005) the target matrix $\mathbf{D}$ are formed as follows: $\mathbf{D} = \left\{ \mathbf{d}, \widetilde{\mathbf{z}}^{(1)}, \ldots, \widetilde{\mathbf{z}}^{(M)} \right\}^T$. Furthermore, $\boldsymbol{\Phi} = \left\{ \phi_{ij} \right\}$ $(i = 1, \ldots, M+1)$ is calculated from the training sample and memory items as follows:

$$\phi_{1j} = \exp\left(-\frac{\left\| \mathbf{x} - \mathbf{c}_j \right\|^2}{\sigma_j^2} \right), \quad \phi_{i+1,j} = \exp\left(-\frac{\left\| \widetilde{\mathbf{x}}_i - \mathbf{c}_j \right\|^2}{\sigma_j^2} \right) \quad (j = 1, \ldots, J; i = 1, \ldots, M). \tag{26}$$

Singular Value Decomposition (SVD) can be used for solving $\mathbf{W}$ in Eq. (25). The learning algorithm of RAN-LTM is summarized in **Algorithm 4**.

Algorithm 4: *Learning of RAN-LTM*

Input: RAN-LTM and L training samples $(\mathbf{x}^{(p)}, \mathbf{d}^{(p)})$ $(p = 1, \ldots, L)$.

  For $p = 1$ to L **do**

    Find the nearest center $\mathbf{c}^*$ to the pth input $\mathbf{x}^{(p)}$.

    Calculate the output error E.

    If $E > \varepsilon$ and $\left\| \mathbf{x}^{(p)} - \mathbf{c}^* \right\| > \delta$ **then**

      Increment the numbers of hidden units and memory items: $J \leftarrow J + 1$ and $M \leftarrow M + 1$.

      Add a hidden unit and create a memory item $(\widetilde{\mathbf{x}}^{(M)}, \widetilde{\mathbf{z}}^{(M)})$, and set the values as follows: $\mathbf{c}_J = \mathbf{x}^{(p)}$, $\mathbf{w}_J = \mathbf{d}^{(p)} - \mathbf{z}$, $\widetilde{\mathbf{x}}^{(M)} = \mathbf{x}^{(p)}$, $\widetilde{\mathbf{z}}^{(M)} = \mathbf{d}^{(p)}$.

    Else

      Calculate hidden outputs for the training sample $(\mathbf{x}^{(p)}, \mathbf{d}^{(p)})$ and M memory items $(\widetilde{\mathbf{x}}^{(M)}, \widetilde{\mathbf{z}}^{(M)})$ $(m = 1, \ldots, M)$ using Eq. (26).

      Define the activity matrix $\boldsymbol{\Phi}$.

      Decompose $\boldsymbol{\Phi}$ with SVD as follows: $\boldsymbol{\Phi} = \mathbf{U}\mathbf{H}\mathbf{V}^T$ where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices, and $\mathbf{H}$ is a diagonal matrix.

      Calculate the weight matrix as follows: $\mathbf{W} = \mathbf{V}\mathbf{H}^{-1}\mathbf{U}^T\mathbf{D}$.

      Give the input $\mathbf{x}^{(p)}$ to RAN-LTM again, and calculate the output error E.

      If $E > \varepsilon$ **then**

        $J \leftarrow J + 1$ and $M \leftarrow M + 1$.

        Add a hidden unit and create a memory item $(\widetilde{\mathbf{x}}^{(M)}, \widetilde{\mathbf{z}}^{(M)})$ and set the values as follows: $\mathbf{c}_J = \mathbf{x}^{(p)}$, $\mathbf{w}_J = \mathbf{d}^{(p)} - \mathbf{z}$, $\widetilde{\mathbf{x}}^{(M)} = \mathbf{x}^{(p)}$, $\widetilde{\mathbf{z}}^{(M)} = \mathbf{d}^{(p)}$.

      End if

    End if

  End For

Output: RAN-LTM

---

5. Face recognition system

Figure 3 shows the overall process in the proposed face recognition system. As seen from Fig. 3, the proposed system mainly consists of the following four sections: face detection, face recognition, face image verification, and incremental learning. The information processing in each section is explained below.

5.1 Face detection

In the face detection part, we adopt a conventional algorithm that consists of two operations: face localization and face feature detection. Figure 4 shows an example of the face detection process.

Facial regions are first localized in an input image by using the skin color information and horizontal edges. The skin color information is obtained by projecting every pixel in the input image to a skin-color axis. This axis was obtained from Japanese skin images in advance. In our preliminary experiment, the face localization works very well with 99% accuracy for a Japanese face database.
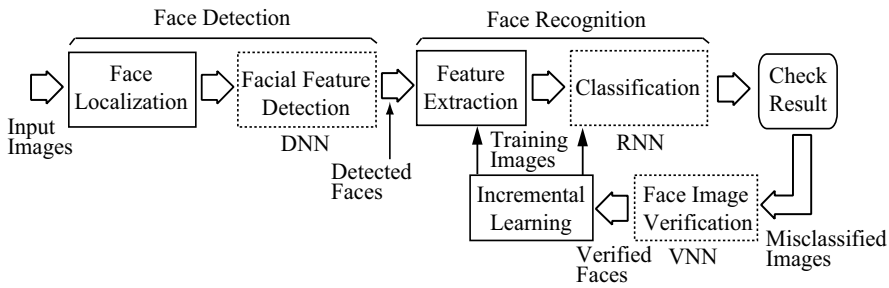
Fig. 3. The processing flow in the face recognition system. DNN and VNN are implemented by RBF networks, while RNN is implemented by RAN-LTM that could learn misclassified face images incrementally. In the feature extraction part, an eigen-space model is incrementally updated for misclassified face images by using Chunk IPCA.
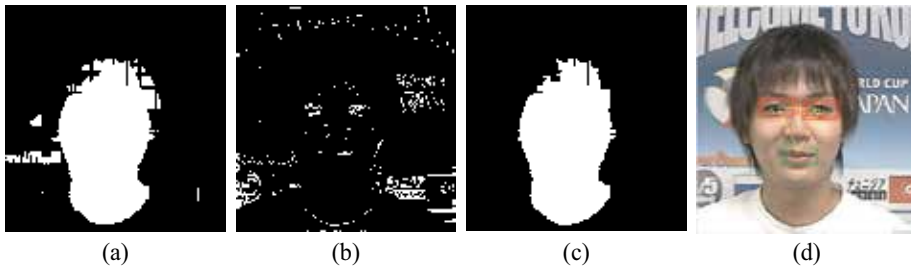

Fig. 4. The process of face detection: (a) an output of skin colour filter and (b) an output of edge filter, (c) a face region extracted from the two filter outputs in (a) and (b), and (d) the final result of the face detection part. Only one face was detected in this case.

After the face localization, three types of facial features (eye, nose, mouth) are searched for within the localized regions through raster operations. In each raster operation, a small sub-image is separated from a localized region. Then, the eigen-features of the sub-image are given to *Detection Neural Network* (DNN) to verify if it corresponds to one of the facial features. The eigenspace model for the face detection part was obtained by applying PCA to a large image dataset of human eye, nose, and mouth in advance. This dataset is also used for the training of DNN.

After all raster operations are done, face candidates are generated by combining the identified facial features. All combinations of three facial features are checked if they satisfy a predefined facial geometric constraint. A combination of three features found on the geometric template qualifies as a face candidate. The output of the face detection part is the center position of a face candidate.

The overall process in the face detection part is summarized in **Algorithm 5**.

5.2 Face recognition and face image verification

In the face recognition part, all the detected face candidates are classified into registered or non-registered faces. This part consists of the following two operations: feature extraction and classification (see Fig. 3). In the feature extraction part, the eigenface approach is adopted here to find informative face features. A face candidate is first

---

Algorithm 5: *Face Detection*

Input: Eigenspace model for the face detection part, DNN, and input image
  For every pixel in the input image **do**
    Calculate the projection (called skin-color feature) of the RGB values of a pixel to a skin-color axis.
    If the skin-color feature exists within a designated domain **then**
      Define the pixel as a skin region.
    Else
      Define it as a non-skin region.
    End if
  End for
  Remove grainy skin regions using a MinMax filter.
  Identify the smallest rectangular region surrounding all the skin regions.
  Redefine this rectangle as a skin region.
  Perform the edge extraction by applying a MaxMin filter and a Sobel horizontal filter to the skin region.
  Search for facial sub-regions including both the skin-color and edge information.
  Extract the sub-regions from the input image.
  For every facial sub-region **do**
    Set the starting point to the upper left corner of the sub-region.
    Repeat
      Extract a 40x40-pixel sub-image.
      Obtain eigen-features of the sub-image from the eigenspace model.
      Provide the eigen-features to DNN as inputs to identify one of the following facial features: eye, nose, or mouth.
      Move the starting point in a raster way.
    Until the starting point reaches to the lower right corner
    Find face candidates satisfying the predefined facial geometrical constraints.
    Obtain the center position of the face candidates.
  End for
Output: Center positions of face candidates.

---

projected to the eigen-axes to calculate eigen-features, and they are given to RAN-LTM called *Recognition Neural Network* (RNN). Then, the classification is carried out based on the outputs of RNN.

If the recognition is correct, RNN should be unchanged. Otherwise, RNN must be trained with the misclassified images to be classified correctly afterward. The misclassified images are collected to carry out incremental learning for both feature extraction part and classifier (RNN). Since the perfect face detection cannot be always ensured, there is a possibility that non-face images happen to be mixed with the misclassified face images. Apparently the training of these non-face images will deteriorate the recognition performance of RNN. Thus, another RBF network called *Verification Neural Network* (VNN) is introduced into this part in order to filter non-face images out.

The procedures of face recognition and verification are summarized in **Algorithm 6**.

Algorithm 6: *Recognition and Verification of Face*

Input: Eigenspace model $\Omega = (\overline{\mathbf{x}}, \mathbf{U}_k, \mathbf{\Lambda}_k, N)$, RNN, VNN, input image, and center positions of face candidates.

   For every face candidate **do**

      Extract a 90x90 sub-image from the input image around the center potition of the face candidate.

      Obtain eigen-features by projecting the sub-image to the eigen-axes $\mathbf{U}_k$.

      Provide the eigen-features to RNN to classify the input face image.

      If misclassification occurs **then**

         Provide the eigen-features to VNN to verify if it is face or non-face.

         If it is verified as a face **then**

            Keep the sub-image with the correct class label.

         End if

      End if

   End for

Output: Misclassified face sub-images and the class labels

5.3 Incremental learning

Since face images of a person can vary depending on various temporal and special factors, the classifier should have high adaptability to those variations. In addition, the useful features might also change over time for the same reason. Therefore, in the face recognition part, the incremental learning should be conducted for both feature extraction part and classifier.

The incremental learning of the feature extraction part is easily carried out by applying Chunk IPCA to misclassified face images. However, the incremental learning of classification part (RNN) cannot be done in a straightforward manner due to the learning of the eigenspace model. That is to say, the inputs of RNN would dynamically change not only in their values but also in the number of input variables due to the eigen-axis rotation and the dimensional augmentation in Chunk IPCA. Therefore, to make RNN adapt to the change in the feature extraction part, not only the network parameters (i.e., weights, RBF centers, etc.) but also its network structure have to be modified.

Under one-pass incremental learning circumstances, this reconstruction of RNN is not easily done without unexpected forgetting of the mapping function that has been acquired so far. If the original RAN is adopted as a classifier, there is no way of retraining the neural classifier to ensure that all previously trained samples can be correctly classified again after the update of the feature space because the past training samples are already thrown away. This can be solved if a minimum number of representative samples are properly selected and used for retraining the classifier. RAN-LTM is suitable for this purpose.

To implement this idea, we need to devise an efficient way to adapt the memory items in RAN-LTM to the updated eigenspace. Let an input vector of the mth memory item be $\widetilde{\mathbf{x}}^{(m)} \in R^I$ and let its target vector be $\widetilde{\mathbf{z}}^{(m)} \in R^K$: $\left\{ \widetilde{\mathbf{x}}^{(m)}, \widetilde{\mathbf{z}}^{(m)} \right\}$ $(m = 1, \ldots, M)$. Furthermore, let the original vector associated with $\widetilde{\mathbf{x}}^{(m)}$ in the input space be $\mathbf{x}^{(m)} \in R^n$. The two input vectors have the following relation: $\widetilde{\mathbf{x}}^{(m)} = \mathbf{U}^T (\mathbf{x}^{(m)} - \overline{\mathbf{x}})$. Now, assume that a new eigenspace model $\Omega' = (\overline{\mathbf{x}}', \mathbf{U}'_{k+l}, \mathbf{\Lambda}'_{k+l}, N + L)$ is obtained by applying Chunk IPCA to a chunk of L training samples $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_L)$. Then, the updated memory item $\widetilde{\mathbf{x}}^{(m)}{}'$ should satisfy the following equation:

$$\widetilde{\mathbf{x}}^{(m)\prime} = \mathbf{U}_{k+l}^{\prime T}(\mathbf{x}^{(m)} - \overline{\mathbf{x}}^{\prime}) = \mathbf{U}_{k+l}^{\prime T}(\mathbf{x}^{(m)} - \overline{\mathbf{x}}) + \frac{1}{N+1}\mathbf{U}_{k+l}^{\prime T}(\overline{\mathbf{x}} - \overline{\mathbf{y}}) \tag{27}$$

where $\overline{\mathbf{x}}^{\prime}$ and $\mathbf{U}_{k+l}^{\prime}$ are given by Eqs. (10) and (16), respectively. The second term in the right-hand side of Eq. (27) is easily calculated. To calculate the first term exactly, however, the information on $\mathbf{x}^{(m)}$, which is not usually kept in the system for reasons of memory efficiency, is needed. Here, let us consider the approximation to the first term without keeping $\mathbf{x}^{(m)}$.

Assume that l of eigen-axes $\mathbf{H}_l$ is augmented. Substituting Eq. (16) into the first term on the right-hand side of Eq. (27), then the first term is reduced to

$$\mathbf{U}_{k+l}^{\prime T}(\mathbf{x}^{(m)} - \overline{\mathbf{x}}) = \mathbf{R}^{T} \begin{bmatrix} \widetilde{\mathbf{x}}^{(m)} \\ \mathbf{H}_l^{T}(\mathbf{x}^{(m)} - \overline{\mathbf{x}}) \end{bmatrix}. \tag{28}$$

As seen from Eq. (28), we still need the information on $\mathbf{x}^{(m)}$ in the subspace spanned by the eigen-axes in $\mathbf{H}_l$. This information was lost during the dimensional reduction process. The information loss caused by this approximation depends on how a feature space evolves throughout the learning. In general, the approximation error depends on the presentation order of training data, the data distribution, and the threshold θ for the accumulation ratio in Eqs. (11) and (12). In addition, the recognition performance depends on the generalization performance of RNN; thus, the effect of the approximation error for memory items is not easily estimated in general. However, recalling a fact that the eigen-axes in $\mathbf{H}_l$ are orthogonal to every vector in the subspace spanned by $\mathbf{U}_k$, the error could be small if an appropriate threshold θ is selected. Then, we can approximate the term $\mathbf{H}_l^{T}(\mathbf{x}^{(m)} - \overline{\mathbf{x}})$ to zero, and Eq. (27) is reduced to

$$\widetilde{\mathbf{x}}^{(m)\prime} \approx \mathbf{R}^{T}\begin{bmatrix} \widetilde{\mathbf{x}}^{(m)} \\ 0 \end{bmatrix} + \frac{1}{N+1}\mathbf{U}^{\prime T}(\overline{\mathbf{x}} - \overline{\mathbf{y}}) \tag{29}$$

Using Eq. (29), memory items in RNN can be recalculated without keeping the memory items $\mathbf{x}^{(m)} \in R^n$ in the input domain even after the eigenspace model is updated by Chunk IPCA. Then, RNN, which is implemented by RAN-LTM, is retrained with L training samples and M updated memory items $\left\{\widetilde{\mathbf{x}}^{(m)\prime}, \widetilde{\mathbf{z}}^{(m)\prime}\right\}$ $(m = 1, \dots, M)$ based on **Algorithm 4**. The procedure of incremental learning is summarized in **Algorithm 7**.

Algorithm 7: *Incremental Learning of Face Recognition Part*

Input: Eigenspace model $\Omega = (\overline{\mathbf{x}}, \mathbf{U}_k, \Lambda_k, N)$, RNN (RAN-LTM + memory items $\left\{\widetilde{\mathbf{x}}^{(m)}, \widetilde{\mathbf{z}}^{(m)}\right\}$ $(m = 1, \dots, M)$), and L training samples $(\mathbf{x}^{(p)}, \mathbf{d}^{(p)})$ $(p = 1, \dots, L)$.

Perform *Chunk IPCA* (**Algorithm 2**) for the training samples $(\mathbf{x}^{(p)}, \mathbf{d}^{(p)})$ $(p = 1, \dots, L)$ to update the current eigenspace model $\Omega = (\overline{\mathbf{x}}, \mathbf{U}_k, \Lambda_k, N)$.

Obtain the eigen-features of $(\mathbf{x}^{(p)}, \mathbf{d}^{(p)})$ $(p = 1, \dots, L)$.

Update all the memory items $\left\{\widetilde{\mathbf{x}}^{(m)}, \widetilde{\mathbf{z}}^{(m)}\right\}$ $(m = 1, \dots, M)$ using Eq. (29).

Perform *Learning of RAN-LTM* (**Algorithm 4**) to train RNN with the eigen-features of the L training samples and the M memory items.

Output: Eigenspace model $\Omega^{\prime} = (\overline{\mathbf{x}}^{\prime}, \mathbf{U}_{k+l}^{\prime}, \Lambda_{k+l}^{\prime}, N+L)$ and RNN

5.4 Overall algorithm of online incremental face recognition system

As mentioned in Section 5.1, the eigenspace model for the face detection part is obtained by applying PCA to a large dataset of human eye, nose, and mouth images. This dataset is used for training DNN as well. The training of VNN is carried out using a different dataset which includes a large amount of face and non-face images. Note that DNN and VNN are trained based on the learning algorithm of RAN (see Section 2). All the trainings for the face detection part, DNN, and VNN are conducted in advance.

Finally, we summarize the overall algorithm of the proposed online incremental face recognition system in **Algorithm 8**.

Algorithm 8: *Online Incremental Face Recognition System*

Input: Initial training data $(\mathbf{x}^{(i)}, \mathbf{d}^{(i)})$ $(i=1, ..., N)$.

    Perform *Training of Initial Eigenspace* (**Algorithm 3**) for $(\mathbf{x}^{(i)}, \mathbf{d}^{(i)})$ $(i=1, ..., N)$ to obtain the eigenspace model $\Omega = (\overline{\mathbf{x}}, \mathbf{U}_k, \Lambda_k, N)$.

    Obtain the eigen-features of $(\mathbf{x}^{(i)}, \mathbf{d}^{(i)})$ $(i=1, ..., N)$.

    Perform *Learning of RAN-LTM* (**Algorithm 4**) to train RNN with the eigen-features.

Loop

    Input: video images of a person

      Repeat

        Perform *Face Detection* (**Algorithm 5**).

        Perform *Recognition and Verification of Face* (**Algorithm 6**).

        Keep a set of misclassified face sub-images and the class labels in the system.

      Until the person is out of sight.

    Perform *Incremental Learning of Face Recognition Part* (**Algorithm 7**).

End loop

---

6. Performance evaluation

6.1 Experimental Setup

To simulate real-life environments, 224 video clips are collected for 22 persons (19 males / 3 females) during about 11 months such that temporal changes in facial appearances are included. Seven people (5 males / 2 females) are chosen as registrants and the other people (14 males /a female) are non-registrants. The duration of each video clip is 5-15 (sec.). A video clip is given to the face detection part, and the detected face images are automatically forwarded to the face recognition part. The numbers of detected face images are summarized in Table 1. The three letters in Table 1 indicate the code of the 22 subjects in which M/F and R/U mean Male/Female and Registered/Unregistered, respectively; for example, the third registered male is coded as MR3.

The recognition performance is evaluated through two-fold cross-validation; thus, the whole dataset is subdivided into two subsets: Set A and Set B. When Set A is used for learning RNN, Set B is used for testing the generalization performance, and vice versa. Note that since the incremental learning is applied only for misclassified face images, the recognition accuracy before the incremental learning is an important performance measure. Hence, there are at least two performance measures for the training dataset: one is the performance of RNN using a set of training samples given at each learning stage, and the other is the performance using all

training datasets given so far after the incremental learning is carried out. In the following, let us call the former and latter datasets as *incremental dataset* and *training dataset*, respectively. Besides, let us call the performances over the incremental dataset and training dataset as *incremental performance* and *training performance*, respectively. We divide the whole dataset into 16 subsets, each of which corresponds to an incremental dataset. Table 2 shows the number of images included in the incremental datasets.

Set	MR1	FR1	MR2	MR3	FR2	MR4	MR5	FU1
A	351	254	364	381	241	400	136	133
B	170	220	297	671	297	241	359	126

Set	MU1	MU2	MU3	MU4	MU5	MU6	MU7	MU8
A	131	294	110	103	170	136	174	33
B	228	292	80	233	117	202	182	14

Set	MU9	MU10	Mu1	Mu12	Mu13	Mu14	Total
A	79	15	75	17	10	9	3766
B	9	14	28	18	9	9	3816

Table 1. Two face datasets (Set A and Set B) for training and test. The three letters in the upper row mean the registrant code and the values in the second and third rows are the numbers of face images.

Set	1	2	3	4	5	6	7	8
A	220	232	304	205	228	272	239	258
B	288	204	269	246	273	270	240	281

Set	9	10	11	12	13	14	15	16
A	212	233	290	212	257	188	199	217
B	205	249	194	241	214	226	210	206

Table 2. Number of images included in the 16 incremental datasets.

Stage	Init.	1	2	…	12	13	14	15
Case 1	1	2	3	…	13	14	15	16
Case 2	1,2	3	4	…	14	15	16	---
Case 3	1,2,3	4	5	…	15	16	---	---

Table 3. Three series of incremental datasets. The number in Table 2 corresponds to the tag number of the corresponding incremental dataset.
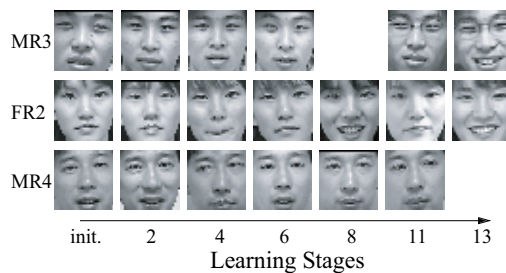
Fig. 6. Examples of face images trained at different learning stages.

The size of an initial dataset can influence the test performance because different initial eigen-spaces are constructed. However, if the incremental learning is successfully carried out, the final performance should not depend on the size of the initial dataset. Hence, the three different series of incremental datasets shown in Table 3 are defined to see the influence. Note that the number in Table 3 corresponds to the tag number (1-16) of the incremental dataset in Table 2. Hence, we can see that Case 1 has 15 learning stages and the number of images in the initial dataset is 220 for Set A and 288 for Set B, which correspond to 6.7% and 7.5% over the whole data. On the other hand, the sizes of the initial datasets in Case 2 and Case 3 are set to a larger value as compared with that in Case 1; while the numbers of learning stages are smaller than that in Case 1. Figure 6 shows the examples of detected face images for three registered persons at several learning stages.

When an initial dataset is trained by RNN, the number of hidden units is fixed with 50 in this experiment. The other parameters are set as follows: $\sigma^2 = 7$, $\varepsilon = 0.01$, and $\delta = 5$. The threshold θ of the accumulation ratio in IPCA is set to 0.9; thus, when the accumulation ratio is below 0.9, new eigen-axes are augmented.

6.2 Experimental results

Figure 7 shows the evolution of learning time over 15 learning stages when the chunk size L is 10 in Chunk IPCA (CIPCA). The curves of *CIPCA* and *IPCA* show the learning time for feature extraction, while those of *CIPCA+RAN-LTM* and *IPCA+RAN-LTM* mean the learning time for both feature extraction part and classifier. As you can see from the results, the learning time of feature extraction by Chunk IPCA is greatly reduced as compared with IPCA. This is also confirmed in Table 4.

The learning time of Chunk IPCA decreases as the chunk size increases, and Chunk IPCA is much faster than IPCA even though the feature dimensions at the final stage do not have large differences between IPCA and Chunk IPCA. When the chunk size is 10, Chunk IPCA is about 8 times faster than IPCA. The reason why the decreasing rate of the learning time becomes small for larger chunk size is that the time for finding eigen-axes dominates the total learning time (Ozawa et al., 2008).

To evaluate the effectiveness of learning an eigenspace, the classification accuracy of RAN-LTM is examined when the following three eigenspace models are adopted:

1.   Static eigenspace model using PCA
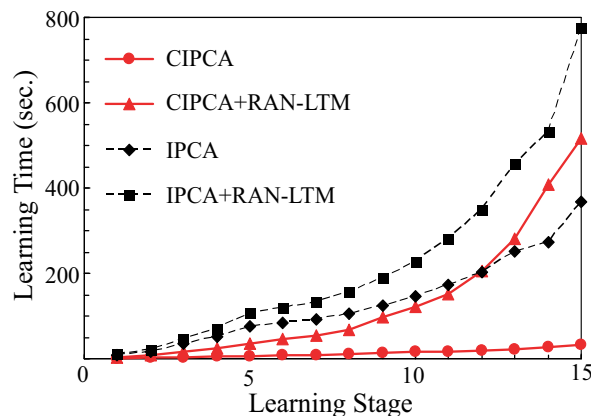
Fig. 7. Evolution of learning time for four different models (sec.).

	IPCA	CIPCA(10)	CIPCA(50)	CIPCA(100)
Time (sec.)	376.2	45.6	22.5	18.1
Dimensions	178	167	186	192

Table 4. Comparisons of Learning time and dimensions of feature vectors at the final learning stage. CIPCA(10), CIPCA(50), and CIPCA(100) stand for Chunk IPCA in which the chunk sizes are set to 10, 50, and 100, respectively.

2.   Adaptive eigenspace model using the extended IPCA
3.   Adaptive eigenspace model using Chunk IPCA.

Figures 8 (a)-(c) show the evolution of recognition accuracy over 15 learning stages when the percentage of initial training data is (a) 6.7%, (b) 12.5%, and (c) 20%, respectively. As stated before, the size of an initial dataset can influence the recognition accuracy because different eigenspaces are constructed at the starting point. As seen from Figs. 8 (a)-(c), the initial test performance at stage 0 is higher when the number of initial training data is larger; however, the test performance of IPCA and Chunk IPCA is monotonously enhanced over the learning stages and it reaches almost the same accuracy regardless of the initial datasets. Considering that the total number of training data is the same among the three cases, we can say that the information on training samples is stably accumulated in RNN without serious forgetting even though RNN is reconstructed all the time the eigenspace model is updated.

In addition, the test performance of RNN with IPCA and Chunk IPCA has significant improvement against RNN with PCA. This result shows that the incremental learning of a

(a) Case 1 (Initial Data: 6.7%)   (b) Case 2 (Initial Data: 12.5%)

(c) Case 3 (Initial Data: 20%)
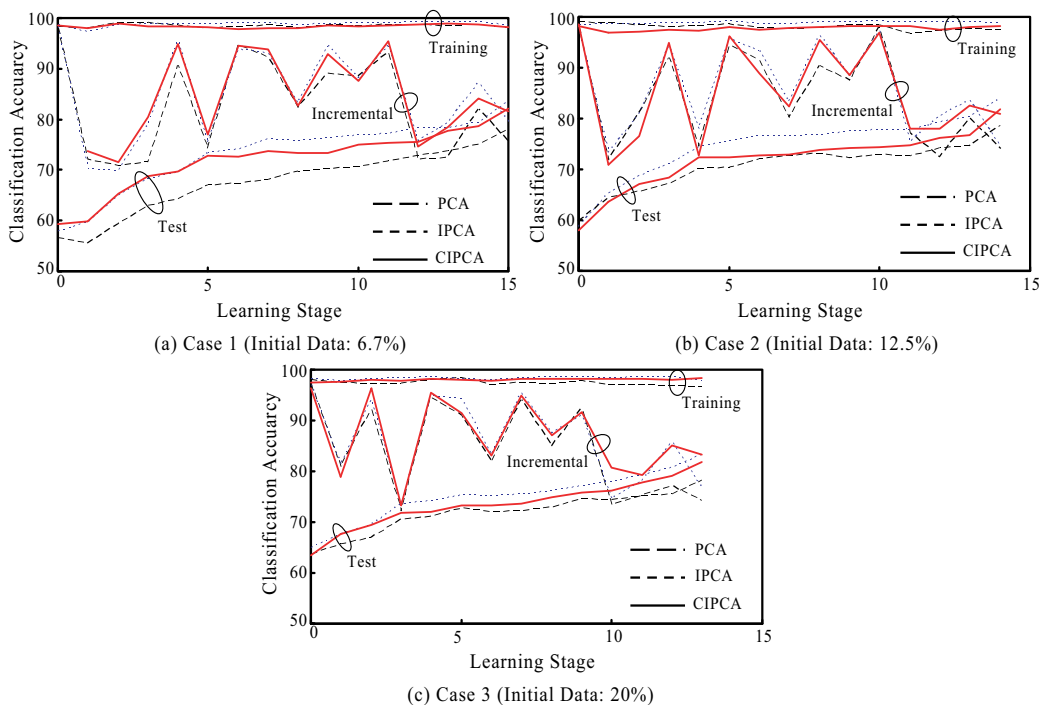
Fig. 8. Evolution of recognition accuracy for three different datasets (incremental, training, test) over the learning stages when the percentages of initial training datasets are set to (a) 6.7%, (b) 12.5%, and (c) 20.0%.

feature space is very effective to enhance the generalization performance of RNN. However, Chunk IPCA has slightly lower performance than IPCA. It is considered that this degradation originates from the approximation error of the eigenspace model using Chunk IPCA.

In Figs. 8 (a)-(c), we can see that although the incremental performance is fluctuated, the training performance of RNN with IPCA and Chunk IPCA changes very stably over the learning stages. On the other hand, the training performance of RNN with PCA rather drops down as the learning stage proceeds. Since the incremental performance is defined as a kind of test performance for the incoming training dataset, it is natural to be fluctuated. The important result is that the misclassified images in the incremental dataset are trained stably without degrading the classification accuracy for the past training data.

From the above results, it is concluded that the proposed incremental learning scheme, in which Chunk IPCA and RAN-LTM are simultaneously trained in an online fashion, works quite well and the learning time is significantly reduced by introducing Chunk IPCA into the learning of the feature extraction part.

7. Conclusions

This chapter described a new approach to constructing adaptive face recognition systems in which a low-dimensional feature space and a classifier are simultaneously learned in an online way. To learn a useful feature space incrementally, we adopted Chunk Incremental Principal Component Analysis in which a chunk of given training samples are learned at a time to update an eigenspace model. On the other hand, Resource Allocating Network with Long-Term Memory (RAN-LTM) is adopted as a classifier model not only because incremental learning of incoming samples is stably carried out, but also because the network can be easily reconstructed to adapt to dynamically changed eigenspace models.

To evaluate the incremental learning performance of the face recognition system, a self-compiled face image database was used. In the experiments, we verify that the incremental learning of the feature extraction part and classifier works well without serious forgetting, and that the test performance is improved as the incremental learning stages proceed. Furthermore, we also show that Chunk IPCA is very efficient compared with IPCA in term of learning time; in fact, the learning speed of Chunk IPCA was at least 8 times faster than IPCA.

8. References

Carpenter, G. A. and Grossberg, S. (1988). The ART of Adaptive Pattern Recognition by a Self-organizing Neural Network, *IEEE Computer*, Vol. 21, No. 3, pp. 77-88.

Hall, P. & Martin, R. (1998). Incremental Eigenanalysis for Classification, *Proceedings of British Machine Vision Conference*, Vol. 1, pp. 286-295.

Hall, P. Marshall, D. & Martin, R. (2000). Merging and Splitting Eigenspace Models, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 9, pp. 1042-1049.

Jain, L. C., Halici, U., Hayashi, I., & Lee, S. B. (1999). *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, CRC Press.

Kasabov, N. (2007). *Evolving Connectionist Systems: The Knowledge Engineering Approach*, Springer, London.

Kobayashi, M., Zamani, A., Ozawa, S. & Abe, S. (2001). Reducing Computations in Incremental Learning for Feedforward Neural Network with Long-term Memory, *Proceedings of Int. Joint Conf. on Neural Networks*, Vol. 3, pp. 1989-1994.

Oja, E. & Karhunen, J. (1985). On Stochastic Approximation of the Eigenvectors and Eigenvalues of the Expectation of a Random Matrix, *J. Math. Analysis and Application*, Vol. 106, pp. 69-84.

Ozawa, S., Pang, S., & Kasabov, N. (2004) A Modified Incremental Principal Component Analysis for On-line Learning of Feature Space and Classifier. In: *PRICAI 2004: Trends in Artificial Intelligence*, Zhang, C. et al. (Eds.), LNAI, Springer-Verlag, pp. 231-240.

Ozawa, S., Toh, S. L., Abe, S., Pang, S., & Kasabov, N. (2005). Incremental Learning of Feature Space and Classifier for Face Recognition, *Neural Networks*, Vol. 18, Nos. 5-6, pp. 575-584.

Ozawa, S., Pang, S., & Kasabov, N. (2008). Incremental Learning of Chunk Data for On-line Pattern Classification Systems, *IEEE Trans. on Neural Networks*, Vol. 19, No. 6, pp. 1061-1074.

Pang, S., Ozawa, S. & Kasabov, N. (2005). Incremental Linear Discriminant Analysis for Classification of Data Streams, *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, Vol. 35, No. 5, pp. 905-914.

Sanger, T. D. (1989). Optimal Unsupervised Learning in a Single-layer Linear Feedforward Neural Network, *Neural Networks*, Vol. 2, No. 6, pp. 459-473.

Weng, J., Zhang Y. & Hwang, W.-S. (2003). Candid Covariance-Free Incremental Principal Component Analysis, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 8, pp. 1034-1040.

Weng, J. & Hwang, W.-S. (2007). Incremental Hierarchical Discriminant Regression, *IEEE Trans. on Neural Networks*, Vol. 18, No. 2, pp. 397-415.

Zhao, H., Yuen, P. C. & Kwok, J. T. (2006). A Novel Incremental Principal Component Analysis and Its Application for Face Recognition, *IEEE Trans. on Systems, Man and Cybernetics, Part B*, Vol. 36, No. 4, pp. 873-886.

High Speed Holographic Optical Correlator for Face Recognition

Eriko Watanabe and Kashiko Kodate
Faculty of Science, Japan Women's University,
Japan

1. Introduction

Owing to the Japanese government plan, U-Japan, which promised to bring about the so-called 'ubiquitous society' by 2010, the use of Internet has dramatically increased and accordingly, development of the system through IT networks is thriving. The term 'ubiquitous society' became a buzzword, signifying easy access to content on the internet for anybody, anywhere and at any time. Face recognition has become the key technique, as a 'face' carries valuable information, captured for security purposes, without physical contact. They can function as identity information for purposes such as login for bank accounts, access to buildings, anti-theft or crime detection systems using CCTV cameras. Furthermore, within the domain of entertainment, face recognition techniques are applied to search for celebrities who look alike. Against this backdrop, a high performance face recognition system is sought after.

Face recognition has been used in a wide range of security systems, such as monitoring credit card users, identifying individuals with surveillance cameras and monitoring passengers at immigration control. Face recognition has been studied since the 1970s, with extensive research into and development of digital processing (Kaneko & Hasegawa, 1999; Kanade, 1971 ; Sirovich & Kirby, 1991 ; Savvides, M. et al. 2004). Yet there are still many technical challenges to overcome; for instance, the number of images that can be stored is limited in currently available systems, and the recognition rate needs to be improved to take account of photographic images taken at different angles and in varying conditions.

In contrast to digital recognition, optical analog operations process two-dimensional images instantaneously and in parallel, using a lens-based Fourier transform function. In the 1960s, two main types of correlator came into existence; the Vanderlugt Correlator and the Joint Transform Correlator (JTC) (Goodman & Moeller, 2004). Some correlators were a combination of the two (Thapliya & Kamiya, 2000; Kodate Inaba Watanabe & Kamiya, 2002 ; Kobayashi & Toyoda, 1999 ; Carrott Mallaley Dydyk & Mills, 1998). The authors previously proposed and produced the FARCO (Fast Face Recognition Optical Correlator), which was based on the Vanderlugt Correlator((a) Watanabe & Kodate 2005; (b)Watanabe & Kodate, 2005). Combined with high-speed display devices, four-channel processing was able to achieve operational speeds of up to 4000 faces/s. Running trial experiments on a 1-to-N identification basis using the optical parallel correlator, we succeeded in acquiring low error rates of 1 % False Acceptance Rate (FAR) and 2.3 % False Rejection Rate (FRR)(Savvides et

al., 2004).We also developed an algorithm for a simple filter by optimizing the calculation algorithm, the quantization digits and the carrier spatial frequency for optical correlation. This correlation filter is more accurate, compared with classical correlation.

Recently, a novel holographic optical storage system that utilizes collinear holography has been demonstrated (Horimai & Tan, 2005). This scheme can realize practical and small holographic optical storage systems more easily than conventional off-axis holographic optical systems. At present, the system seems to be most promising for ultrahigh density volumetric optical storage.

Moreover, we proposed the super high-speed FARCO (S-FARCO) ((a) Watanabe & Kodate, 2006; (b) Watanabe & Kodate, 2006) that integrates optical correlation technology used in FARCO and a co-axial holographic optical storage system (Horimai Tan & Li, 2006). Preliminary correlation experiments using the co-axial optical set-up show an excellent performance of high correlation peaks and low error rates. This enables optical correlation without the need to decode information in the database, greatly reducing correlation time. We expect the optical correlation speed to be about 3 µs/frame, assuming 24000 pages of hologram in one track rotating at 600 rpm. A correlation speed faster than 370,000 frames/s was acquired when the system was used. Therefore, the S-FARCO system proved effective as a 1-to-N recognition system with a large database. It should be noted also that the advantage of our system lies in its wide applicability to various correlation schemes.

In recent years, the processing speed of computers has improved greatly. For example, the operation speed of a 128x128 pixels Fast Fourier Transform (FFT) is now about 30ms (CPU: 3GHz, 2GB). When processing the images of several tens of people, the recognition process time can be calculated by the software within a few seconds. Against this background, we propose three different configurations, which depend on the correlation speed and size of shown in Figure 1. FARCO is used for several thousand people at a correlation speed of 4000 faces per second. In response to demand for greater speed or more images, the S-FARCO system was applied. Optical correlation of 2.7µs/face is expected, assuming that 376800 faces can be processed in one second with 10 µm pitch of hologram in one track rotating at 600 rpm in S-FARCO 2.0 and 2.5. S-FARCO 2.5 is a smaller version of the

	FARCO Software	S-FARCO2.0	S-FARCO2.5
	2007 ver.1	2007	2008
Application	1:1 verification or 1:N identification for subjects numbering in the tens	1:N identification for several thousand to several hundred thousand or more subjects	1:N identification for several thousand to several hundred thousand or more subjects
Size [mm]		680 (W)×1120(B)×400(H)	450 (W)×750(B)×400(H)
Operation speed	10ms/faces	3µs/frames	3µs/frames
Format			

Fig. 1. Three different FARCO configurations

previous model (S-FARCO 2.0), with its size reduced by 40%, and is portable. Applied as a face recognition system, it is then possible to correlate more than 376800 faces per second. Software was also proposed for one-to-one ID recognition, which requires less calculation time.

In this chapter, we propose a much more rapid face recognition system using a holographic optical disc system named FARCO 2.0. Section 2 describes the concept of the optical parallel correlation system for facial recognition and the dedicated algorithm. Section 3 presents a correlation engine of a much higher speed for face, image and video data using optical correlation.

Section 4 presents an online face recognition system using the software which was constructed for FARCO algorithm based on phase information. Section 5 proposes a video identification system using a S-FARCO. Section 6 presents a discussion based on the results and summarizes the paper.

2. The concept of the optical correlation system for facial recognition and the dedicated algorithm

In this section we describe the concept of the optical parallel correlation system for facial recognition and the dedicated algorithm. A novel filtering correlation for face recognition which uses phase information with emphasis on the Fourier domain will be introduced. The filtering correlation method will be evaluated by comparing it with various other correlation methods.

2.1 Fast Face Recognition Optical Correlator (FARCO)

An algorithm for the FARCO is shown in Figure 2. In this system, pre- and post-processes with a PC are highly conducive to the enhancement of the S/N ratio and robustness. Firstly, facial images were captured automatically by a digital video camera. The two eyes are used
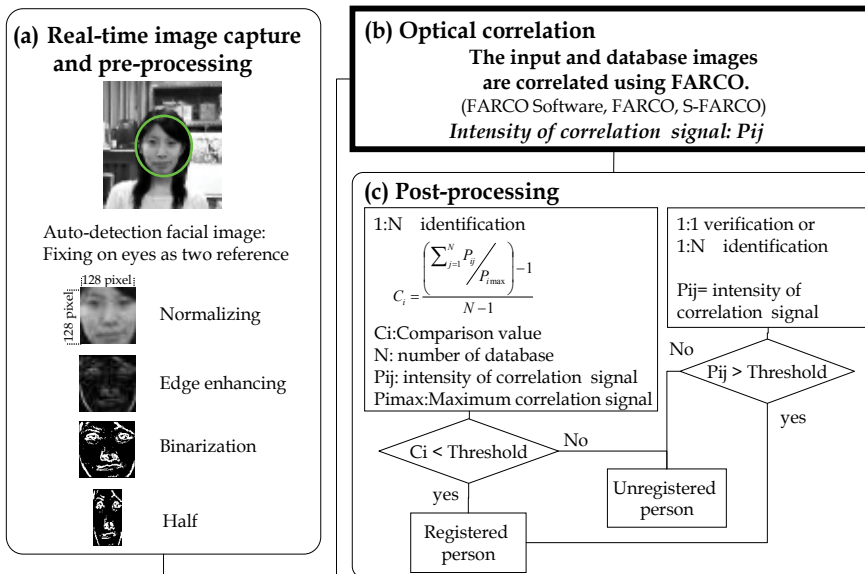
Fig. 2. Our hybrid facial recognition system: flow-chart representation

as focal points. The size of the extracted image was normalized to 128 x 128 pixels by the center. For input images taken at an angle, affined. transformation was used to adjust the image and the image was normalized, fixing on the position of the eyes. This was followed by edge enhancement with a Sobel filter, which binarized and defined the white area as 20%, and equalized the volume of transmitted light in the image. We have shown previously that binarization of the input (and database) images with appropriate adjustment of brightness is effective in improving the quality of the correlation signal.

The correlation signal is classified by a threshold level. In practical applications, the threshold value must be customized. The threshold value varies depending on its security level; on whether the system is designed to reject an unregistered person or permit at least one registered person. The optimum threshold value must be decided using the appropriate number of database images based on the biometrics guideline (Mansfield & Wayman, 2002) for each application. In this paper, the threshold value is fixed where the Equal Error Rate (EER) is at its lowest.

2.2 Design of correlation filter for practical face recognition software
2.2.1 Filtering correlation

In our previous work, the correlation filter of FARCO for the optical correlator was designed by focusing on the binary level and correlation signals, not overlapped by the 0th-order image, with an emphasis on the Fourier domain. The carrier-spatial frequency should be contained within the minimum frequency range of facial characteristics (details described in ((c) Watanabe & Kodate, 2005). In this section, we select parameters to optimize the correlation filter in accordance with the correlation speed for software. We call this method "filtering correlation" (Horner & Gianino, 1982), which will be evaluated in reference to the following two other methods (Watanabe & Kodate, 2005).

2.2.2 Phase-only correlation

The correlation function $g(x, y)$ between two signals, $f(x, y)$ and $h(x, y)$ is expressed as the following Equation (1) using Fourier transform formulation

$$g (x , y) = F [F(u , v)H^* (u , v)] \tag{1}$$

in which * denotes its conjugate. F denotes the Fourier transform operator. While $F(u , v)$ is the Fourier transform of one signal $f(x, y)$, $H^*(u, v)$ is the correlation filter corresponding to the other signal $h(x, y)$, and u and v stand for two vector components of the spatial frequency domain. The classical correlation filter for a signal $h(x, y)$ was defined as $H^*(u, v)$. By setting every amplitude at the number equal to 1 or alternatively by multiplying it by $1/H (u, v)$, we obtained the phase-only filter (Horner & Gianinor, 1984).

$$H_P(u , v) = \exp \{- i \, \varphi(u , v)\} \tag{2}$$

where p stands for phase.

The performance of the two correlation methods was evaluated through one-to-N identification with a database of 30 frontal facial images. As shown in Figure 3, the database (Tarres (web)) is composed of facial images that vary in different ways (laughing, wearing glasses, different races and so on). Three correlation methods were examined for three image sizes: (a) 32 x 16, (b) 64 x 32 and (c) 128 x 64 respectively (Figure 4).

Fig. 3. Examples of database and input facial images
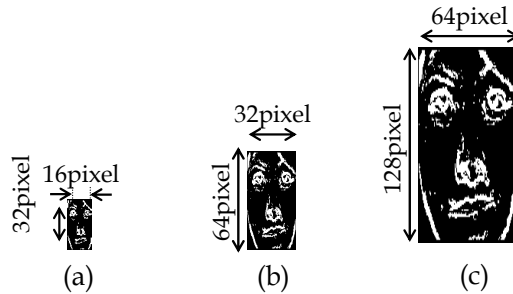
Fig. 4. Examples of database images of different sizes.(a) 32 x 16pixel, (b) 64 x 32pixel (c) 128 x 64pixel.

2.3 Experimental results

Experimental error rates of two different types of correlation methods are shown in Figure 5 and Table 1. If the intensity exceeded a threshold value, the input image would be regarded as a match with a registered person. Error rates divided by the total number of cases were given by the FRR and FAR. With the threshold value set at an optimum value (arbitrary units), the FAR and FRR are shown in Table 1. Error rates are plotted on the vertical axis and comparison values on the horizontal axis. EER has been improved by 0%.

As for the filtering correlation, EER attained the lowest value from among all the correlation methods, as shown in Figure 5 and Table 1. In a low resolution of 64x64 pixels, the EER reached 0% in the Filtering correlation only (both FRR and FAR are 0% as shown in Table 1. If the resolution is lowered to 32 pixels, the FRR becomes 100% at FAR 0%. These results indicate that the registered person cannot be recognized without accepting the others.

Because the FRR value can be improved by trying to log in as a user of the system several times, the value of the FRR at FAR 0% is important for the recognition system. Therefore, the Filtering correlation can be counted as an advantage. The Filtering correlation works effectively in the application targeted at low resolution images.
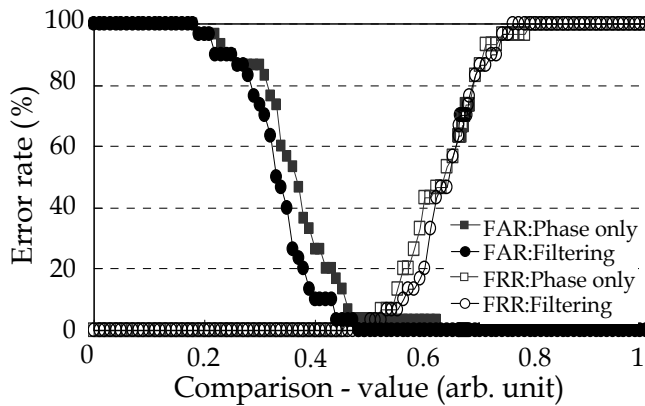
Fig. 5. Results for two kinds of correlation with 64 x 64 pixel

Image size Method	32x32(pixels)			64x64(pixels)			128x128(pixels)		
	FAR	FRR	**EER**	FAR	FRR	**EER**	FAR	FRR	**EER**
Phase-only correlation	0	100	**33**	0	46.7	**3.3**	0	3.3	**3.3**
Filtering correlation	0	100	**26**	**0**	**0**	**0**	0	0	**0**

Table 1. Experimental error rates of two different methods

3. A fast face recognition optical correlator of a much higher speed for face, image and video data using holographic optical correlator filter

This section presents a correlator of a much higher speed for face, image and video data using optical correlation. The data access rate of a conventional correlator is limited to a maximum of 1Gbps, due to the data transfer speed of the HDD used to store digital reference images. Therefore, a conventional correlator has a weakness in its image data transmission speed. Recently, a novel holographic optical storage system that utilizes co-axial holography has been demonstrated. This scheme can realize practical and small holographic optical storage systems more easily than conventional on-axis holographic optical systems. Using the ability of parallel transformation as holographic optical memory, the recognition rate can be vastly improved. In addition, the large capacity of optical storage allows us to increase the amount of data in the reference database. Preliminary correlation experiments using the holographic optical disc set-up show an excellent performance of high correlation peaks and low error rates at a multiplexing pitch of 10 μm and rotational speed of 300rpm. It is clear that the processing speed of our holographic optical calculation is remarkably high compared to the conventional digital signal processing architecture.

No storage device has yet been found, which meets both conditions, i.e. transfer speed and data capacity. DRAM has a high-speed data transfer rate, yet with a limited data capacity of up to several GB. The typical secondary storage devices include the hard disk drive, optical disc drive and magnetic tape streamer devices. HDD technology has been making significant progress in expanding data capacity. Recently, the capacity of HDD data storage has expanded to more than 1TB. However, even if a RAID system (Redundant Arrays of Inexpensive Disks) is used, the maximum transfer rate of a conventional HDD system is

limited to the order of G bps. Typically, the input digital data is first transferred from HDD to the DRAM, followed by calculation of correlation. Therefore, a conventional image search correlation with large image database has a weakness in its image data transmission speed (Figure 6). It is demonstrated that the processing speed of our holographic optical calculation is remarkably higher than that of the conventional digital signal processing architecture (Figure 7).
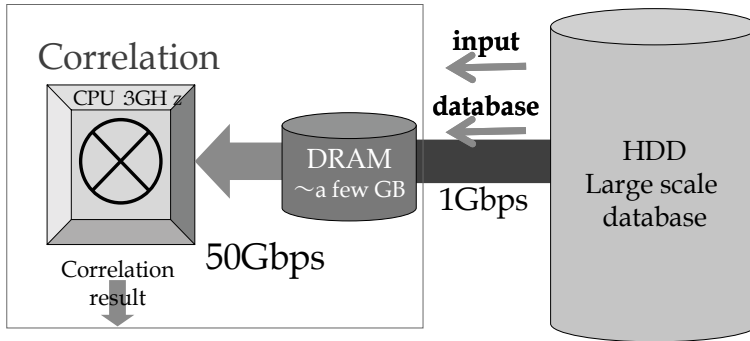
Fig. 6. Conventional search engine
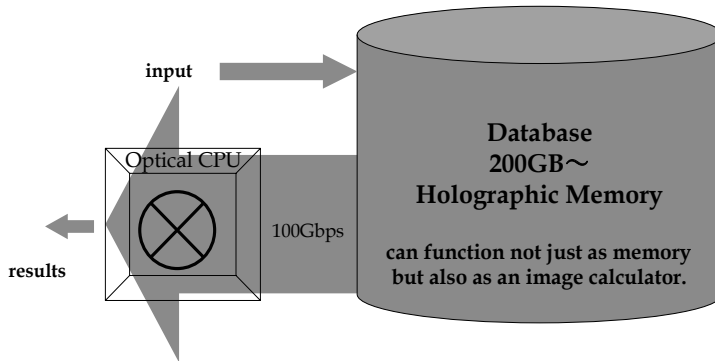
Fig. 7. Optical Correlation system

Figure 8 shows the concept of the high-speed optical correlator with a holographic optical disc. We call this system the Super Fast Recognition Optical Correlator, S-FARCO. A huge amount of data can be stored in the holographic optical disc in the form of matched filter patterns. In case the correlation process, an input image on the same position are illuminated the laser beam, the correlation signal appears through the matched filter on the output plane. The optical correlation process speeds up by simply rotating the optical disc at higher latency.

3.1 Holographic optical memory

Holographic optical memory, as the fourth-generation memory device with a large data storage capacity, has been developed with high expectations for replacing the current optical disc devices such as Blu-ray Disc and HD-DVD. Among other devices which belong to the same 'generation' (i.e. category), there are Near-field optical memory (Goto, K. 2004, Super-RENS (Tominaga et al., 2002), two-photon absorption memory (Kawata & Nakano, 2005). However, they are essentially all fit for recording two-dimensional data. Some enable
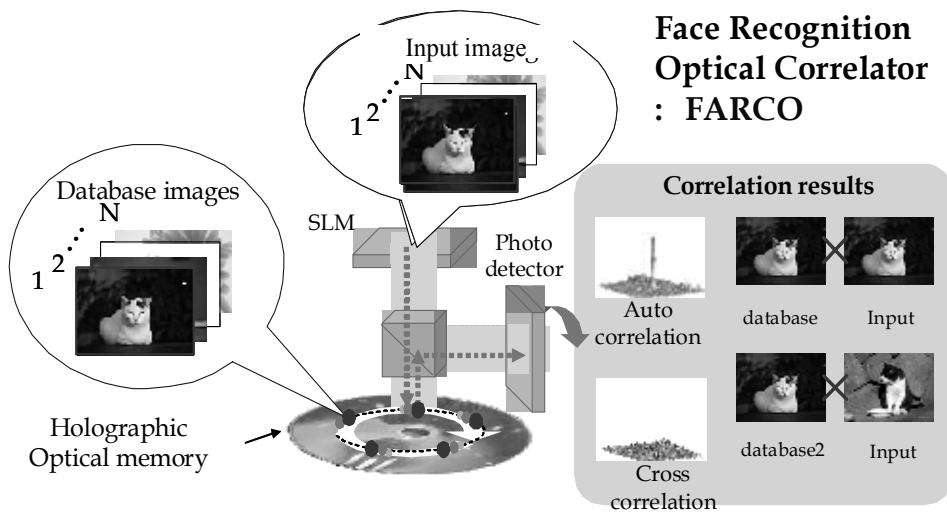
Fig. 8. Optical correlator using holographic optical disc : S-FARCO

high density by setting the recording bit below the level of the diffraction limit, while others make it possible to record data on multi-layers, holding the density constant. In contrast, holographic optical memory records data three-dimensionally across the whole recording material. The history of research into holographic optical memory dates back to 1948, one year after Dennis Gabor discovered holography (Coufal Psaltis & Sincerbox, 2000). In 1960, when holographic optical memory was first applied, combined with laser as a light source, some attention was focused on the technique of recording and reproducing wavelength. It was van Heerden who proposed holography as a memory device in 1963 (van Heerden 1963). Nevertheless, despite a rather long history in research, holographic optical memory was not applied for practical use. This could be ascribed to the fact that sufficient progress had not been made in two-dimensional image display, image pick-up devices and recording materials. In the 1990s, there were some breakthroughs in the development of PRISM (photorefractive information storage materials) and HDSS (holographic data storage system), due to the US government-funded projects (Hesselink, 2000; Orlov, 2000). In parallel with this development, holographic optical memory made progress. However, there were still a number of issues to overcome before it could be applied more widely. For instance, a large size (of space) is required for optical setup due to two interference or the difficulty in preventing deterioration of recording material quality. With this background, in 2004, a optical disc-shaped co-axial-type holographic optical memory was developed (Horimai & Tan 2005). This holographic optical memory enabled both a reference beam and object beam to be juxtaposed on the same axis, which is conducive to miniaturization. This could solve the issue of size, which was common under the two-interference system. Moreover, it is a reflecting-type optical disc memory of 12cm in diameter to which strengths in optical disc drive technique can be directly applicable. Therefore, this optical disc memory could be a promising device for the next generation.

The basic structure of the conventional optical device and co-axial holographic optical memory are shown in Figure 9(a) and (b) . Comparing these two types, it is predicted that the latter system, in which juxtaposition of reference and object beam on the same axis is possible, can be slimmed down.

The co-axial holographic optical system consists of a DMD (Digital Micro-mirror Device) as a two-dimensional spatial laser modulator, which displays two-dimensional digital data on the two-dimensional plane, and photopolymer as a recording material, a CMOS camera as a image device for reading out reproduced two-dimensional data and a lens (NA: 0.55) for image formation. Holding two beams (i.e. object and reference) on the same axis, the object light is placed at the centre of the image, while the reference light is on the outside. The beam from the DMD is passing through at the objective lens, and causes interference in the recording medium. DMD is illuminated by plane waves, its mirror focus light, which was modulated by the on/off switch into the recording material by objective lens. At the time of recording the data, both reference and signal beam are displayed. When images are reproduced, only reference image is displayed. The reproduced image becomes higher power, when it is closer to the reference image at the time of recording.

(a)   (b)
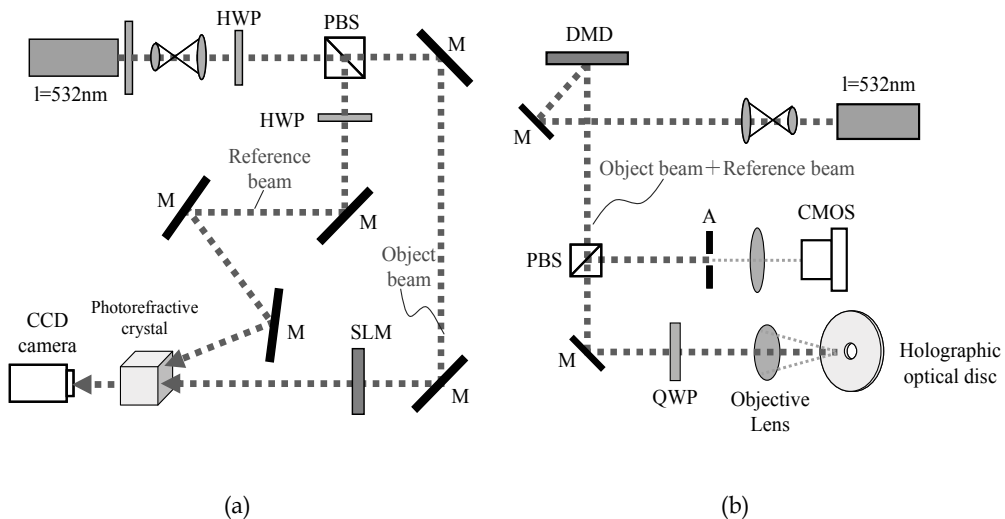
Fig. 9. (a) Two beam interference optical system, (b) Co-axial holographic optical memory system

An outline of the structure of the co-axial holographic optical memory is given in Figure 10. The recording material is sandwiched between two glasses, one coated by AL and the other by AR coated, and it is a reflection type memory. Write once photopolymer is used as a recording material. The spatial distribution is recorded through the distribution of refraction (Schilling L. M. et al. 1999 ; Sato, et al., 2006). Photopolymer is a photopolymerization monomer. At the initial stage, there are two types of monomers for maintaining the configurations: monomer 1 which photopolymerize by corresponding to recording light and monomer 2 which does not correspond to the recording light. In proportion to the intensity of light, monomer 1 becomes polymerized, as monomer 2 gets pushed out into polymer-free space. At the stage of multiple recording, the monomer reduces in its density, and its sensitivity decreases accordingly. As all the data are recorded and the remaining monomer is completely polymerized, there will be no change even when it is illuminated by reproduced light.

(a)   (b)
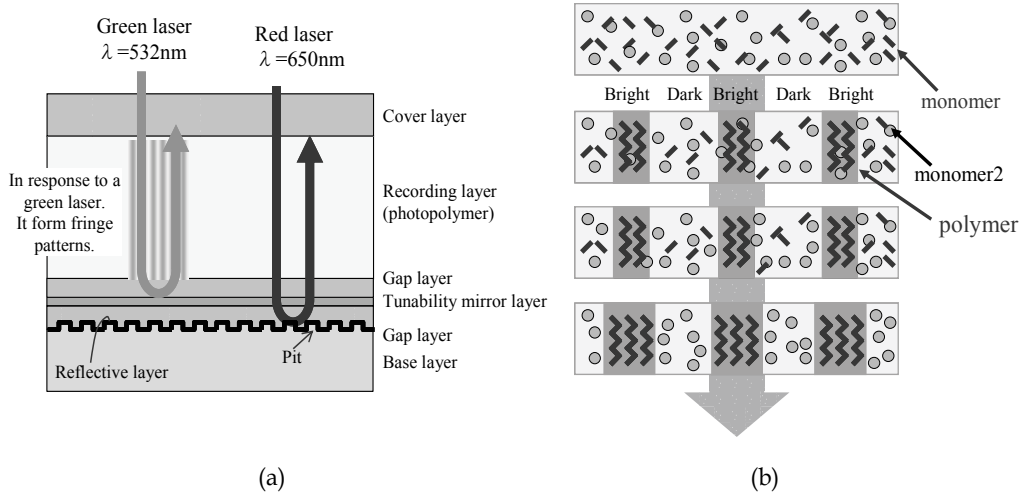
Fig. 10. The configuration of co-axial holographic optical memory and photopolymer. (a) Configuration of holographic optical memory, (b) Photopolymer curing

3.2 High speed optical correlation system

Figure 11 shows the schematic of our optical configuration, which is identical to the one used in a collinear holographic optical storage system. Note that in the collinear holographic system, the recording plane is the Fourier plane of the digital mirror device (DMD) image, as shown in the close-up part. The recorded image is composed of a reference point and the image to be recorded in the database, as shown in Figure 11 This image is Fourier transformed by the objective lens shown in Figure 11, and recorded as a hologram. This hologram works as the correlation filter. With the recorded image of one pixel as a delta function and database image, we can easily obtain the correlation filter in the co-axial holography system. Figure 11 shows the optical setup of the Fourier plane in close up.
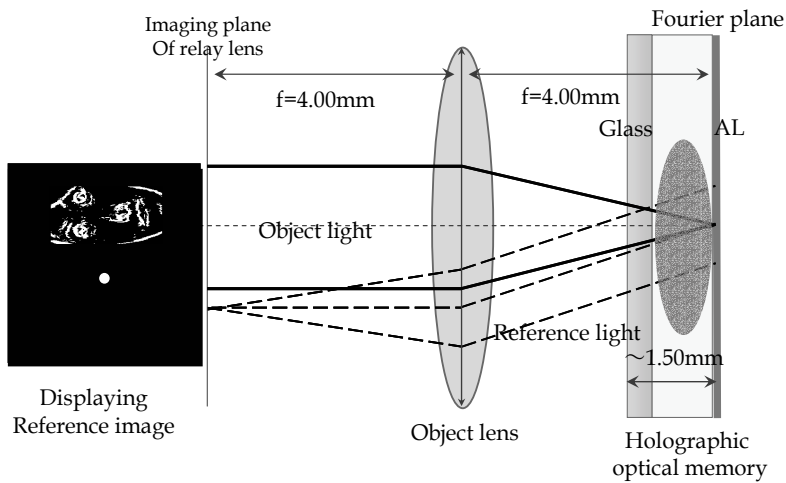
Fig. 11. The inset shows the enlarged part of the Fourier transformation part

Writing a matched filter hologram, the recording image on the DMD is Fourier-transformed by the object lens. Thus, correlation filters are implemented with ease in the co-axial holography. In the case of the correlation process, an input facial image on the same position is Fourier-transformed by the same objective lens. The correlation signal emerges on the CMOS plane.

3.3 Optical correlation using holographic optical matched filter
3.3.1 Experimental results of multiplex recording and correlation

Holographic optical memory features both high density and rapid playback. The above-mentioned co-axial holography method allows for image recoding with photopolymer (Schilling, et al. (1999) (thickness: 500μm) at multiplex recording pitch (10μm) (Figure 12) (Ichikawa Watanabe & Kodate (2006). For this experiment, correlations were further examined using facial images which were recorded in the same method (Figure 13).
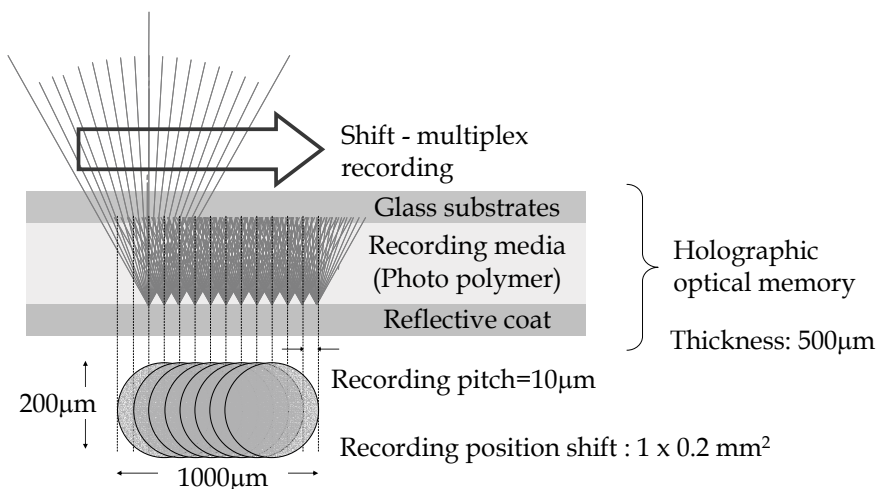
Fig. 12. Multiplex recording method
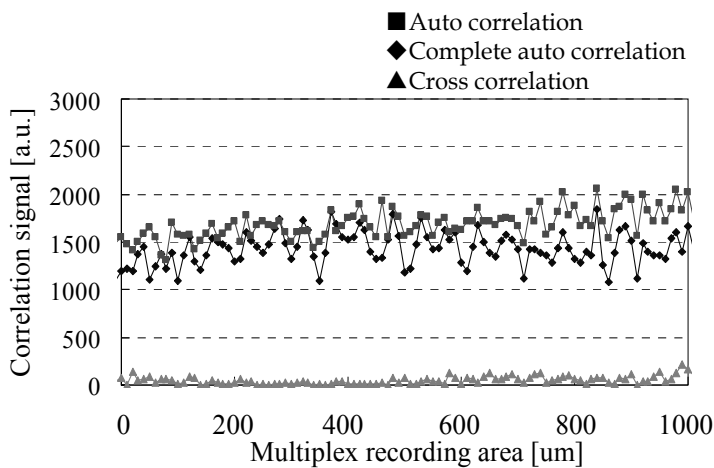
Fig. 13. Experimental results of 100 multiplex memory recording

Images shown in Figure 14 are the database and input images. Shift-multiplexing was adopted as a recoding method, while S-FARCO (wavelength: 532nm) was used as an optical set-up. The holographic optical media is composed of an AR-coated glass on the upper plane and an AL-coated glass with the photopolymer in between on the lower plane. Since the spot diameter of the laser is 200um, correlation results for 100 multiplex memory holograms can be acquired all at once on the condition that the multiplex recording pitch is 10μm. In this experiment, intensity values of correlation signals were obtained by CMOS sensor.
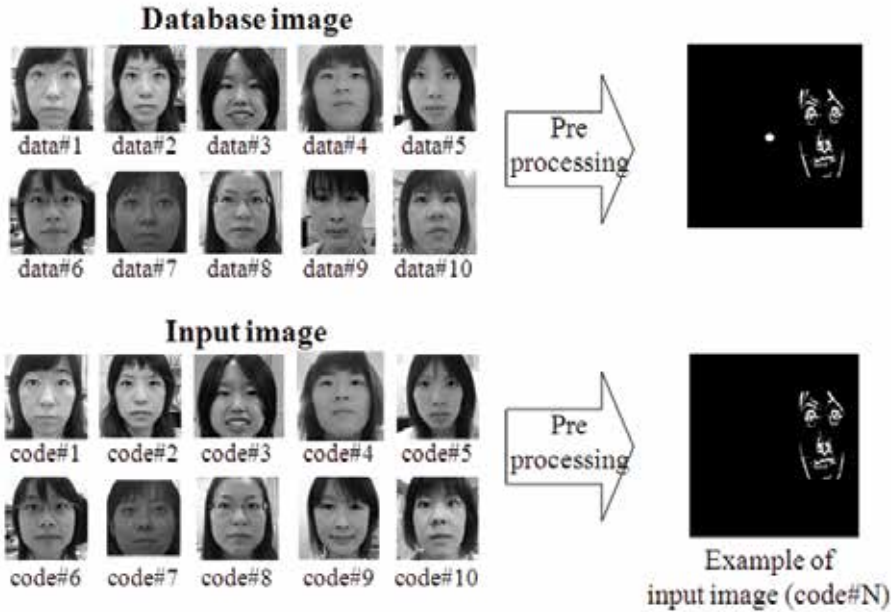
Fig. 14. Experimental samples of facial images

3.3.2 Experimental results of S-FARCO

We performed a correlation experiment under the conditions shown in Table 2. The intensities of the correlation peaks are compared with the threshold for verification. Figure 15 shows the dependences of the recognition error rates on the threshold: (a) the false-match rate, and false non- match rate and (b) the correlation between identical images. The intersection of lines (a) represents the equal error rate (EER) (when the threshold is chosen optimally), producing an EER of 0% in this experiment. An ultra high-speed system can achieve a processing speed of 5.3 μs/correlation at a multiplexing pitch of 10 micrometers and a rotational speed of 300rpm.

3.3.3 The correlation speed of a holographic optical matched filter

These preprocessed video images are recorded on a co-axial holographic optical system. The correlation speed of multiplexed recording is given by:

$$V_c = \frac{2\pi r}{d} \cdot \frac{R}{60},$$

(3)

Write Mode	Laser	Q-SW
	Rotation speed	300
	Database image	30
	Input image	30
	Recorded pitch (μm)	10
	Input image size (pixels)	64 x 128
	Hologram media (μm/cm)	400 / 12
Correlation Mode	Laser	CW
	Rotation speed (rpm)	300
	Database image	30
	Detect device	PMT

Table 2. Experimental condition for holographic optical disc correlator
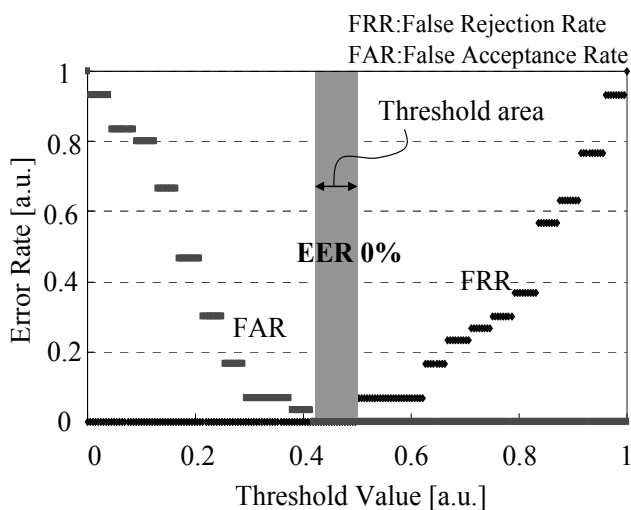
Fig. 15. Dependences of experimental recognition error rates with threshold

Multiplex recording pitch	Rotation (rpm)	Number of images for correlation per second (frames / s)	Image (320 x 240 pixels) Transfer speed (Gbps)
10μm	300	188,400	14
	600	376,800	29
	1000	628,000	48
	2000	1,256,000	96

Table 3. Correlation speed of the outermost track of an optical holographic optical disc

where r [mm], d [mm] and R [rpm] represent the diameter of the optical disc, the recording pitch and the rotating speed respectively. In a conventional correlation calculation which uses a digital computer, the data transfer and correlation calculation are achieved separately. In this system, if 240 x320 pixel information is written onto a holographic optical disc at 10 micrometer pitch and at 2,400 rpm, this is equivalent to data transfer of more than 100 G bps. An important point is that the correlation result is applied to an image of 320 x 240 bits, and the output signal of the correlation operation requires only 1.3 Mbps against the data transfer of 100 Gbps.

4. An online face recognition system

Section 4 presents an online face recognition system using the software which was constructed for the FARCO algorithm based on phase information. When FARCO software was optimized for the online environment, a low-resolution facial image size (64 x 64 pixels) was successfully implemented. An operation speed of less than 10ms was achieved using a personal computer with a CPU of 3 GHz and 2 GB memory. Furthermore, by applying eye coordinate detection in order to normalize facial images, online automatic face recognition became possible. The performance of our system was examined using 30 subjects. The experiment yielded excellent results, with low error rates, i.e. 0 % False Acceptance Rate and 0 % False Rejection Rate. Therefore, the online face recognition system proved efficient, and can be applied practically.

4.1 Application of online face recognition system

Applying the algorithm used for FARCO, a high-security online face recognition system was designed (Figure 16.). The registration process for facial images has four steps. First, an administrator informs users of the URL on which the online face recognition system is based. Then, the users access the URL. Several facial images were taken as reference images in their PCs or blogs on the internet. They were uploaded to the server together with their IDs, distributed at the time of registration in advance. Their facial images can be checked by the users themselves. A web page from an online face recognition is shown in Figure 16. (KEY images). The recognition process can be described as follows. When a facial image together with the subject's ID is inputted, the pre-processed image will be checked with the stored images in the database. The recognition result will be displayed on the webpage as in Figure 16 (Recognition result). As the system interface was designed for a web camera or surveillance camera, it can be applied widely and introduced at various places such as schools, offices and hospitals for multiple purposes.

The online face recognition system based on the algorithm for FARCO was constructed, with which a simulation was conducted (Ishikawa Watanabe & Kodate, (2007) ; Ishikawa Watanabe Ohta &Kodate, 2006). If the intensity exceeded a threshold value, the input image would be regarded as a match with a registered person. Error rates divided by the total number of cases were given by the false rejection rate (FRR) and false acceptance rate (FAR). Results demonstrated considerably low error rates: 0 % as FAR, 1.0 % as FRR and EER. However, in FARCO software, images are stored as digital data in the database such, as a hard disk drive. As a result, extra time is required for reading out data. In order to achieve high operation speeds by optical processing, it is necessary to eliminate this bottleneck.
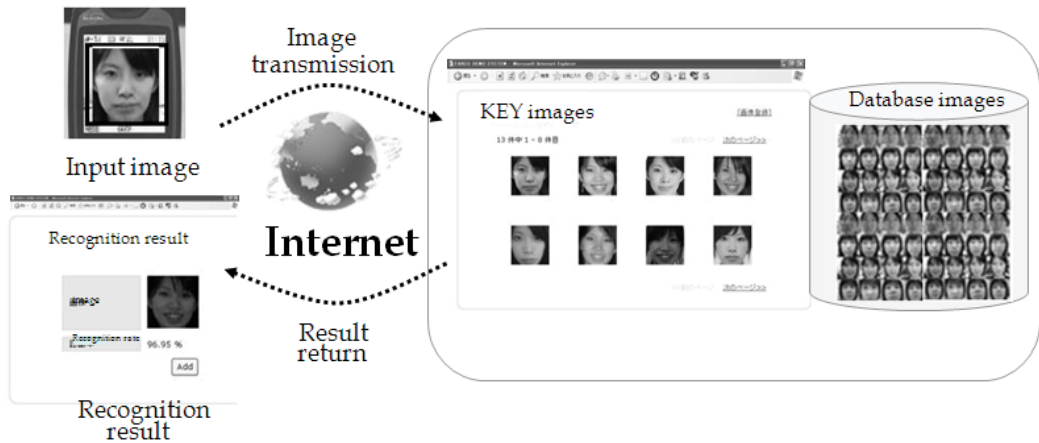
Fig. 16. Online face recognition system

4.2 Cellular phone face recognition system

Cellular phones are applied in a wide range of mobile systems, including e-mail, Internet, cameras and GPS. In this section, we propose a high-security facial recognition system with our Filtering correlation with 64x64 pixels that uses a cellular phone on a mobile network.

4.2.1 Structure of the system

A block diagram depicting the cellular phone face recognition system is shown in Figure 17. This system consists of the FARCO software for facial recognition, a control server for pre- and post-processing, and a cellular camera phone.

4.2.2 Operation of the system (Watanabe Ishikawa Ohta & Kodate, 2007)

(1) Registration

The registration process for students' facial images has four steps. Firstly, the administrator sends students the URL for i-application via e-mail. Secondly, students access the URL and download the Java application for taking input images on their own cellular phone. Thirdly, they start up the Java application and take their facial images as reference, then transmit them to the server along with their student IDs, which are issued to them beforehand. Finally, the administrator checks whether the student IDs and images in the server match, and then uploads their facial images onto the database.

(2) Recognition

The recognition process is as follows:

- Students start up the camera with the Java application and take their own facial images.
- Students transmit the image and ID (allocated at registration) back to the face image recognition server. Since the image and information are transferred on the https protocol, the privacy of the student is protected.
- In the face recognition server, the position coordinates of both eyes and nostrils are extracted from the input images. After normalization on the basis of coordinates to 128×128pixels, cutting, edge-enhancing and binarization take place.

- Subsequently, using the FARCO software, the correlation signal intensity is calculated in proportion to the resemblance of the two images.
- Using the intensity level, the system attempts to recognize the student's face based on the threshold value, which is set beforehand.
- If the student in question is recognized as a registered person, the server creates a one-time-password which will be sent with the result to the student.
- Students who acquire the password in this way can log in to the remote lecture contents server. Moreover, the face recognition server controls student registration and its database and recognition record. The Administrator can check this information through a web browser. A facial image and registration time are then recorded, which can help minimize fraud. Furthermore, images at registration can be renewed by freshly recorded images. A flow-chart of face recognition based on the Java application is shown in Figure 17.
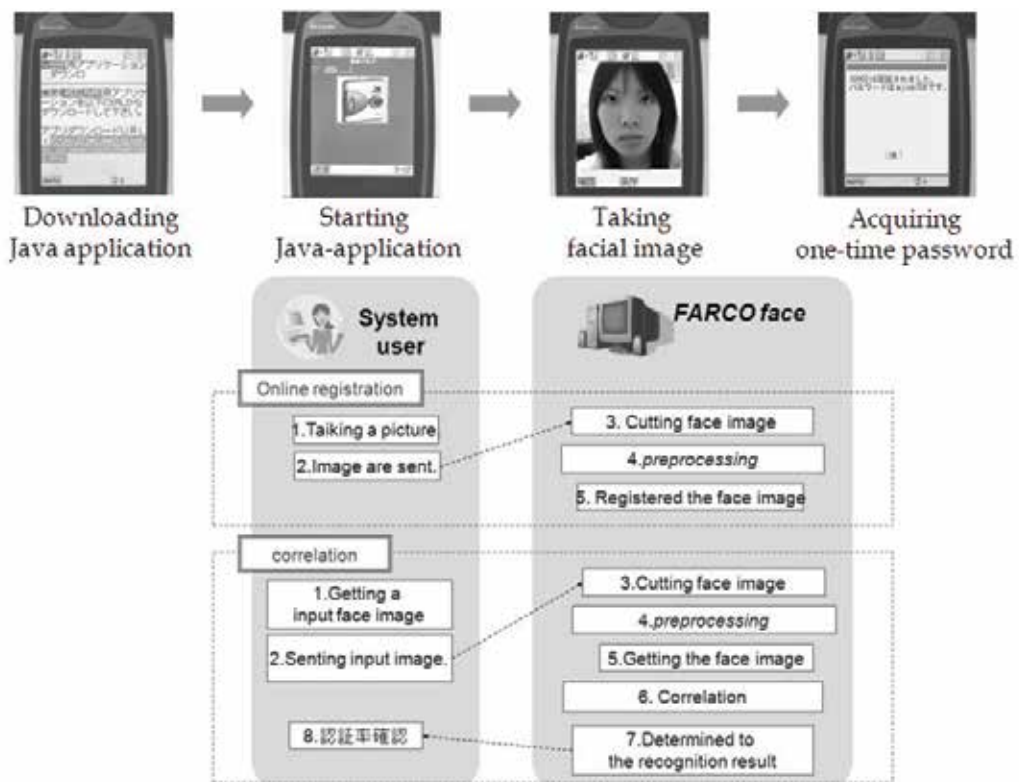
Fig. 17. A flow-chart of face recognition based on the Java application

4.2.3 Attendance management system experiment on students

Our cellular phone face recognition system was used as a lecture attendance management system, implemented 12 times on 30 students over a period of three months. The D505is and D506i (Mitsubishi Co.) were chosen from among various cellular phone types for the experiment. Students took their own facial images with the cellular phone and transmitted them to the server. Images were in the jpeg format (size 120x120pixels, 7kB).

The database images, composed of the registered 10 multiplexed images and the recognized images, were added as new database images. The experimental error rates over the duration of the three months are shown in Table 4. Results show considerably low error rates: 0 % as FAR and 2.0 % as FRR.

	First trial (%)	Second trial (%)	Third trial (%)
1st week	6.7	0	0
5th week	10.0	0	0
10th week	13.3	3.3	3.3
15th week	5.0	0	0
20th week	0	0	0
average	**9.9**	**2.9**	**2.0**

Table 4. Experimental error rates over duration of three months.

5. Various applications - video identification system -

It is widely acknowledged that current image retrieval technology is restricted to text browsing and index data searching. For unknown images and videos, the searching process can be highly complicated. As a result, the technology for this kind of image searching has not become established. In this section, we propose a video identification system using a holographic optical correlator. Taking advantage of the fast data processing capacity of FARCO, we constructed a high speed recognition system by registering the optimized video image file. Experiments on the system demonstrated that the processing speed of our holographic optical calculation is remarkably higher than that of the conventional digital signal processing architecture.

The users post the video contents to the FARCO server by the web interface as shown in Figure 18 (a). The video contents on the FARCO server are preprocessed (i.e. normalization, color information and other feature extraction) and transferred as binary data. These binary data are recorded in the form of matched filtering patterns.

With the explosion of use of video-sharing site, there is a high demand for a recognition system for moving images, working at high speed and with high accuracy. So far, the technology is restricted to text search through the tags attached to those motion images. This type of index search has weaknesses such as the difficulty in pinning down the actual content and specifying scenes, as well as the costs of creating tags for each item. In order to overcome the problems posed by these characteristics, several techniques are being actively researched and proposed. However, the ambiguity of labeling images and the sheer variety pose a number of challenges for this complex issue of differentiating the images. So far, we have developed a Fast Recognition Correlator system (FARCO) using the speed and parallelism of light. FARCO has been tested rigorously and proved its high performance. Combining this with the promising nature of the holographic optical disc, which was described above, we have proposed an all-optical ultra-fast image search engine system. The

section below presents our newly developed FARCO video system with which motion images can be distinguished using our techniques applied to our face recognition.

5.1 Basic structure of the moving image recognition system and its algorithm

FARCO video system enables moving image search which is on the video-sharing site, identifying those images registered on the server.

5.1.1 Registration for moving images

- Users upload moving images that need to be singled out from the Web interface in the FARCO video.
- Those uploaded images are preprocessed, i.e. the images are frame-compressed, the color information is extracted and binarized.
- The data will be stored as basic information about the images.

5.1.2 Recognition for moving images

The recognition process is as follows:

- Users execute recognition of motion images on the web interface in FARCO video.
- By keyword search, input images have to be pinned down from the video-sharing site, and downloaded. Currently, twenty video-sharing sites are included in our data search system.
- By making the resolution level variable, quality adjustment becomes possible. Input data are preprocessed, prior to cross-comparison with registered moving images.
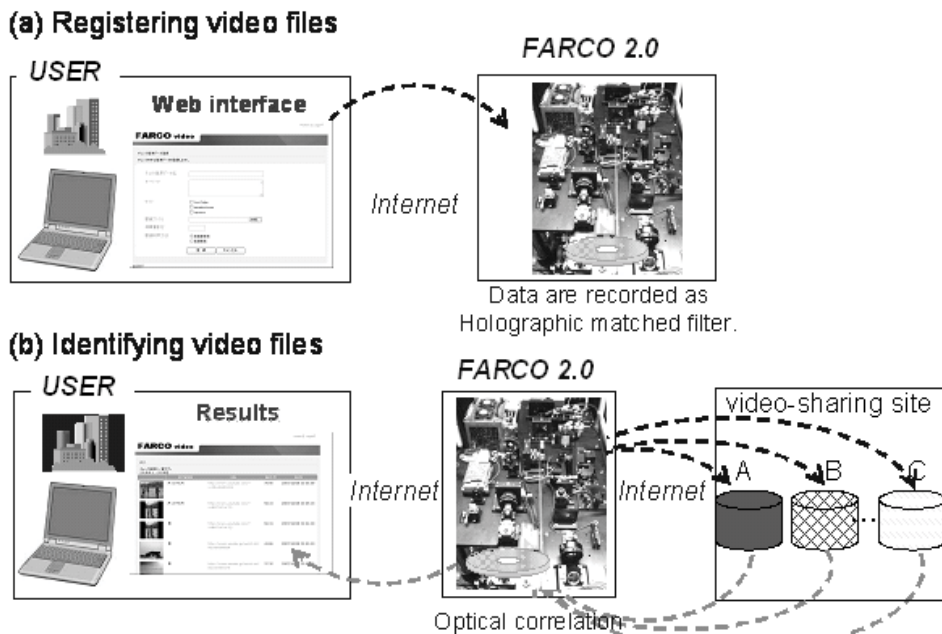
Fig. 18. The concept of video filtering system. (a) Registering video files, (b) Identifying video files

5.2 Experimental results

In our experimental system, each image file taken from DVD is registered as a video file, while the input video image file is downloaded from video sharing sites. We performed a correlation experiment using a co-axial holographic optical memory system. The example registered video image files are shown in Figure 19 (a). The intensities of the correlation peaks are compared with the threshold for verification. Figure 19. shows the dependence of the recognition error rates on the threshold: (a) false-match rate and false non-match rate and (b) the correlation between identical images. The intersection of lines (a) represents the equal error rate (EER) (when the threshold is chosen optimally), and in this experiment an EER of 0% was achieved. This ultra high-speed system can achieve a processing speed of 25 microseconds/correlation at a multiplexing pitch of 10 micron and rotational speed of 300rpm.

(a)   (b)
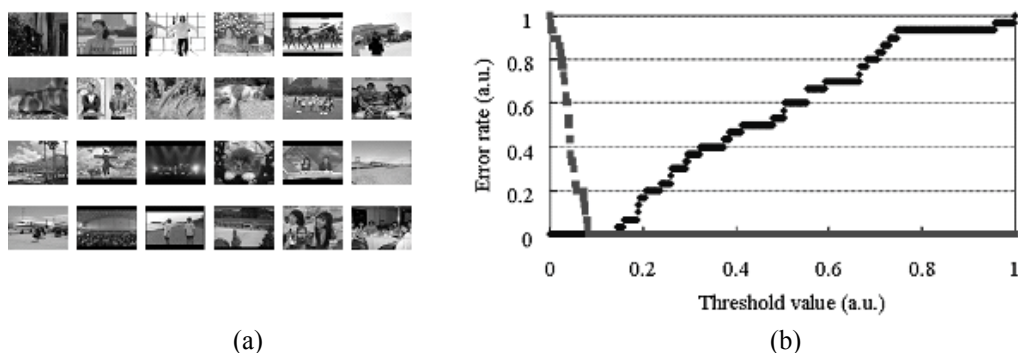
Fig. 19. (a) Frame image examples (b) Experimental results using holographic optical matched filter

6. Conclusions

We presented an ultra high-speed optical correlation system for face recognition (S-FARCO) using holographic optical memory. By means of preliminary correlation experiments using the holographic optical disc set-up demonstration, we acquired low error rates, e.g. 0% Equal Error Rate. These are the world's first experimental results for an ultra high speed correlation system using a holographic optical disc. The S-FARCO is potentially 1000 times faster than FARCO software. We also constructed and evaluated the software correlation filter covering several hundred volunteers, demonstrating that the system is highly accurate, as facial images with low resolution (64x64 pixels) have been used successfully. Using a CPU with 3 GHz and 2 GB memory, an operation speed of less than 10ms was achieved. We obtained highly accurate experimental results and low error rates, i.e. 0 % FAR and 2.0 % FRR, using a high-security cellular phone face recognition system with our Filtering correlation. Even if the size of the image is small, an accurate result can be obtained using Filtering correlation. Therefore, Filtering correlation works effectively with web applications and face recognition using a monitoring camera. We have proposed a holographic optical video filtering system using a holographic optical correlator. Taking advantage of the fast data processing capacity of S-FARCO, we explored the possibility of realizing a high-speed

recognition system by registering the optimized video image file. The results demonstrated that the processing speed of our holographic optical calculation was remarkably higher compared to the conventional digital signal processing architecture.

7. Acknowledgments

8. References

Kaneko, M. & Hasegawa, O. (1999). Processing of Face Images and Its Applications. *IEICE Trans. Inf., Syst,* E82-D, (1999) pp.589.

Kanade, T. (1971). Dr. Engineering, *Faculty of Engineering, University of Kyoto, Kyoto*, 1971).

Sirovich, L. & Kirby, M. (1991). Low-dimensional procedure for the characterization of human face. *J. Opt. Soc. Am.,* Vol.4, No.3, (1987). pp.519-524.

Savvides, M. et al. (2004). Eigen phases vs. Eigen faces. *Proc. ICPR.*, Vol.3, (2004) pp.810-813.

Goodman, J. & Moeller, M. (2004). Introduction to Fourier Optics Roberts & Company Publishers, Colorado, Vol.1, Chap.8, (2004) p.237.

Thapliya, R. & Kamiya, T. (2000). Optimization of Multichannel Parallel Joint Transform Correlator for Accelerated Pattern Recognition. *Appl. Opt.,* Vol.39, No.29, (2000) pp.5309-5317.

Inaba, R. Watanabe, E. & Kodate, K. (2003). Security Applications of Optical Face recognition System: Access Control in E-Learning. *Opt. Rev.,* Vol.10, No.4, (2003) p.255-261.

Kodate, K. Inaba, R. Watanabe, E. & Kamiya, T. (2002). Facial Recognition by Compact parallel Optical Correlator. *Measurement Science and Technology*. Vol.13, No.11, 2002) pp.1756-1766.

Kobayashi, Y. & Toyoda, Y. (1999). Development of an optical joint transform correlation system for finger- print recognition. *Opt. Eng.,* No.38, (1999) pp.1205-1210.

Carrott, T. David Gary Mallaley, R. Dydyk B. & Mills, A. (1998). Stuart Third generation miniature ruggedized optical correlator (MROC) module. Proc. SPIE, Vol.3386 (1998) pp.38-44.

(a) Watanabe, E. & Kodate, K. (2005). Implementation of high-speed face recognition system using an optical parallel correlator. *Appl.Opt.,* Vol.44, No.6, (2005) p.666-676.

(b) Watanabe, E. & Kodate, K. (2005). Face-Recognition Optical Parallel Correlator Using High accuracy Correlation Filter. Opt.Rev., Vol.12, No.6, (2005) pp.460-465.

Horimai, H. & Tan, X. (2005). Collinear holography. *Appl. Opt.*, Vol. 44, No.13, (2005) pp. 2575-2579.

(a) Watanabe, E. & Kodate, K. (2006). Optical Correlator for Face Recognition Using Collinear holographic System. Jpn. J. Appl. Phys. Vol.45, No.8B, (2006) pp.6759-6761.

(b) Watanabe, E. & Kodate, K. (2006) . High speed image search engine using collinear holography. Proc. SPIE, Vol.6245 (2006) pp.147-154.

Horimai, H. Tan, X. & Li, J. (2006). Collinear technology for a holographic versatile disk. *ppl. Opt.*, Vol.45, No.5, (2006) pp.910-914.

Mansfield, J. A. & Wayman , L. J. (2002). Biometric Testing Best Practices Version 2.01 National Physical Laboratory, Teddington, 2002).

L Horner, J. Gianino, P. D. (1982) Phase-only matched filtering. *Appl. Opt.*, Vol.23, No.6, (1982) p.812-816.

(c)Watanabe, E. & Kodate, K. (2005). Fast Face-Recognition Optical Parallel Correlator Using High Accuracy Correlation Filter. *Opt.Rev.*, Vol.12, No.6, (2005) pp. 460-466.

Watanabe, E. & Kodate, K. (2005). Fast Face-Recognition Optical Parallel correlator Using High Accuracy Correlation Filter. *Opt.Rev.*, Vol.12, No.6, 2005.8) pp. 460-466.

Horner, J. L. & Gianino, P. D. (1984). Phase-only matched filtering. Appl. Opt., Vol.23, No.6, 1984) pp.812-816.

Tarres (web), F. & Rama, A. GTAV Face Database. available at. (http://gps-tsc.upc.es/GTAV/ResearchAreas/UPCFaceDatabase/GTAVFaceDatabase.html).

Goto, K. (2004). Near-Field Optical Storage Technology. *The Review of laser engineering,* Vol.32, No.1, (2004) pp.22-28 [in Japanese].

Tominaga, J. et al. (2002). Optical data storage with super-resolution near-field structure. *Appl. phys.*, vol.71, No.6, (2002) pp.700 -704 [in Japanese].

Kawata, Y. & Nakano, M. (2005). New Development of Multilayered Optical Memory for Terabyte Data Storage. *IEEE Trans. Magn.*, Vol.41, No.2, (2005) pp.997-1000.

Coufal, J. H. Psaltis, D. & Sincerbox, T. G. (2000). Holographic Data Storage. *eds* (2000), pp.259-269.

Van Heerden, J. P. (1963). Theory of optical information. storage in solids. Appl. Opt., Vol.2, (1963) pp.393-400.

Hesselink, L. (2000). Ultra high-Density data storage. *Communication of the ACM*, vol.43, No.11, (2000) pp.33-36.

Orlov, Sergei S. (2000). Volume Holographic Data Strage. *Communication of the ACM,* vol.43, No.11, (2000) pp.46-54..

Schilling, L. M. et al. (1999). Acrylate oligomer-based photopolymers for optical storage applications. *Chem. Mater.,* No.11, (1999) pp.247-253.

Sato, A. et al. (2006). Photopolymer system for holographic recording and application for hologram memory. *HODIC Circular*, Vol.26, No.1, (2006) pp.14-19 [in Japanese].

Ichikawa, Y. Watanabe, E. & Kodate, K. (2006). *Proc. MOC2006*, (2006) pp.156-157.

Ishikawa, S. Watanabe, E. & Kodate, K. (2007). Online face recognition system using a highly accurate filtering correlation. *Proc. MOC 2007*, (2007) pp.130-131.

Ishikawa, S. Watanabe, E. Ohta, M.& Kodate, K. (2006). Highly-accurate face recognition using a novel filtering correlation. *Proc. ODF*, (2006) pp.305-306.

Watanabe, E. Ishikawa, S. Ohta M. & Kodate, K. (2007). Cellular phone face recognition system based on optical phase correlation. *IEEJ Trans. EIS,* Vol.127, No.4, (2007) pp.636-643 [in Japanese].

3D Face Mesh Modeling for 3D Face Recognition

Ansari A-Nasser[1], Mahoor Mohammad[2] and Abdel-Mottaleb Mohamed[3]

[1]Ministry of Defence, QENF, Doha, Qatar
[2]University of Denver, Denver, Colorado
[3]University of Miami, Coral Gables, Florida,
[1]Qatar
[2,3]U.S.A

1. Introduction

Face recognition has rapidly emerged as an important area of research within many scientific and engineering disciplines. It has attracted research institutes, commercial industries, and numerous government agencies. This fact is evident by the existence of large number of face recognition conferences such as the International Conference on Automatic Face and Gesture and the Biometric Consortium conference. Special issues of well known journals, are being dedicated to face modeling and recognition, such as the journal of Computer Vision and Image Understanding (CVIU), and the systematic empirical evaluations of face recognition techniques including the FERET (Phillips et al., 2000), XM2VTS (Messer et al., 1999), FRVT 2000 (Blackburn et al., 2000), FRVT 2002 (Phillips et al., 2002), and FRVT 2006, which evolved substantially in the last few years. There are few reasons for this trend; first the demands for machine automations, securities, and law enforcements have created a wide range of commercial applications. The second is the availability of feasible technologies developed by researchers in the areas of image processing, pattern recognition, neural network, computer vision, and computer graphics. Another reason for this growing interest is to help us better understand ourselves through the fields of psychology and cognitive science which targeted the perception of faces in the brain. Because our natural face recognition abilities are very powerful, the study of the brain system could offer important guidance in the development of automatic face recognition. Research with animals has shown that these capabilities are not unique to humans. Sheep, for example, are known to have a remarkable memory for faces (Kendrick et al., 2000). In addition, we constantly use our faces while interacting with each others in a conversation. Face gesturing helps us understand what is being said. Facial expression is an important cue in understanding a person's emotional state. In sign languages, faces also convey meanings that are essential part of the language.

A wealth of 2D image-based algorithms has been published in the last few decades (Zhaho et al., 2003). Due to the numerous limitations of 2D approaches, 3D range image-based algorithms are born. Generally, 3D facial range image or data is rich, yet making full use of its high resolution for face recognition is very challenging. It is difficult to extract

numerously reliable facial features in 3D. As a result, it becomes more challenging and computationally expensive to accurately match two sets of 3D data (e.g., matching a subject's probe data with the gallery's). Our objective in this chapter is to illustrate a model-based approach that represents the 3D facial data of a given subject by a deformed 3D mesh model useful for face recognition application (Ansari, 2007). The general block diagram of the system is shown in Fig.1, which consists of the modeling stage and the recognition stage. In the modeling stage, only three facial feature points are first extracted from the range image and then used to align the 3D generic face model to the entire range data of a given subject's face. Then each aligned triangle of the mesh model, with three vertices, is treated as a surface plane which is then fitted (deformed) to its corresponding interior 3D range data, using least squares plane fitting. Via triangular vertices subdivisions, a higher resolution model is generated from the coordinates of the aligned and fitted model. Finally the model and its triangular surfaces are fitted once again resulting in a smoother mesh model that resembles and captures the surface characteristic of the face. In the recognition stage, a 3D probe face is similarly modeled and compared to all faces in the database. Experimental application of the final deformed model in 3D face recognition, using a publicly available database, demonstrates promising recognition rates.
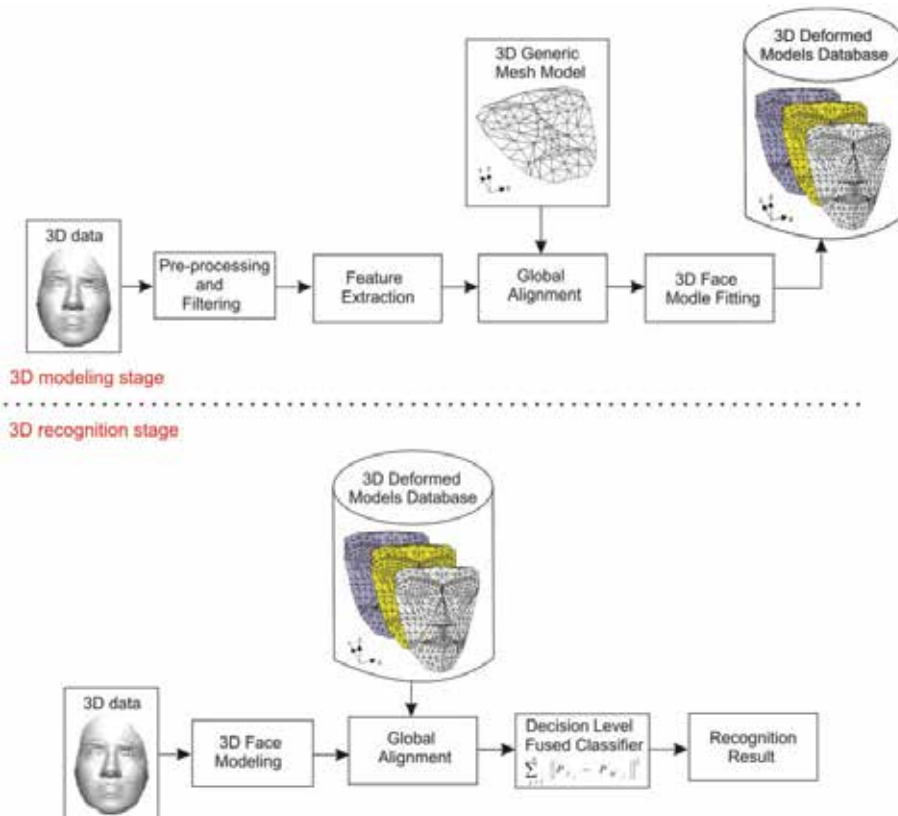
Fig. 1. 3D face modeling and recognition system.

This chapter is organized as follow: Section 2 explains the limitations and challenges of face recognition. Section 3 covers a review of related work. Section 4 describes the data pre-

processing and facial features extraction. Section 5 illustrates the process of 3D face modeling. Section 6 demonstrates experimental results. Finally, conclusion and discussion are given in section 7.

2. Limitations and challenges

Despite the great potentials and the significant advances in face recognition technology, it is still not robust and prone to high error rates, especially in unconstrained environments and in large scale applications. The process of identifying a person from facial appearance has to be performed in the presence of many often conflicting factors which alter the facial appearance and make the task difficult. In Table 1, we categorize the variations in facial appearance into two types: intrinsic and extrinsic sources of variation.

Variation in appearance	sources	Effects / possible task
Extrinsic	Viewing geometry Illumination Imaging process Other objects	Head Pose light variations, shadow, self shadow Resolution, scale, focus, sampling Occlusion, shadowing, indirect illumination, hair, make-up, surgery
Intrinsic	Identity Facial expression Age Sex speech	Identification, known-unknown Inference of emotion or intension Estimating age Decide if male or female Lip reading

Table 1. Extrinsic and intrinsic variations in facial appearance.

The intrinsic variations take place independently of any observer (camera) and are due purely to the physical nature of the face. The identity source is an important intrinsic variation in identifying people from one another, yet problems arise when combined with aging or facial expression because they are difficult to characterize analytically. The extrinsic sources of pose variations, due to the relative position of the camera, and illuminations present a major challenge in face recognition. Recognition systems are highly sensitive to the light conditions and circumstances under which the images being compared are captured. These lighting conditions can be due to the environment or the physical characteristics of the image capturing device, i.e., two cameras of the same brand may give different exposures. The pose of the face is determined by the relative three dimensional position and orientation of the capturing device. Usually, two face images of the same subject taken at different poses are more different than two images of two subjects taken at the same pose. While change in pose is considered a rigid 3D motion, a face can also undergo non-rigid motion when its 3D shape changes due to speech or facial expression. It is very difficult to model both types of motion at the same time. All these factors and conditions make the images used for training the recognition system different from the images obtained for recognition. If these factors are not incorporated and modeled properly, they dramatically degrade the accuracy and performance of a recognition system.

Another challenge is the need for an evaluation standard for measuring recognition performance under different environments and conditions. As a result of this necessity, an

independent government evaluation standard was initiated, called Face Recognition Vendor Tests (FRVT) (Blackburn et al., 2000). FRVT was developed to provide evaluations of commercially available and prototype face recognition technologies. These evaluations are designed to provide the U.S. government and law enforcement agencies with information to assist them in determining where and how facial recognition technology can best be deployed. In addition, FRVT results help identify future research directions for the face recognition community. In the past, many factors have been evaluated in FRVT 2002 (Phillips et al., 2003).

A recent challenge to face recognition systems is the concern about possible privacy violations. For example, the American Civil Liberties Union (ACLU) opposes the use of face recognition systems at airports due to false identification and privacy concerns. The ACLU claims that face recognition technology poses the danger of evolving into a widespread tool for spying on citizens as they move about in public places.

3. Related work

Semi-automated facial recognition system dates way back to 1965. (Chan & Bledos, 1965) showed that a computer program provided with facial features extracted manually could perform recognition with satisfactory performance. In the past few years, face recognition has received great attentions. A literature survey of face recognition is given in (Zhaho et al., 2003), where most of the paper surveys 2D algorithms. In addition, the work and survey by (Bowyer et al., 2004) compare face recognition techniques based on 2D data, 3D data, and 2D+3D data fusion (also refered to as multimodal). They reported that 3D face recognition approaches outperform 2D approaches and the fusion of 2D + 3D data produces slightly better results than 3D alone. Recently, a survey by (Boyer et al., 2006) cited some algorithms with 2D recognition approaches outperforming the 3D approaches. There is a belief that it is still premature to make this judgment because current approaches do not yet make full use of 3D data either in the recognition algorithms or the rigorous experimental methodology. In this chapter, we only review relevent 3D algorithms processed on range images (3D data) alone.

We can broadly classify 3D face recognition into three categories, namely, 3D surface matching, representative domain, and model-based approaches. A surface matching method, known as Iteratively Closest Point (ICP) approach (Besl & McKay, 1992), is often used as a necessary step in aligning or matching the datasets of two subjects (Lu & Jain, 2006). ICP is based on the search of pairs of nearest points in the two datasets and estimation of the rigid transformation that aligns them. Then the rigid transformation is applied to the points of one set and the procedure is iterated until convergence. Hausdorff distance is another matching approach which is often used in conjunction with ICP. Hausdorff distance attempts to match two datasets based on subset points from the datasets (Huttenlocher et al., 1993); (Russ et al., 2005). The problems with these two matching approaches are expensive computations and sometimes fail to give accurate results. The main reason for using ICP or Hausdorff is not having direct correspondences between the two compared datasets. In the presented algorithm of this chapter, the two compared datasets have direct feature correspondences, which eliminate the need for the above alignment/matching algorithms. (Medioni & Waupotitsch, 2003) present an authentication system that acquires the 3D image of the subject using stereo images based on internally and externally calibrated cameras. They use the ICP algorithm to calculate similarity between two faces achieving

98% on a database of 100 subjects. (Lu et al., 2004) filter and stitch five multiple views of 2.5D facial scan of each subject to obtain a more complete 3D facial scan. The complete 3D facial scan model is used in the gallery for recognition and the partial 2.5D scans are used as probes. Matching is performed using ICP between a 3D scanned test face with the faces in the database. A 96% recognition rate is obtained with a database of 19 subjects. (Lu & Jain, 2005) extended their previous work using an ICP-based recognition approach by (Russ et al., 2004) to deal explicitly with variations due to the smiling expression. In their experiments, they used a 100-person dataset, with neutral-expression and smiling probes, matched to neutral-expression gallery images. The gallery entries were whole-head 3D models, whereas the probes were 2.5D scan frontal views. They report that most of the errors are after the rigid transformation resulted from smiling probes, and these errors are reduced substantially after the non-rigid deformation stage. For the total of 196 probes (98 neutral and 98 smiling), performance reached 89%. (Uchida et al., 2005) propose two sets of a passive stereo system using four cameras to capture facial images. One set contains two cameras with short baseline intended for accurate correspondence matching. The other two cameras are separated with wide baseline for accurate 3D reconstruction. ICP matching is used between the probe and the gallery faces of a database of 18 subjects each with four simultaneous images. Unfortunately, no recognition rate was reported. (Chang et al., 2005) present an Adaptive Rigid Multi-region Selection (ARMS) approach to independently match multiple facial regions and create a fused result. The ARMS is a classifier type approach in which multiple overlapping sub-regions (e.g., areas around the nose) are independently matched by ICP. Then, the results of the multiple 3D matching are fused. Their experiments on FRGC version 2.0 database resulted in a 91.9 % rank-one recognition rate for automatic Regions of Interest (ROIs) finding and 92.3 % rank-one recognition rate for manual ROIs finding. (Achermann & Bunke, 2000) used two range scanners to capture ranges image in order to overcome the holes and missing data that might result from using one scanner. In addition, they used an extension of 3D Hausdorff distance for 3D face matching. Using 10 images per each of the 24 subjects, they reported 100% recognition rate. (Lee & Shim, 2004) incorporate depth information with local facial features in 3D recognition using Hausdorff distance weighted by a function based on depth values. The weights have different values at important facial features such as the nose, eyes, mouth, and face contour. They achieved rank five recognition rate of 98%. (Russ et al., 2004) use Hausdorff distance matching for range images. In a verification experiment for 200 subjects enrolled in the gallery and the same 200 persons plus an additional 68 in the probe set, they report a verification rate of 98%. In a recognition experiment, 30 persons enrolled in the gallery and the same 30 persons imaged at a later time were used in the probe set. A 50% recognition rate is achieved at a false alarm rate of 0.

Other researchers attempted to represent the 3D data in a different domain and made recognition comparison in the representative domain. Examples of those are 3D Principle Component Analysis (PCA) (Hesher et al., 2003), shape index (Lu et al., 2006), point signature (Chua et al., 2000), spine image (Johnson & Hebert, 1999), and local shape map (Wu et al., 2004). PCA is a statistical approach commonly used in recognition. One reason for using PCA is to reduce the dimensionality of the data, while sacrificing the performance of the recognition algorithm. (Hesher et al., 2003) explore PCA techniques using different number of eigenvectors, image sizes, and different expressions. They report a high recognition rate, but their system degrades if the expression of a test face is different from

the expressions in the database. (Xu et al., 2004a) slightly improved the recognition rate by computing a feature vector from the data in the local regions of the mouth, nose, left eye, and right eye. The dimensionality of the feature vector is reduced with PCA and matching is based on minimum Euclidean distance. Experiments on 120 subjects in the dataset resulted in 72% recognition rate, and on a subset of 30 subjects resulted in a 96% recognition rate. It should be remarked that the reported performance was obtained with five images of a person used for enrollment in the gallery. Performance is generally expected to be higher with more images used to enroll a person. (Pan et al., 2005) apply PCA to the range images using a novel mapping technique. Finding the nose tip to use as a center point, and an axis of symmetry to use for alignment, the face data are mapped to a circular range image. Experimental results are reported for the FRGC version 1.0 data set with 95% rank-one recognition rate and 2.8% Equal Error Rate (EER). Another example of a representative domain approach is the use of transform or wavelet. (Cook et al., 2006) present an approach based on Log-Gabor template for providing insensitivity to expression variation in range images. They decompose the facial image into overlapping 147 sub-jets (49 sub-regions and three scales) using Log-Gabor wavelets. For face verification, they use the Mahalanobis cosine distance measure and un-weighted summation to combine the result of classifying each region. Their experiments resulted in a 92.3 % rank-one recognition rate.

Model-based approaches use a priori facial model such as graph or mesh model. Graph representation has shown to be successful (Wiskott et al., 1997); (Blome, 2003). The idea is to use a graph to model the face with nodes and edges. The edges are labeled with distance information and nodes are labeled with local wavelet responses. However, the graph models in the literature have some limitations. For example, there is no justification for defining the edges of the graph. (Mahoor et al., 2008) improved a graph model which they refer to as Attributed Relational Graphs (ARG). The ARG is a geometric graph also with nodes and edges, where the nodes represent the facial landmarks and the edges connects the nodes based on Delaunay triangulation. A set of mutual relations between the sides of the triangles are defined in the model and are used in the recognition process in addition to the nodes and edges.

Mesh model approaches use a priori defined facial mesh which is usually morphed of deformed to a given face. A detailed example of this approach is illustrated in this chapter, which has the advantages of eliminating some of the previously stated problems of both the surface matching and the representative domain algorithms. Firstly, by representing the huge facial range data by a mesh model with smaller number of vertices, we reduce the amount of data points for facial processing, data storage, and recognition comparisons. Secondly, having a predefined and labeled-vertices in the deformed mesh model, establishes direct features correspondences between compared probe's and gallery's facial data. Hence faster recognition comparisons are achieved. Both the labeling of the model's vertices and the data reduction, resulting from representing the face by the vertices of the model, are vital in reducing the complexity of the face recognition system. The presented method in this chapter is similar to work of (Xu et al., 2004b) but differs in the followings: (a) The method in this chapter uses a generic face mesh model and (Xu et al., 2004b) use a general mesh grid model, (b) here, the aligned model's mesh triangles coordinate are deformed to the data and (Xu et al., 2004b) simply align the grid mesh coordinates to the range data then copy the z coordinate at each x and y coordinates, hence in their way the pose of the z coordinate is not considered, (c) the presented system establishes direct correspondences

with other models in the database, hence direct comparison is achieved in recognition, while the method of (Xu et al., 2004b) has no correspondences and would require facial surface alignment and matching.

(Vetter & Blanz, 1999) proposed a face recognition algorithm based on computer graphics techniques, where they synthesize the 3D model of a face from a single 2D image of known orientation and illumination. However, their algorithm is computationally expensive and initially requires manual user assistance and a database of 200 different real scans of faces obtained from a 3D scanner. Correspondences across these 3D scans are pre-computed. The input face image is estimated as a linear combination of the projected 3D scans in the database; subsequently, the output 3D model is a linear combination of the 3D scans. Similar approach is proposed by (Jiang et al., 2004) which they referred to as analysis-by-synthesis 2D to 3D face reconstruction, in which they use a single frontal 2D image of the face with a database of 100 3D faces captured by 3D scanner. In this approach frontal face detection and alignment are utilized to locate a frontal face and the facial feature points within an image, such as the contour points of the face, left and right eyes, mouth, and nose. Then, the 3D face shape is reconstructed according to the feature points and a 3D face database. Next, the face model is textured-mapped by projecting the input 2D onto the 3D face shape. Finally, based on the resulting 3D model, virtual samples of 3D models are synthesized with pose and expression variations and are projected to 2D for recognition. (Hsu & Jain, 2001) adapts a generic face model to the facial features extracted from both registered range and color images. The deformation iteratively moves the vertices of the mesh model using vertices displacement propagation. (Ansari & Abdel-Mottaleb., 2005) deformed a generic model to few 3D facial features obtained from one frontal and one profile view calibrated stereo images. The additional profile view complements and provides additional information not available in the frontal view. For 29 subjects, a recognition rate of 96.2 % is reported. In (Ansari et al., 2006) an improved modeling and recognition accuracy is presenting using dense range data obtained from two frontal and one profile view stereo images for 50 subjects attaining 98% recognition rate.

4. Data pre-processing and facial features extraction

This section explains the pre-processing of the data, localization of the facial region, and the facial features extraction. Further details are given in (Mahoor et al., 2007). Range images, captured by laser scanners, have some artifacts, noise, and gaps. In the pre-processing step, we first apply median filtering to remove sharp spikes and noise, that occur during the scanning of the face, followed by interpolation to fill up the gaps, and low pass filtering to smooth the final surface. This is followed by face localization using facial template matching to discard the neck, hair, and the background areas of the range image. The facial range image template is correlated with the range images of a given face using normalized cross-correlation. We start by roughly detecting the location of the nose tip and then translate the template such that the detected tip of the nose is placed on the location of the nose tip of the range image under test. Afterward, we iteratively apply a rigid transformation to the template and cross-correlate the result with the subject's range image to find the best pose. Finally, the area underneath the template with the maximum correlation is considered as the localized facial region. Subsequently, we use Gaussian curvature to extract the two inner corners of the eyes and the tip of the nose. The surface that either has a peak or a pit shape has a positive Gaussian curvature value $K > 0$ (Dorai & Jain, 1997). Each of the two inner

corners of the eyes has a pit surface type and the tip of the nose has a peak surface type that is detectable based on the Gaussian curvature. These points have the highest positive Gaussian curvature values among the points on the face surface. Fig.2.a shows the result of calculating the Gaussian curvature for one of the sample range images in the gallery.
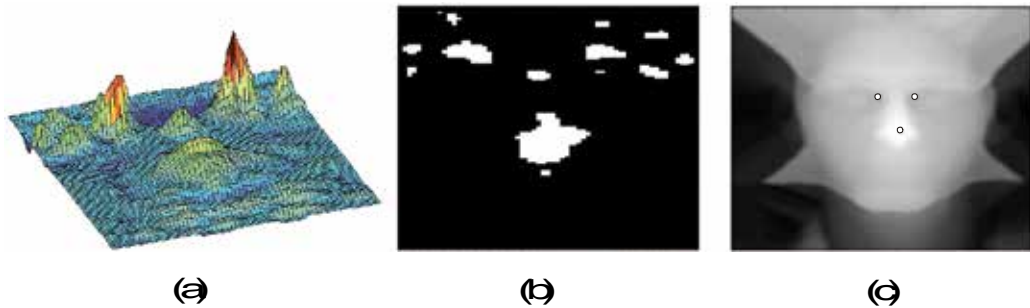

(a)   (b)   (c)

Fig. 2. Features extraction process (a) Gaussian curvature showing high values at the nose tip and eyes corners (b) Result of thresholding Fig.2.a (c) Final result of feature extraction.

The highest points in Fig.2.a correspond to the points with pit/peak shape. We threshold the Gaussian curvature to find the areas that have positive values greater than a threshold, producing a binary image. See Fig.2.b. The threshold is calculated based on a small training data set different from the images used in the recognition experiments. Finally, the three regions with the largest average value of the Gaussian curvature are the candidate regions that include the feature points. The locations of the points with maximum Gaussian curvature in these regions are labeled as feature points. Fig.2.c shows a final result of the three feature extraction points. These features are used in the 3D model alignment as we show next.

5. 3D face modeling

This section deals with modeling the human face using its extracted features and a generic 3D mesh model. The idea is to align the 3D model to a given face using the extracted 3D features then proceed with fitting the aligned triangles of the mesh to the range data, using least square plane fitting. Next, the aligned triangles of the model are subdivided to higher resolution triangles, before applying a second round of plane fitting, to obtain a more realistic and a smoother fitted surface resembling the actual surface of the face. Fig.3.a shows our neutral 3D model with a total of 109 labeled feature vertices and 188 defined polygonal meshes. In addition, the model is designed such that the left and right sides of the jaw fall within but not on the edges of the face boundary. This approach avoids incorporating inaccurate data at the facial edges of the captured range images. We explain next the process of aligning the mesh model to the range data.

5.1 Global alignment
In the global alignment step, we rigidly align the 3D model using the three 3D feature points, P_I, obtained from the range image, and their corresponding feature vertices, P_M, in the model. Subscripts $_I$ and $_M$ indicate image features and model vertices, respectively. To achieve this goal, the model must be rotated, translated, and scaled. Eq.1 gives the sum squared error between P_I and P_M in terms of scale S, rotation R, and translation T for $n = 3$ points.

$$Min \quad E(S, R, T) = \sum_{j=1}^{n} \left\| P_{I_j} - P_{M_j} \right\|^2 \tag{1}$$

An example of the aligned 3D model to the range data is demonstrated in Fig.3.b and Fig.3.c for 2D view and 3D view, respectively. As shown in the figures, the triangles of the model are buried either totally or partially above or below the 3D data. We show next how to segment the 3D data points within the aligned 3D model.
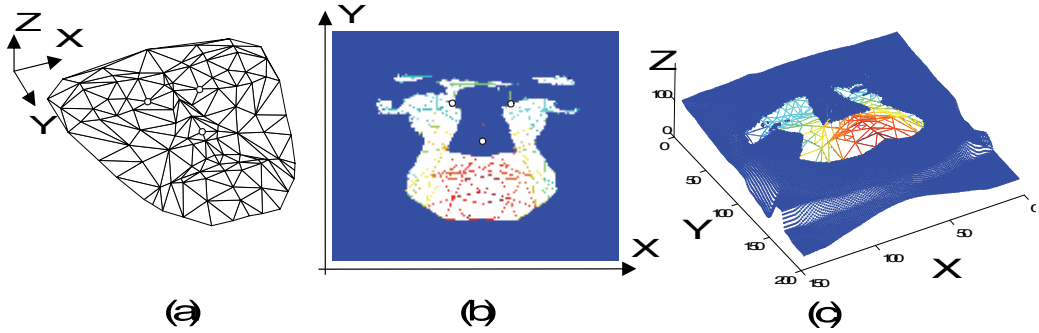

Fig. 3. (a) 3D mesh model (b) 2D view of aligned 3D model vertices to the range data (c) 3D view of model to the range data.

5.2 3D facial points segmentation

The first step prior to deforming the model is to segment and extract the 3D data points facing (above, below, or within) each mesh triangle using a computer graphic technique referred to as *Barycentric Coordinate* (Coxeter, 1969). A barycentric combination of three point vertices P_1, P_2, and P_3, forming a triangular plane is shown in Fig.4
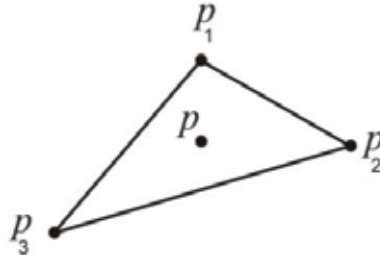
Fig. 4. The barycentric coordinates of a point p with respect to the triangle vertices.

The coordinate of point p inside the triangle is defined by

$$p = up_1 + vp_2 + wp_3 \quad \text{where} \quad u + v + w = 1 \tag{2}$$

Therefore, p lies inside the triangle and we say $[u, v, w]$ are the barycentric coordinates of p with respect to p_1, p_2, and p_3 respectively. Equivalently, we may write

$$p = up_1 + vp_2 + (1 - u - v)p_3 \tag{3}$$

Eq.3 represents three equations and thus we can form a linear system given by

$$\begin{bmatrix} p_1 & p_2 & p_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = p \tag{4}$$

which can be solved for the unknowns in $[u, v, w]$. Points inside a triangle have positive u, v, and w. On the other hand, points outside a triangle have at least one negative coordinate. Eq.4 is computationally expensive because each of the 188 triangles of the mesh model has to check all the range data coordinates to determine whether or not the coordinate points, if any, fall within its interior. A practical implementation is to window the data enclosed by the triangle coordinates as shown in Fig.5. Only the point coordinates within the rectangle are applied in Eq.4.
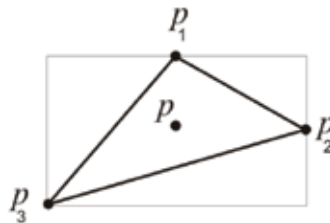
Fig. 5. Windowed 3D range data points.

Figure 6 shows a 2D view of an actual 3D mesh model, superimposed on the range data points. The figure shows an example of segmented 3D data points within one triangle of the eyebrow meshes. We show next how to fit and deform the model's triangles to be as closely as possible to the 3D data.
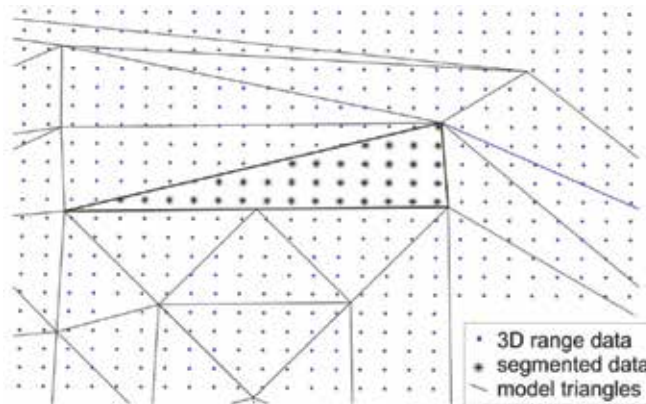
Fig. 6. Segmentation example of the 3D data points within one triangle using barycentric coordinates.

5.3 3D face model deformations
Once the cloud of the 3D data points is segmented by the barycentric coordinate, they are represented by a plane using least square fitting. The general equation of a plane, with non-zero normal vector N, is defined in 3D as

$$aX + bY + cZ + d = 0 \text{ , where } N = (a,b,c) \tag{5}$$

For *n* number of points, Eq.5 can be written in least square form as

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = AB = 0 \tag{6}$$

where the coordinates (X_i, Y_i, Z_i)'s are those of all the data points segmented by the barycentric coordinate. Eq.6 can be solved for the plane equation parameters, $B = [a, b, c, d]$, which is then substituted in Eq.5, leading to a plane representing the 3D data points. Fig.7.a illustrates a concept example of a triangle with 3D data points in 3D space. Fig.7.b shows the segmented data within the triangle which are represented by a plane using Eq.5. From the mathematical geometry of a plane, having the parameters of *B*, any point on the plane can be evaluated. In this work, we deform each corresponding mesh triangle to the 3D data points, by first discarding the three vertices *Z* coordinates, evaluating the *X* and *Y* coordinates, and solving for the new *Z* coordinate (given the parameters in *B* from Eq.6). This produces a mesh triangle, with new depth coordinates, lying on the plane that is approximated by the dense 3D data points. Fig.7.c shows the concept of deforming the mesh triangle to the plane representing the data. Essentially, the pose of the triangle is changed to match that of the plane.
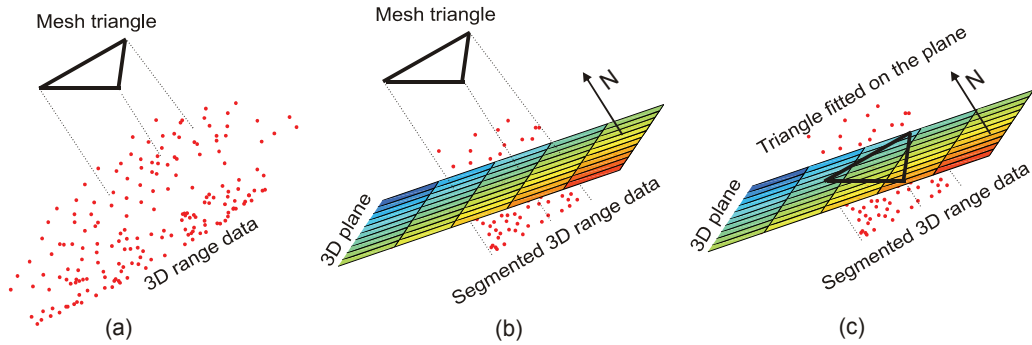
Fig. 7. The process of deforming the triangles of the 3D mesh model, (a) Given cloud of 3D data and a mesh triangle (b) Segmenting the 3D data and plane fitting (c) Deforming the mesh triangle to the plane representing the 3D data.

Subsequently, we repeat the deformation process to all the triangles of the mesh model. Fig.8. shows an example of a complete deformed model superimposed on the data in 2D and 3D views. Comparing Fig.8.a-b with the initially aligned model of Fig.3.c-d, we see that the deformation and fitting of the model to the range data are clearly observed. The triangles of the mesh model have come closer to the data.

The deformed model of Fig.8 is a good representation of the data, yet it's not smooth enough to represent the high resolution and curvatures of the 3D data. In the next step, we subdivide the triangles of the model to a higher resolution in a manner shown in Fig.9.a. New vertices are computed based on the locations of the deformed vertices. Fig. 9.b shows the result of subdividing the deformed model of Fig.8. This process increases the number of vertices and triangles (meshes) of the original model from 109 and 188, respectively, to 401 vertices and 752 polygonal meshes. Finally, because the new triangles do not reflect actual

deformation to the data, we deform them once again using the same deformation process explained above.
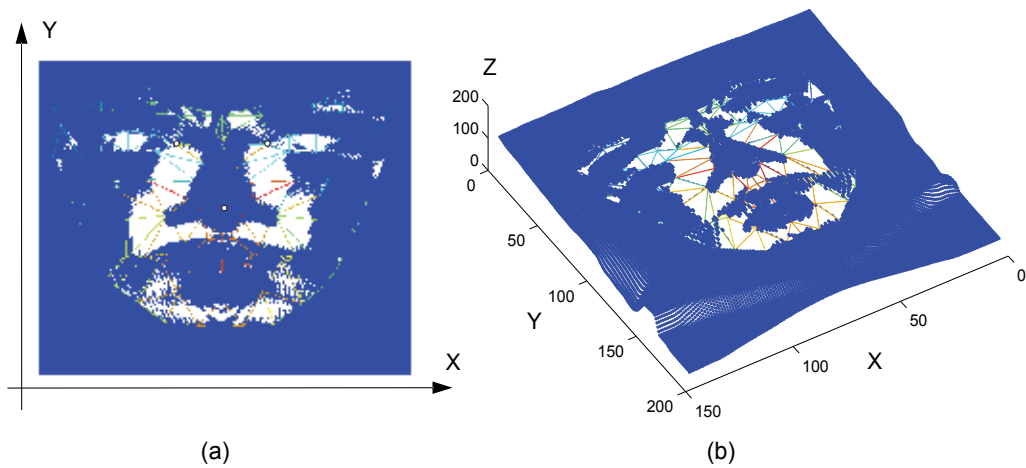

(a)   (b)

Fig. 8. Deformed model superimposed on the range data. (a) 2D view (b) 3D view.

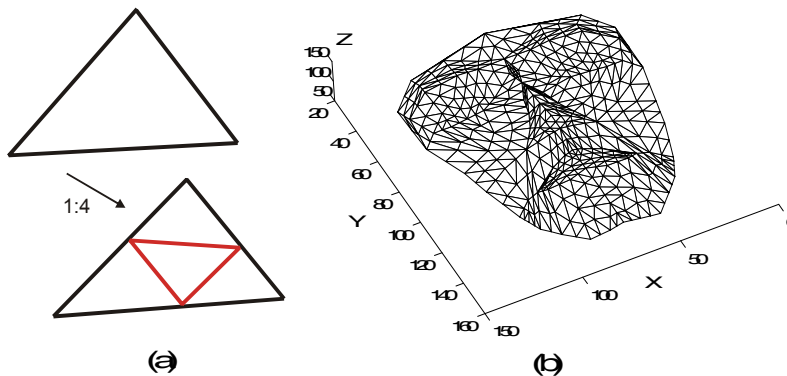(a)   (b)

Fig. 9. (a) 1 to 4 triangle subdivision (b) Result after applying triangle subdivisions to the deformed model of Fig.5.

The introduction of smaller triangles gives more effective triangle fitting of the data especially at areas of high curvatures. Fig. 10.a-b-c show the final result of the deformed model, superimposed on the data in 2D view, 3D view, and a profile 2D view, respectively. In Fig.10.a-b-c, because most of the models' vertices are embedded within the data, we use the "*" symbol to clearly show their locations. Fig.10.d shows a profile (YZ-axis) view of the model in Fig.10.c without the data. This deformed model, containing 401 vertices points, is the final representation of the facial data, which originally contained about 19,000 points (based on an average range image size of 150 by 130). This is nearly a 98 % data reduction. We summarize below the 3D mesh model deformation algorithm:

a.   Given an aligned 3D mesh model to the facial range data, extract the 3D points within each triangle of the mesh model using the barycentric coordinate approach.
b.   For each triangle, fit a plane to the extracted 3D data points and solve for the B parameters in Eq.6.

c.   For each of the three vertices of the mesh triangle, solve for the unknown Z coordinate by evaluating the coordinates of X, Y, and B parameters in Eq.5. This fits the triangle on the plane.
d.   Repeat steps (b) to (c) for all the mesh triangles of the model.
e.   Subdivide the resulting model and repeat steps (a) to (d).
f.   Further subdivision is possible depending on the resolution, quality, or accuracy of the captured range data points.

We show next the application of the deformed model in 3D face recognition.
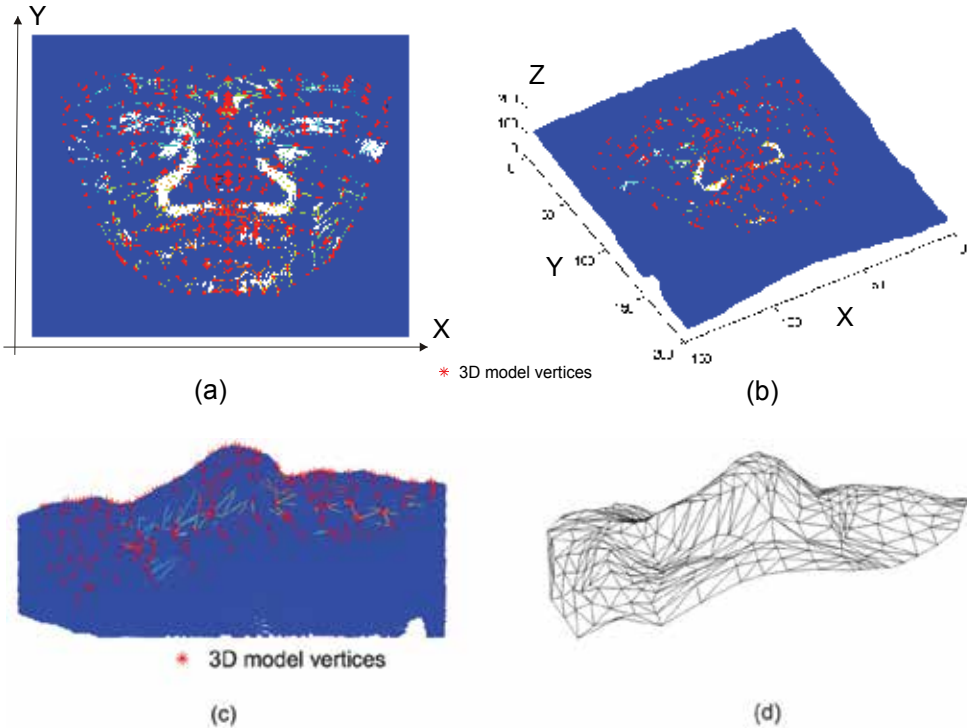
Fig. 10. Final deformed model. (a) 2D view (b) 3D view (c) profile view superimposed on the data (d) mesh model without showing the data.

6. 3D face recognition

Face recognition has received great attentions in the past few years. A recent literature survey for 3D face recognition is given by (Bowyer et al., 2006). The final result of Fig.10 gives a model with 401 deformed vertices specific to a given subject's 3D range data. In this section we explore for different subjects the use of the deformed final models in 3D face recognition. The recognition score is based on a decision level classifier applied to the deformed models obtained from ranges images of a public database.

6.1 Range image database

The range images we use in this chapter are obtained from the publicly available GAVAB database captured by a 3D scanner (Moreno & Sanchez, 2004). This database contains seven

facial range images of 61 subjects: two frontal images with normal expression, two images looking up and down, and three images with facial expression. Many subjects contain instances of dark regions in the face which do not reflect successful 3D scanning, producing in these cases incomplete facial surfaces. As a result, range image pre-processing and filtering are necessary preliminary steps. In this chapter, we are only concerned with modeling and recognizing the frontal images of the database under neutral expressions. Figure 11 contains an example of two views of the texture and range images of one subject. The texture images are not publicly available. For both sets of the frontal range images we obtain the 3D face models as outlined in previous section. One model is used as a query (probe) and the other model is used in the gallery (database). We explain next the recognition technique.

(a)   (b)   (c)
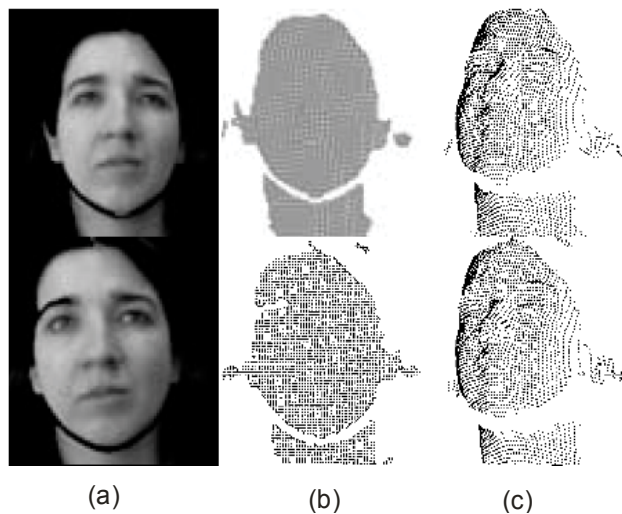
Fig. 11. Subject example of GavabDB (a) Textured 3D image (b) Same image without texture (c) Same image rotated in 3D.

6.2 Decision level fused classifiers

In the recognition stage of Fig.1, a query face model is aligned with all faces in the database and then classified for recognition based on Euclidian distance and voting classifiers. We compute the identification rate using the fusion of both Euclidean distance-based and voting-based classifiers at the decision level of the recognition system. Fig.12 shows a block diagram of the decision level classifier.

The Euclidean distance classifier, even though widely used, its performance can be greatly degraded in the presence of noise. The degradation is due to the equal summation of squared distances over all the features. Any noisy feature with a large distance can mask all other features and as a result the classification considers only the noisy feature, neglecting the information provided by the other features. To overcome this drawback, we use a voting classifier to decide on the final score of the recognition system. The voting classier counts the maximum number of minimum distances of the features between corresponding features points. In this case the feature points are the 401 deformed vertices of the mesh model. In the voting classifier a face is recognized when it has the maximum number of feature (votes) when compared with the corresponding features of the other subjects in the

database. In the presented algorithm, when a query face model is given to the recognition system of Fig.12, it runs through both classifiers; a direct decision for a recognized face is made only when both classifiers' outputs agree on the same recognized face in the database (E=V1 in Fig.12). If the two classifiers are in disagreement, then a different procedure is taken before a final decision is made. In this case, the probe face is directly compared with the recognized face by the Euclidean and the voting classifier, using the voting approach. As a result, the second voting classifier is comparing only two faces. This approach reduces wrong decisions that might be taken by the Euclidean distance classifier, because of possible masking of noisy feature(s), and reroutes the final decision to another voting classifier for final recognition decision. In a scenario when both classifiers actually have the wrong decision, then there is no other clue and a wrong face is falsely recognized.
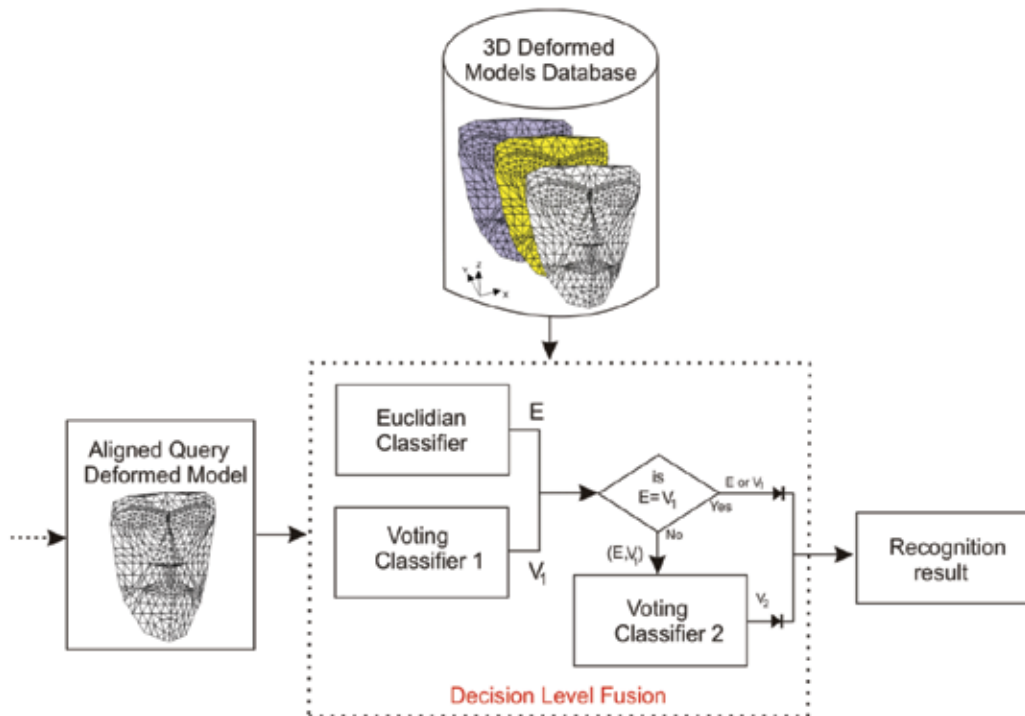
Fig. 12. Decision level fused 3D face recognition system.

6.3 Face recognition experiments

Following the procedure illustrated by the recognition stage in Fig.1 and the classifier of Fig.12, we test the recognition algorithm seperately using the Euclidian distance-based classifier, the voting-based classifier, and the fused classifier of the recognition system. Fig.13 shows the overall Cumulative Match Curve (CMC) identification rate for the 61 subjects of the GAVAB database. From the performance figure, the fused rank one identification rate achieves a 90.2% compared to a lower single classifier rate of 85.2% or 65.6% by the Euclidean or the voting classifier, respectively. The fusion obviously gives superior performance at all ranks. It has been reported that the same database was used in (Moreno et al., 2003) achieving 78 % rank one identification rate for 60 out of 61 subjects using 68 curvature-based extracted features.
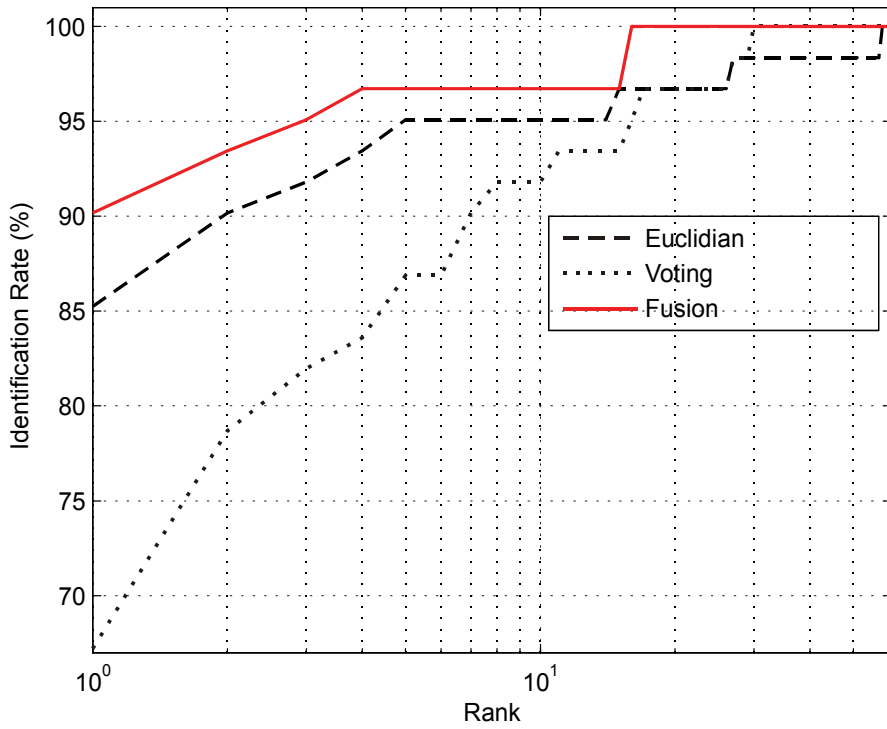
Fig. 13. CMC curves using single and fused classifiers for 401 model's vertices.
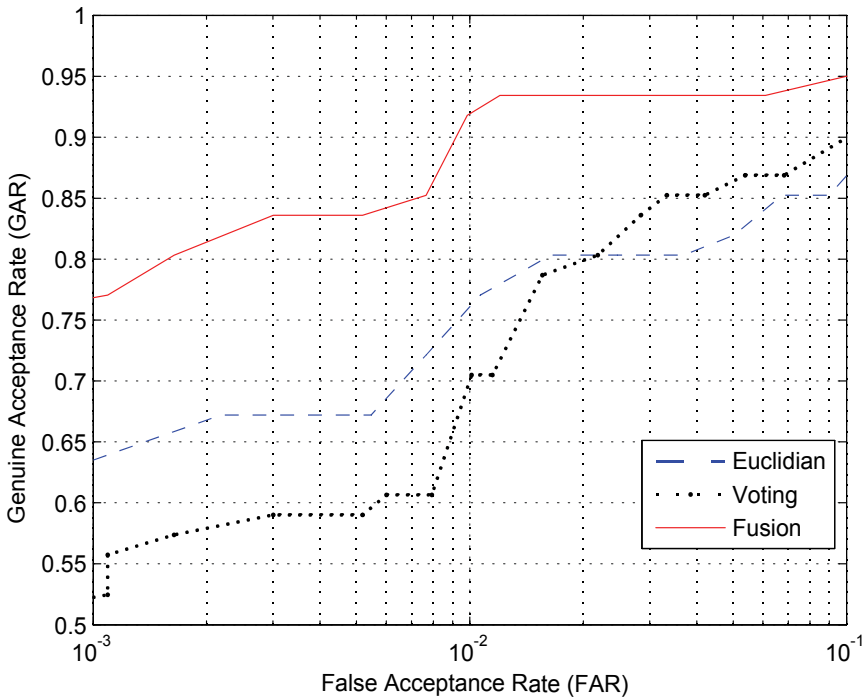

Fig. 14. ROC curves using single and fused classifiers for 401 model's vertices.

Similarly, testing the system in the verification mode, Fig.14 shows the Receiver Operating Characteristic ROC performance curves of the recognition system. At false acceptance rates of 0.1% and 1%, the fused result of the recognition system acheieves genuine acceptance rates of 76% and 92%, respectively.

7. Conclusion and discussion

A model-based algorithm for 3D face recognition from range images is presented. The algorithm relies on deforming the triangular meshes of the model to the range data establishing direct model vertices correspondences with other deformed models in the database. These features correspondences greatly facilitate faster computational time, accuracy, and recognition comparisons. By only detecting three facial features and a generic model, we achieved a 90.2% rank one identification rate using a noisy database. The presented method is proved to be useful for face recognition. However, the method can also be sensitive to noisy or missing data under the mesh model. In the conducted experiments, six subjects out of the 61 were not correctly recognized. The wrong recognition was mainly due to the dataset being either very noisy, incomplete, or the query range image set looks very different from the database set. Unfortunately, the range data pre-processing and
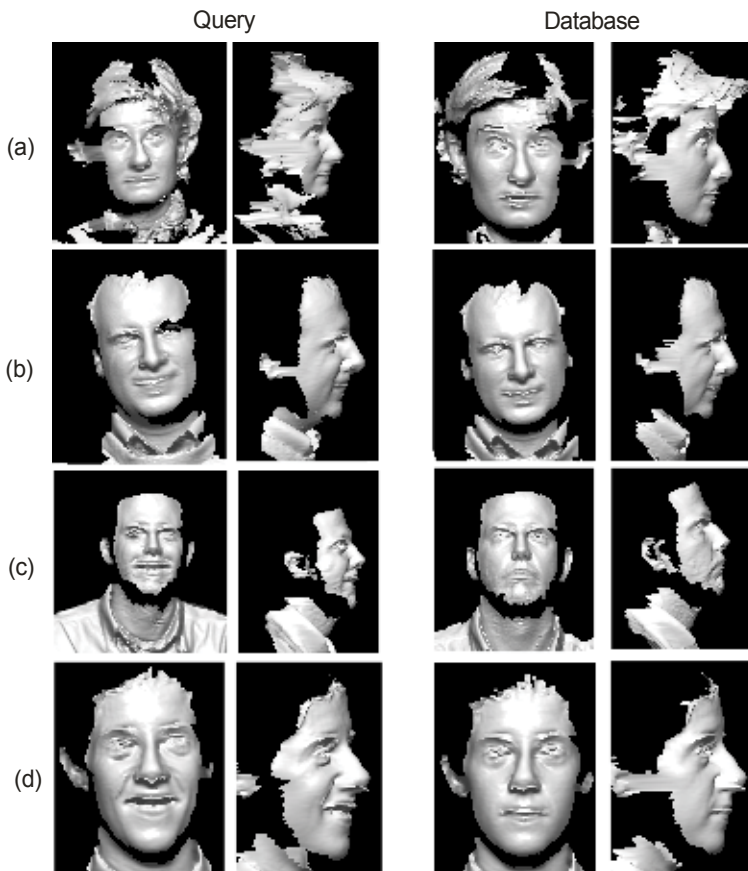

Fig. 15. Probe and database range images of four out of six subjects misrecognized.

filtering, presented in section 4, cannot always cope with large areas of holes or spikes. Fig.15 shows four of the six subjects that were not correctly recognized. Both the query and database sets of Fig.15.a show noisy and incomplete facial scan at the left and right side of the face. Fig.15.b shows similar incomplete data at the eye location. Fig.15.c and Fig.15.d show not only noisy data but also facial expression between the compared query and database images. These factors make the query set of images very different from the database set. In order to demonstrate the robustness of the presented algorithm, better data or another database must be attempted on a large scale datasets captured by high quality 3D scanners. The noise introduced around the subjects' eyes of all subjects in Fig.15 is typical of a lower quality and an older type of 3D scanners. At the time of publishing this chapter, the authors are in process of obtaining a license for the Face Recognition Grand Challenge (FRGC) database (Phillips et al., 2005) in order to apply the algorithm to a much better and cleaner database.

8. References

Achermann, B. & Bunke, H. (2000). Classifying range images of human faces with Hausdorff distance. *The15-th International Conference on Pattern Recognition*, pp. 809–813, September 2000.

Ansari, A. & Abdel-Mottaleb, M. (2005). Automatic facial feature extraction and 3D face modeling using two orthogonal views with application to 3D face recognition. *Pattern Recognition*, Vol. 38, no. 12, pp. 2549-2563, December 2005.

Ansari, A. (2007). 3D face mesh modeling with applications to 3D and 2D face recognition. Ph.D. dissertation, *University of Miami*, Coral Gables, Florida, USA.

Ansari, A.; Abdel-Mottaleb M., & Mahoor M. (2006). Disparity-based 3D face modeling using 3D deformable facial mask for 3D face recognition. *Proceedings of IEEE ICME*, Toronto, Ontario, Canada, July 2006.

Besl, P.; & McKay, N. (1992). A method for registration of 3-D shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.14, pp.239-256, 1992.

Blackburn, D. ; Bone, J. & Phillips, P. (2000). Face recognition vendor test 2000. Technical report, http://www.frvt.org/FRVT2000/documents.htm.

Blanz, V. & Vetter, T. (1999). A morphable model for the synthesis of 3D faces. *Proceedings of SIGGRAPH*, pp. 187-194. 1999.

Bolme, D. (2003). Elastic bunch graph matching, Master thesis, *Colorado State University*, Summer 2003.

Bowyer, K.; Chang, K. & Flynn, P. (2004). A Survey of Approaches to Three-Dimensional Face Recognition. *Proceedings of the International Conference on Pattern Recognition*, Cambridge, England, August 2004.

Bowyer, K.; Chang, K. & Flynn, P. (2006). A Survey of Approaches and chalanges in 3D and multi-modal 3D + 2D face recognition. *Computer Vision and Image Understanding*, vol. 101, pp 1-15, 2006.

Chan, H. & Bledos, W. (1965). A man-machine facial recognition system: some preliminary results. Technical report, *Panaromic Research Inc*, California 1965.

Chang, K.; Bowyer, K. & Flynn, P. (2005). Adaptive Rigid Multi-Region Selection for Handling Expression Variation in 3D Face Recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 157-164, 2005.

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

Coxeter, M. (1969). *Introduction to geometry*, New York, NY, Wiley, 2nd edition, pp. 216-221, 1969.

Chua, C.; Han, F. & Ho Y. (2000). 3D human face recognition using point signature, *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, p.233-238, 2000.

Dorai, C. & Jain, A. (1997). COSMOS- A representation scheme for 3D free-form objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 10, pp. 1115-1130, October, 1997.

Hesher, C.; Srivastava, A. & Erlebacher, G. (2003). A novel technique for face recognition using range images, *The Seventh International Symposium on Signal Processing and Its Application*, 2003.

Hsu, R. & Jain, A. (2001). Face modeling for recognition, *Proceedings of the IEEE International Conference on Image Processing*, Greece, October, 2001.

Huttenlocher, D.; Klanderman, G. & Rucklidge, W. (1993). Comparing images using the Hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 9, pp. 850–863, 1993.

Jiang, D.; Hu, Y., Yan, S., Zhang, L., Zhang, H. & Gao, W. (2004) Efficient 3D reconstruction for face recognition. *Special Issue of Pattern Recognition on Image Understanding for Digital Photos*, 2004.

Johnson, A. & Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3D scenes, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 21, no.5, pp.433-449, 1999.

Kendrick, K.; Da Costa, A., Leigh, A., Hinton M., & Peirce J. (2000). Sheep don't forget a face. *Nature*, pp.165–166, 2000.

Lee, Y. & Shim, J. (2004). Curvature-based human face recognition using depth-weighted Hausdorff distance. *Proceedings of the International Conference on Image Processing*, pp. 1429–1432, 2004.

Lu, X.; Jain, A. & Colbry, D. (2006). Matching 2.5D face scans to 3D models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no.1, pp. 31-43, 2006.

Lu, X.; Colbry, D., Jain, A. (2004). Matching 2.5D scans for face recognition. *Proceedings of the International Conference on Pattern Recognition*, pp. 362–366, 2004.

Lu, X.; Jain, A. (2005). Deformation analysis for 3D face matching. *The 7th IEEE Workshop on Applications of Computer Vision*, pp. 99–104, 2005.

Pan, G.; Han, S., Wu, Z. & Wang, Y. (2005). 3D face recognition using mapped depth images. *IEEE Workshop on Face Recognition Grand Challenge Experiments*, June 2005.

Russ, T.; Koch, K. & Little, C. (2005). A 2D range Hausdorff approach for 3D face recognition, *IEEE Workshop on Face Recognition Grand Challenge Experiments*, 2005.

Russ, T.; Koch, K. & Little, C. (2004). 3D facial recognition: a quantitative analysis. *The 45-th Annual Meeting of the Institute of Nuclear Materials Management*, July 2004.

Mahoor, M.; Ansari, A. & Abdel-Mottaleb, M. (2008). Multi-modal (2D and 3D) face modeling and recognition using attributed relational graph, *Proceedings of the International Conference of Image Processing*, October 2008.

Mahoor, M. & Abdel-Mottaleb, M. (2007). 3D face recognition based on 3D ridge lines in range data, Proceedings of the IEEE International Conference on Image Processing, San Antonio, Texas, September 16-19, 2007.

Medioni, G. & Waupotitsch, R. (2003). Face recognition and modeling in 3D. *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pp. 232–233, October 2003.

Messer, k.; Matas, J., Kittler, J., Luettin, J. & Maitre, G. (1999). XM2VTSDB: The extended M2VTS database. *Proceedings of Audio and Video-based Biometric Person Authentication*, pp. 72-77, March 1999, Washington, D.C.

Moreno, A.; Sánchez, Á., Vélez, J. & Díaz, F. (2003). Face recognition using 3D surface-extracted descriptors, *The Irish Machine Vision and Image Processing Conference*, September 2003.

Moreno, A. & Sanchez, A. (2004). GavabDB: A 3D face database," Proceeding of the 2nd COST275 Workshop on Biometrics on the Internet: Fundamentals, Advances and Applications, Vigo, Spain, March, 2004.

Phillips, P.; Moon, H., Rizvi, S. & Rauss, P. (2000). The FERET evaluation methodology for face recognition algorithmes. *IEEE Transaction on Pattern Analysis Machine Intelligence*, Vol. 22, No. 10, October 2000.

Phillips, P.; Grother, P. , Micheals, R., Blackburn, D., Tabassi, E. & Bone, J. (2003). Face recognition vendor test 2002. http://www.frvt.org/FRVT2002/Default.htm. Evaluation Report, 2003.

Phillips, P.; Flynn, P., Scruggs, T., Bowyer, K., Chang, J., Hoffman, K., Marques, J., Min, J., & Worek, W. (2005). Overview of the face recognition grand challenge, *Proceedings of. IEEE on CVPR* , San Diego, pp. 947-954, June 2005.

Uchida, N.; Shibahara, T., Aoki, T., Nakajima, H. & Kobayashi, K. (2005). 3D face recognition using passive stereo vision. *Proceedings of the IEEE International Conference on Image Processing*, pp. II-950-II-953, Sep. 2005.

Wiskott, L.; Fellous, J., Kruger, N. & Malsburg, C. (1997). Face recognition by elastic bunch graph matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 775–779, 1997.

Wu, Z.; Wang, Y. & Pan G. (2004). 3D face recognition using local shape map, *Proceedings of the. IEEE International Conference on Image Processing*, Vol. 3, pp. 2003- 2006, Oct. 2004.

Xu, C.; Wang, Y., Tan, T. & Quan, L. (2004a). Automatic 3D face recognition combining global geometric features with local shape variation information. *Proceedings of the Sixth International Conference on Automated Face and Gesture Recognition*, pp. 308–313, May 2004.

Xu, C. ; Wang, Y, Tan, T. (2004b). Three-dimensional face recognition using geometric model," *Proceedings of the SPIE*, Vol. 5404, pp. 304-315, August 2004.

Zhaho, W.; Chellappa, R., Phillips, P. & Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Survey*, pp. 399-458, December 2003.

Occlusions in Face Recognition: a 3D Approach

Alessandro Colombo, Claudio Cusano and Raimondo Schettini
Università degli studi di Milano Bicocca,
Italy

1. Introduction

The real challenge in face detection and recognition technologies is the ability to handle all those scenarios where subjects are non-cooperative and the acquisition phase is unconstrained. In the last few years a great deal of effort has been spent to improve the performances where cooperative subjects are acquired in controlled conditions. However, in those scenarios other biometrics, such as fingerprints, have already proved to be well suited; in fact, the performances obtained using them are good enough to implement effective commercial systems. In all those cases where the application requires no constraints during the acquisition phase, the face is one of the best candidates among biometrics. The face is a non-touch biometrics and is also the natural way people use to recognize each other: for this reasons it is more accepted by the final users.

The fundamental problem in recognizing people in unconstrained conditions is the great variability of the visual aspect of the face introduced by various sources. Given a single subject, the appearance of the face image is disturbed by the lighting conditions, the head pose and orientation of the subject, the facial expression, ageing and, last but not least, the image may be corrupted by the presence of occluding objects. This great variability is the reason that makes face detection and recognition two of the toughest problems in the fields of pattern recognition, computer vision and biometrics.

One of the less studied aspects seems to be the presence of occluding objects. In unconstrained real-world applications it is not an uncommon situation to acquire subjects wearing glasses, scarves, hats etc.; or subjects talking on the phone, or, for some reason, having their hands between their face and the camera. In all these kinds of situations most of the proposed algorithms are not able to grant acceptable performances or to produce any kind of response at all.

Some approaches (Lin, 2004; Hotta, 2004) propose the detection of partially occluded faces in two dimensional images using Support Vector Machines (SVM) or a cascade of classifiers trained to detect subparts of the face. For recognition, few approaches working on 2D data are able to recognize people using only the visible parts of the face. For example, (Park et al., 2005) have proposed a method for removing glasses from a frontal image of the human face. A more general solution is needed, however, when the occlusions are unforeseen and the characteristics of the occluding objects are unconstrained. The problem has been addressed using local approaches which divide the face into parts which are independently compared. The final outcome is determined by a voting step. For example, see (Martinez, 2000; Martinez, 2002; Kim et al. 2005). A different approach has been investigated by (Tarrés &

Rama, 2005). Instead of searching for local non-occluded features, they try to eliminate some features which may hinder recognition accuracy in the presence of occlusions or changes in expression. In (De Smet et al., 2006) a morphable model based approach is proposed. The parameters of a 3D morphable model are estimated in order to approximate the appearance of a face in a 2D image. Simultaneously, a visibility map is computed which segments the image into visible and occluded regions.

In this chapter we present our approach: a full automatic recognition pipeline based on 3D imaging (Fig. 1). We have chosen to use 3D data because by having depth information available, it is easier to detect and isolate occluding objects. This makes it possible to detect and recognize partially occluded faces. There are also other well known advantages using 3D sensors: lighting independence, the possibility to normalize pose and orientation, and last but not least, 3D sensors are more difficult to circumvent compared to 2D cameras (Fig. 2).

The first three modules of the pipeline are the core of our method. The face detector, based on an improved version of a primal approach presented in (Colombo et al., 2006), is able to localize the presence of human-size faces in depth images. The algorithm is based on curvature analysis, ICP-based normalization and Gappy PCA classification. The main advantages of the algorithm are its independence from scale and orientation in the image plane and its ability to deal with occlusions if at least two eyes or one eye and the nose are visible. The output of the detector is the normalized position and orientation of the faces in the acquired scene. At this point, the image of each detected face is analyzed by the occlusion detection module. Each pixel is classified as occluded or free using a statistical model of the human face. Once the visibility map is computed, the restoration module reconstructs the occluded parts using gappy PCA reconstruction (Everson & Sirovich, 1996). Since the images are restored, it is possible to adopt any state of the art feature extraction and matching module. Alternatively, a partial matching approach may be adopted using the non-restored face image and the visibility map.
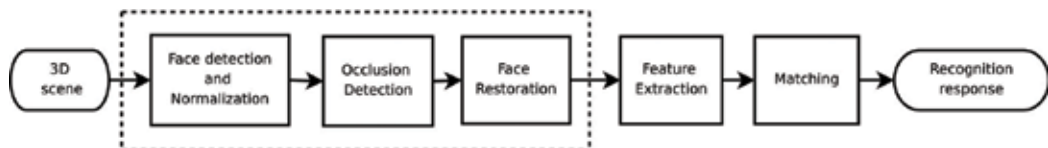

Fig. 1. The proposed approach. The modules enclosed in the dotted box are those described in the paper. Feature extraction and matching modules may be any state of the art approach.

2. Face detection and normalization

Face detection in 3D images is not a common task; a primal approach has been presented in (Colombo et al., 2006). Detecting faces in 3D images presents some benefits: first of all, 3D data is independent from scale. This is a big advantage because face detectors have to deal with one less degree of freedom. The second reason to adopt 3D images regards the nature of the 3D data itself. There are some properties of surfaces, like the curvatures, that are independent from the reference system. Using features like these, it is possible to recognize typical parts of the face independently from pose or viewpoint. For these reasons, 3D data facilitate the problem of face detection.
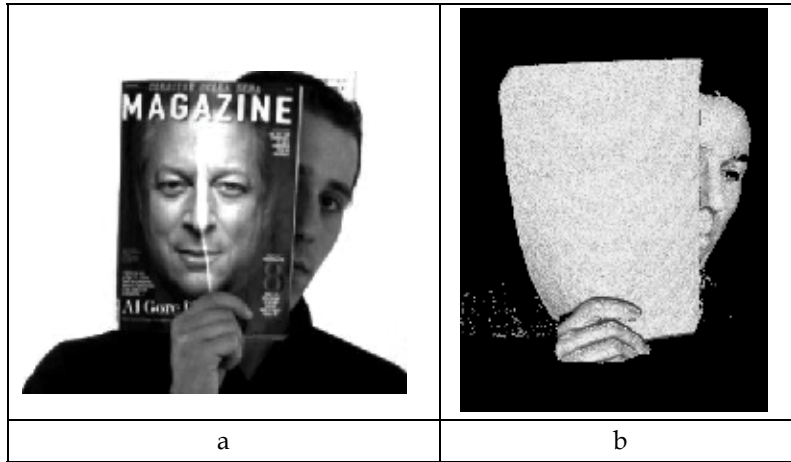
Fig. 2. (a) a subject trying to circumvent a 2D recognition system using a magazine. (b) the same scene acquired by a 3D sensor.

The approach that will be presented here reveals another important feature of 3D data. Considering the depth component of the image, the discrimination between face and occluding objects can be done using simple criteria. Intuitively an occluding object can be simply viewed, in terms of depth information, as something that is added to the original face surface, as can be seen in Figure 3. As will be shown, it is possible to handle this kind of noise in complex tasks like face detection and normalization.
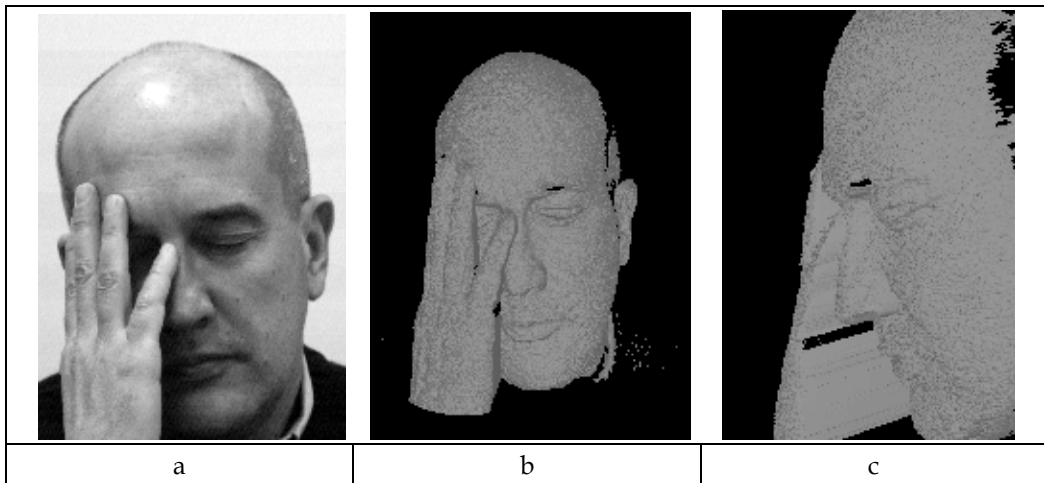

Fig. 3. (a) A two dimensional image of an occluded face. (b) The three dimensional image of the same scene represented in (a). (c) Profile view of (b).

2.1 Algorithm overview
The face detector is based on the work presented in (Colombo et al., 2006). The input of the algorithm is supposed to be a single 3D image of the scene. If other representations are available, a 3D image can be generated using simple and well known rendering techniques, like variants of the Z-Buffer algorithm (e.g. Watt, 1999). Figure 4 shows a concise diagram

representing the main steps of the algorithm. To render the problem less computationally intensive, single facial features, such as eyes and noses, are initially searched. Consequently, this first step results in an image segmentation in regions corresponding to candidate facial features. No relationships are established for the moment, between the segmented elements. In the next step potential faces are created from candidate noses and/or candidate eyes. The goal now is to discriminate between candidate faces that correspond to actual faces and those that do not. The method now registers each candidate face in a standard position and orientation, reducing intra-class face variability. The areas of the range image covered by each candidate face is further analyzed by applying a face versus non-face classifier based on a holistic approach. Knowledge about the structure of the face is used therefore only in the generation of a list of candidate face regions, while the actual classification of these regions as faces is purely holistic.
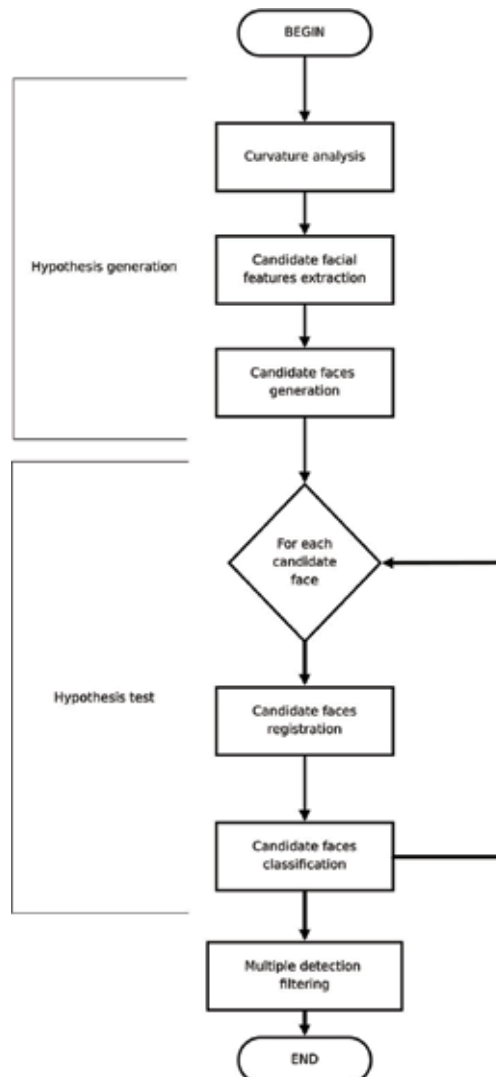

Fig. 4. The face detector main diagram

In Figure 5 a more detailed description of the processing steps is shown. Once the scene is acquired, surface curvature, which has the valuable characteristic of being viewpoint invariant, is exploited to segment candidate eyes and noses. In greater detail: (i) the mean (*H*) and Gaussian (*K*) curvature maps are first computed from a smoothed version of the original range image; (ii) a simple thresholding segments regions of high curvature which
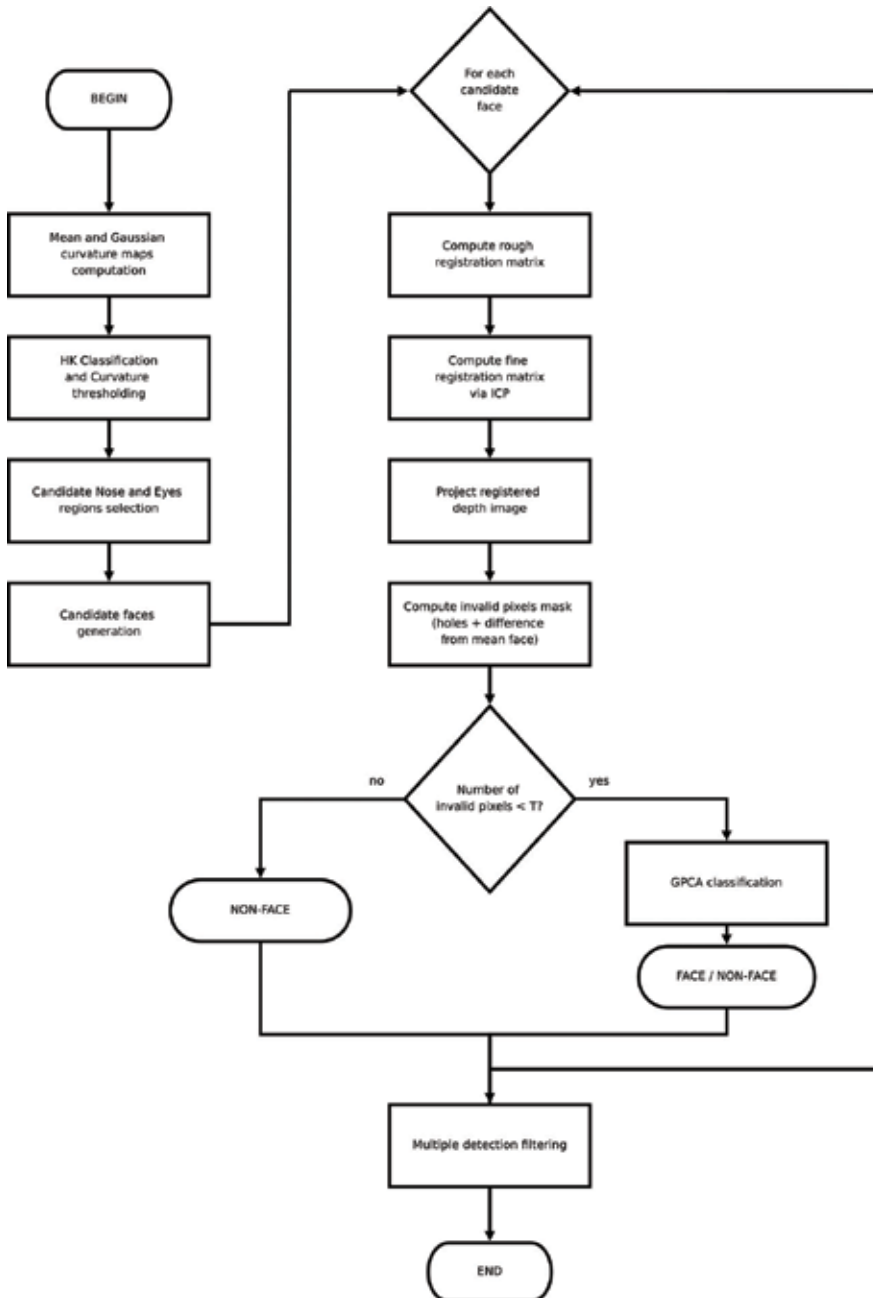

Fig. 5. The face detection algorithm: detailed diagram.

might correspond to eyes and noses; (iii) an HK classification, based on the signs of Gaussian and mean curvature, divides the segmented regions into four types: convex, concave, and two types of saddle regions. Regions that may represent a nose or an eye are then characterized by their type and some statistics of their curvature. The output of the processing step may contain any number of candidate facial features. If no nose or eyes are detected, the method assumes that no faces are present in the acquired scene; while there are no upper bounds on the number of features that can be detected and further processed. Combinations of candidate features are used to select corresponding 3D surface regions including the eyes and the nose, but excluding the mouth and part of the cheeks.

Each region is then rotated and translated into a standard position using a rough+fine registration approach based on an occlusions-tolerant version of the ICP algorithm (Besl & McKay, 1992), and a new depth image of the area containing the candidate facial features is computed. In order to select only the rigid part of the face, the image is cropped with a binary mask. Then, the image is analyzed in order to find occluding objects: if present, occluding objects are eliminated from the image invalidating the corresponding pixels. Finally, a face vs. non-face gappy PCA based classifier, which has been trained on several examples, processes the candidate depth image. The final output of the procedure is a list containing the location and orientation of each detected face.

2.2 Candidate faces generation

Given in input a 3D scene (in the form of a range image in our current implementation) a curvature analysis reveals regions of interest having similar characteristics to those of human eyes and noses. By combining all these regions, it is possible to generate hypotheses about the presence of faces in the scene. For the generation of a single hypothesis, at least two eyes or an eye and a nose must be present. The curvature analysis processing is based on the computation of gaussian and mean curvature maps. Using simple thresholding of low curvature values it is possible to isolate potential noses (convex regions) and eyes (elliptical concave regions). See (Colombo et al., 2006) for an in-depth description of the procedure adopted.

The generation of candidate facial features results in a set composed of two kinds of features: eyes and noses. A single candidate face is generated combining two eyes, or an eye and a nose, or two eyes and one nose. These cases can handle occlusions in some parts of the face; for example a hand on an eye or a scarf in front of the tip of the nose. The regions generating a candidate face must satisfy some constraints about distances between themselves (Gordon, 1991). Figure 6 shows an example of a candidate face generation.

From an actual face multiple candidates could be generated. Moreover, in the case of eye pairs or nose-eye pairs, double candidate faces are generated because of the ambiguities regarding the actual face orientation. For example, from a pair of eyes, either an upward or a downward face might be present. Multiple, spurious detections are eliminated as a final step by a simple filtering process.

2.3 Candidate face normalization

Once generated, candidate face images are normalized in pose and orientation in order to perform a final classification into faces or non-faces. The normalization is computed in two steps. First, a rough normalization is performed using the position of the candidate facial features. A reference position is defined aligning the eye positions with the X axis and the

plane passing through the eyes and the tip of the nose rotated by 45 degrees around the same axis. Finally, the tip of the nose is translated to the origin. In all those cases where one feature is missing (for example in case of occlusions) a degree of freedom is left undetermined.
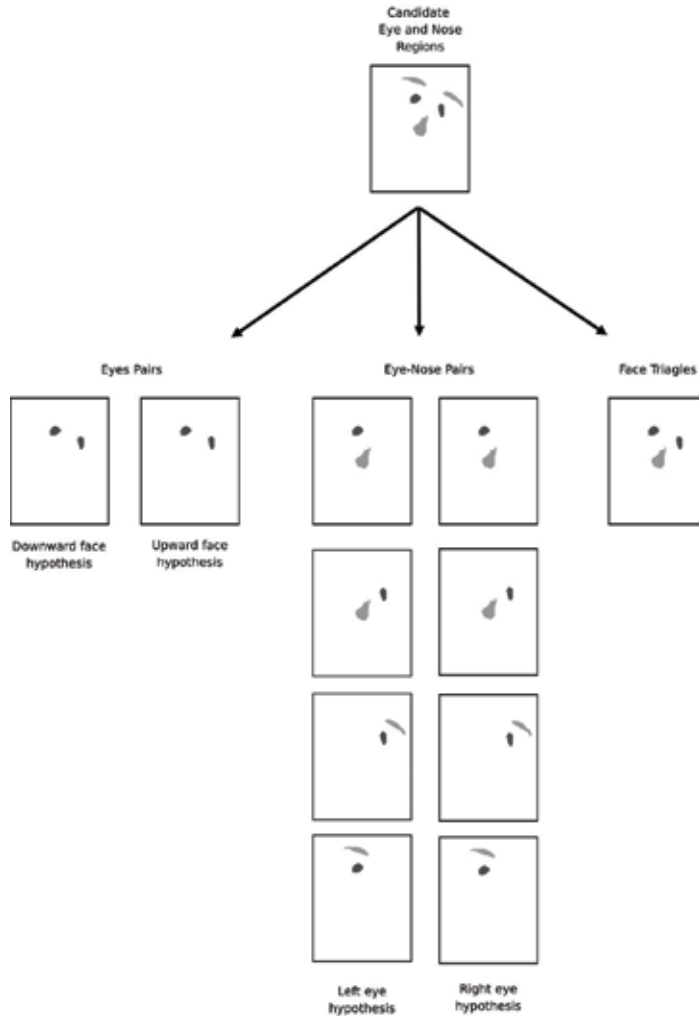

Fig. 6. An example of candidate face generation. The image on top shows the selected candidate features (light gray for noses, dark gray for eyes). Below, all the candidate faces generated by the algorithm are shown. In the case of eye pairs and eye-nose pairs, two candidate faces are generated for each pair because of the impossibility of determining the actual face orientation.

The rough normalization procedure generates a good starting position for the following refinement step based on the ICP algorithm (Besl & McKay, 1992). The idea here is to refine the position registering the candidate face with a mean face template (Fig. 7). We used a customized version of the ICP algorithm aimed to handle the presence of extraneous objects. The algorithm has been inspired by the variants presented in (Rusinkiewicz & Levoy, 2001).

Fig. 7. The mean face template used for fine registration. The surface, composed of approximatively 600 vertexes, has been computed using a training set of manually normalized faces.

The ICP algorithm requires a matching criteria in order to find correspondences between the points of the surfaces to be registered. In our implementation we used a projective matcher . Based on the assumption that the rough registration computes a good registration (i.e. with a low error) between the mean face and the candidate face surface, the projective matcher tries to find correspondences using orthographic projections of each vertex. More precisely, at each iteration of the main ICP loop, the mean face template and the candidate face are orthographically projected using the same camera, resulting in a pair of 3D images, the data image and the model image. For each point located at coordinates *(i,j)* in the data image space, the correspondent point is searched in the model image space in locations *(i±r, j±r)*; where *r≥1* is an integer defining a square region around the current location. The correspondence criterion is the point at minimum distance using the 3D Euclidean metric.

In order to deal with the presence of occlusions, ICP has been customized by including a correspondence rejector which allows the registration process to avoid the use of those points which probably belong to the occluding objects.

Given a correspondence :

$$ c \quad (\vec{p}_d, \vec{p}_\mu, \vec{n}_d, \vec{n}_\mu) $$

(where *p* indicates a surface point, *n* a surface normal; the subscript *d* indicates the data surface while *μ* indicates the mean surface) the rejector verifies that the following conditions are satisfied:

$$ \arccos(\vec{n}_d \cdot \vec{n}_\mu) < T_n. \tag{1} $$

$$ \vec{p}_d - \vec{p}_\mu _2 < T_d. \tag{2} $$

The first condition (Equation 1) checks if the angle between the two normals is inferior to a predefined threshold. We have chosen a value of 90 degrees; so the check filters out all those matches that are clearly wrong because the orientation of the surfaces is very dissimilar.

The second condition (Equation 2) assures that the distance between the two correspondent points must be below a predefined threshold. The value has been computed considering the variations between the normalized non-occluded faces from the training set and the mean

face template. A distance greater than 15mm is very improbable (see Fig.8 for the distribution of the differences) and thus we have chosen this value.

If at least one of the two conditions is not satisfied then the correspondence is rejected. Only the filtered correspondences are used to compute the registration.
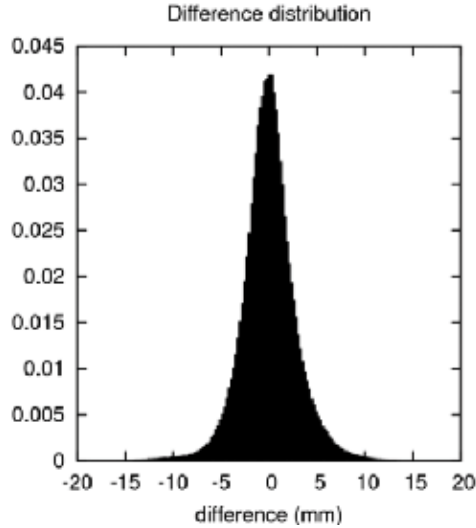

Fig. 8. The histogram of the differences in depth between the mean face template and the normalized faces from the training set.

2.4 Candidate face classification through GPCA

Once registration is computed for each candidate face, depth images are generated using orthographic projections of the original acquisition. After cropping, each image is compared with the mean face in order to detect occluding objects. For each pixel p the following condition is checked:

$$|Y(p) - \mu(p)| \leq T_d.$$

(3)

where Y is the candidate face depth image, while μ is the mean face depth image. T_d is the same threshold used in Equation 2. If the check fails, the pixel is invalidated. In this way, large parts of occluding objects can be eliminated. Those non-face pixels passing the check are assured to be limited in depth by T_d. Since large regions of the image may be invalid, a check on the fraction of valid pixels is performed:

$$\frac{n_v}{N} > T_v$$

(4)

where N is the number of total pixels in the image and T_v is the valid pixel threshold. Images with a low number of valid pixels are more difficult to classify because of the lack of information. The check is also used to avoid degenerative cases; i.e. images composed of only a few pixels. At this point, images present invalid regions of pixels due to occlusion detection or holes generated by acquisition artifacts. In order to classify them, a Gappy PCA classifier has been adopted. The classifier is based on Principal Component Analysis for gappy data (Everson & Sirovich, 1995).

GPCA extends PCA to data sets that are incomplete or gappy. When the intrinsic dimension is smaller than that of its representation, some of the information in the original representation is redundant, and it may be possible to fill in the missing information by exploiting this redundancy. The procedure requires knowledge of which parts of the data are available and which are missing.

A set of N patterns

$$\{\vec{x}_1, \cdots, \vec{x}_N\} \subset R^n,$$

extracted from a training set of normalized non-occluded faces is used to determine the PCA basis so that a generic pattern $\vec{x}$ can be approximated using a limited number, M, of eigenvectors:

$$\vec{x} \simeq \vec{\mu} - \sum_{i=1}^{M} \alpha_i \vec{v}_i, \tag{5}$$

where $\vec{\mu}$ is the mean vector, $\vec{v}_i$ is an eigenvector, and α_i is a coefficient obtained by the inner product between $\vec{x}$ and $\vec{v}_i$.

Suppose there is an incomplete version $\vec{y}$ of $\vec{x}$, and suppose that the location of missing components is encoded in the vector $\vec{m}$ ($\vec{m}_i = 0$ if the i-component is missing, otherwise $\vec{m}_i = 1$). GPCA searches for an expression similar to Equation 5 for the incomplete pattern $\vec{y}$:

$$\vec{y} = \vec{y}' \simeq \vec{\mu} + \sum_{i=1}^{M} \beta_i \vec{v}_i, \tag{6}$$

Note that $\vec{y}'$ has no gaps since the eigenvectors are complete. To compute the coefficients β_i the square reconstruction error E must be minimized:

$$E = \|e\|^2 = |\vec{y} - \vec{y}'|^2. \tag{7}$$

However, this expression includes the missing components, while only the available information must be considered. To do so, it is useful to introduce the gappy inner product and the corresponding gappy norm:

$$(\vec{v}, \vec{u})_{\vec{m}} \quad \sum_{i=1}^{n} \vec{v}_i \vec{u}_i \vec{m}_i. \tag{8}$$

$$|\vec{v}|_{\vec{m}} = \sqrt{(\vec{v}, \vec{v})_{\vec{m}}} \tag{9}$$

Now the error E can be redefined in such a way that only the available components are considered:

$$E = \|e\|_{\vec{m}}^2 = |\vec{y} - \vec{y}'|_{\vec{m}}^2. \tag{10}$$

The gappy pattern $\vec{y}$ can be reconstructed as $\vec{y}'$ using the Equation 6, where the coefficients β_i are found by minimizing E (for additional details see Everson & Sirovich, 1995). Figure 9 shows an example of a face range image reconstructed using GPCA.
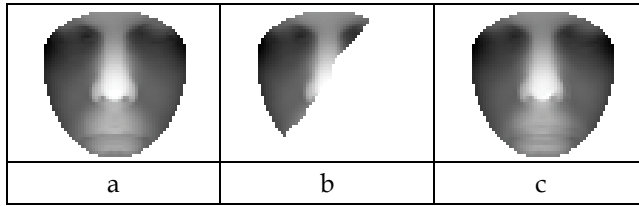
Fig. 9. Example of a reconstruction of a gappy image: (a) the original depth image; (b) the same image with a region of invalidated pixels; (c) the reconstruction of (b) through gpca.

The classifier used for the face detector constructs the vector $\vec{m}$ through lexicographical construction, considering all the invalidated or holes pixels. The error defined in Equation 10 is used as a measure of faceness. Thus, a candidate face is classified as a face if the following condition is satisfied:

$$\frac{E}{n_v} \leq T_f \tag{11}$$

where n_v is the number of valid pixels and T_f is a predefined threshold. The error E needs some kind of normalization because the number of valid components is not fixed. Here the most simple normalization, the division by the number of valid pixels n_v, has been adopted; but other kinds of normalization approaches could be experimented with as well.

3. Occlusion detection and face restoration

At this stage of the pipeline, faces are encoded as range images projected from the normalized position produced by the face detector. Any part of these images that does not look like part of a face and lies between the acquisition device and the acquired face is considered an occlusion (occluding objects may not touch the face). In other words, sets of points which do not fit the model of a non-occluded 3D face are classified as occlusions. The 3D face model is obtained by the popular eigenfaces approach (Sirovich & Kirby, 1990). Occluded faces cannot be well represented by the linear combination of the computed eigenfaces. Therefore, the distance from the face space (DFFS) of occluded faces is expected to be quite large and can be used to reveal the presence of occlusions, though not their location. Differences in the pixels of the reconstructed and the original image (vector $\vec{e}$) are likely to be more pronounced where the face is occluded.

A preliminary mask **M** of occlusions can be simply obtained by thresholding vector $\vec{e}$:

$$\mathbf{M}_i = \begin{cases} 1 & \text{if } \vec{e}_i > T_\tau \\ 0 & \text{if } \vec{e}_i < T_\tau \end{cases} \tag{12}$$

where T_τ is a threshold that must take into account the resolution of the imaging device, acquisition noise, and the accuracy achieved in face detection and pose normalization. Since occlusions must be located between the acquisition device and the acquired face, their z coordinates will be greater than those of the reconstructed face, resulting in positive components of $\vec{e}$. Unfortunately, the reconstruction error of the non-occluded regions is influenced by the occlusions, which may determine an inaccurate choice of the gappy

projection coefficients β_i. However, the effect is significant only for occlusions which are very far from the face, or very large in size. Occluding objects which are far from the face are detected on the basis of the difference between the (possibly) occluded face $\vec{y}$ and the mean (non-occluded) face $\vec{\mu}$. Components of $\vec{x} - \vec{\mu}$ which are positive and large enough can be considered part of an occlusion. A mask $\vec{B}$ of these occlusions is obtained as follows:

$$B_i = \left\{ \begin{array}{ll} 1 & \text{if } y_i - \mu_i > T_\rho \\ 0 & \text{if } \vec{y}_i \quad \vec{\mu}_i < T_\rho \end{array} \right. \tag{13}$$

T_ρ threshold must be tolerant with respect to face variability in the data set to be processed. As previously discussed, it is very unlikely that any face differs more than 15 mm from the mean face. Consequently, the threshold T_ρ has been set at that value.

A more accurate estimate of the occlusion mask **M** can be obtained by excluding from the computation of the reconstruction error E the pixels selected in mask B. The key idea here is to use only non-occluded parts of the face to estimate the distance between the face space and the processed face. For this, we apply Gappy Principal Component Analysis (GPCA). The coefficients obtained by minimizing Equation 10 can now be substituted in Equation 6 to determine a more accurate reconstruction error.

The same technique may be applied to deal with cases in which the 3D face in input is incomplete (e.g. missing data from the scanner). Holes can be encoded in a mask **H**, so that the gappy products $(\cdot,\cdot)_B$ may be replaced by $(\cdot,\cdot)_{B \vee H}$, where $B \vee H$ is the component-wise logical OR of holes and occlusions. Analogously, color information could be exploited if available: points having a color which is not likely to be found on a face can be marked as occlusions.

On the basis of the new reconstruction error E obtained using the coefficients β_i, the occlusion mask **M** can now be determined more precisely. Here again, when the reconstruction error is high, the corresponding pixel is considered occluded. A variation of (12) is used:

$$M_i = \left\{ \begin{array}{ll} 1 & \text{if } (e_i > T_\tau) \vee B_i = 1 \vee H_i = 1 \\ 0 & \text{otherwise} \end{array} \right. \tag{14}$$

As a final step, morphological filters can be used to clean the occlusion mask, enforce the locality of the occlusion, and discard tiny regions. If the resulting mask **M** is empty, the face is considered non-occluded, and can be directly processed for recognition. Otherwise **M** replaces **B** as the mask for the gappy projection. The solution yields the coefficients β_i, which employed as indicated in Equation 6 allow for the restoration of the input face.

At this stage, any recognition algorithm, whether holistic or feature-based, can then be applied to recognize the restored face. Holistic approaches may use the fully restored image while partial matching or feature-based approaches may exclude occlusions using the information contained in **M**. Fig. 10 summarizes the whole occlusion detection and restoration procedure. Figure 11 presents an example of restoration showing the steps in the computation of the occlusion mask.
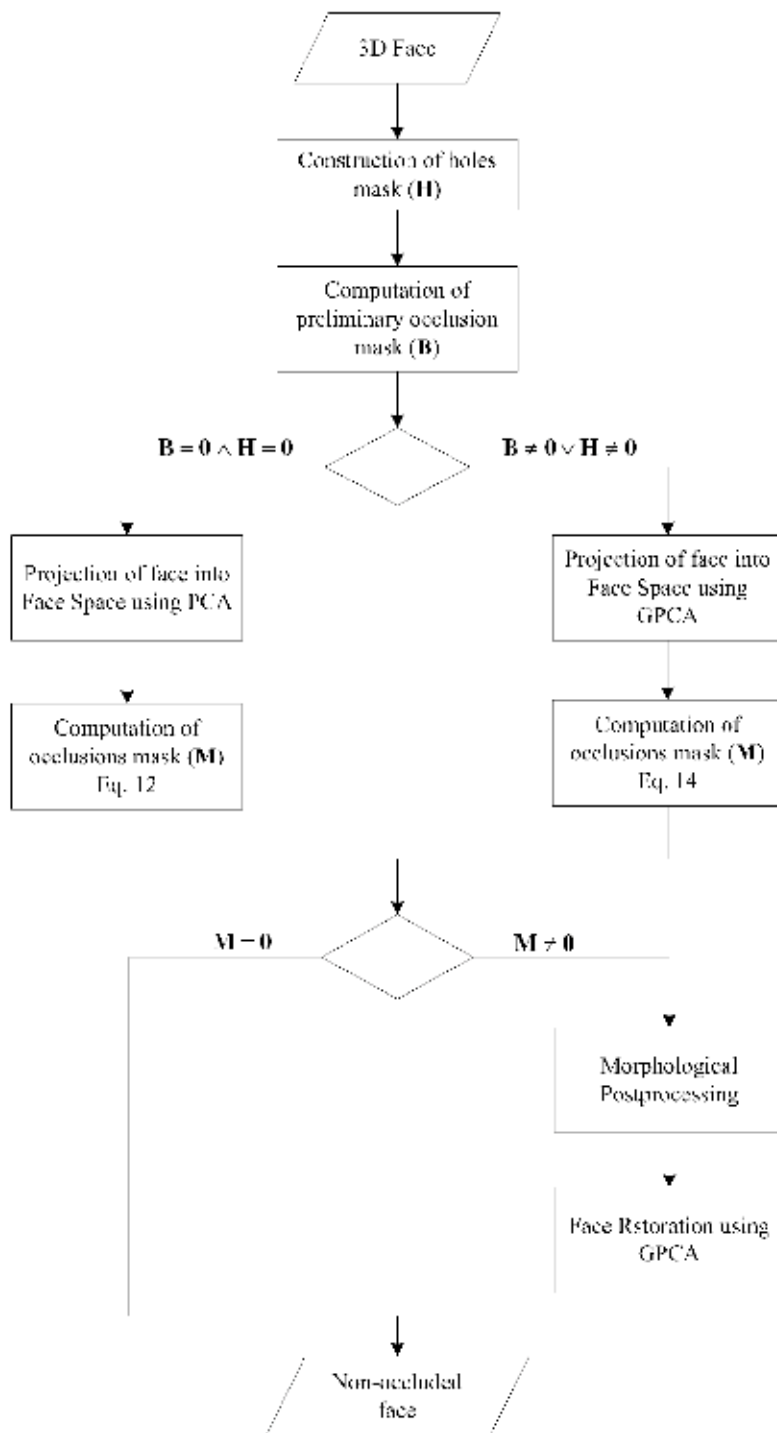
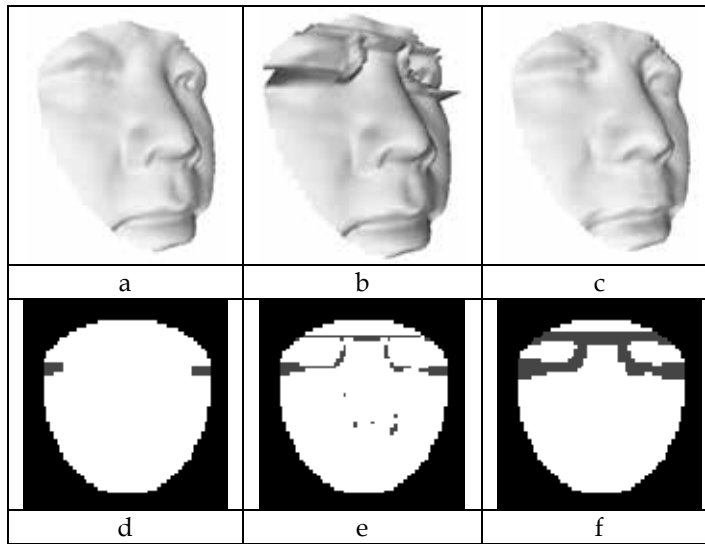Fig. 10. The occlusion detection and face restoration diagram.

Fig. 11. Occlusion detection and restoration on a face artificially occluded by glasses (b). The restored face (c) is very similar to the original non-occluded face (a). A few points have been preliminarily detected (d) as occluded using Eq. 13; the second step correctly identifies most of the occlusion (e) using Eq.14. The final occlusion mask (f) is the result of the application of simple morphological operators.

4. Experimental results

4.1 Dataset and artificial occlusion generation

In order to perform an in-depth analysis of the algorithms (which needs the presence of a complete ground truth) we decided to adopt the artificial occlusion solution. It consists in taking an existing database of non-occluded faces and adding occluding 3D objects in each acquisition. Occluding objects are real world objects acquired at the IVL Laboratory at the University of Milano Bicocca. The entire set of objects includes: a scarf, a hat, a pair of scissors, two types of eyeglasses, a newspaper and hands in different configurations. We have used the UND Database (Chang et al., 2003), which consist of 951 3D+2D acquisition of 277 subjects.

Occluded acquisitions are generated inserting the objects in the acquisition space at plausible positions. This means, for example, that the eyeglasses are placed in the eyes region, the scarf in front of the mouth and so on. Thus, for each type of object, a starting position and orientation (T_o, R_o) is manually predefined considering the normalized mean face template as a reference. Then, for each acquisition an object is randomly chosen and placed on the face. The position and orientation is finally perturbed with random noise in order to increase the variability of the test set. Fig. 12 shows some examples.

4.2 Face detection results

The face detector and normalization module has been tested occluding a subset of 476 acquisitions from the 951 images of the UND Database. A training set of 150 non occluded faces, taken at the IVL Laboratory with the Minolta Vivid 900 range scanner, has been used as the training set for the GPCA eigenspace computation.
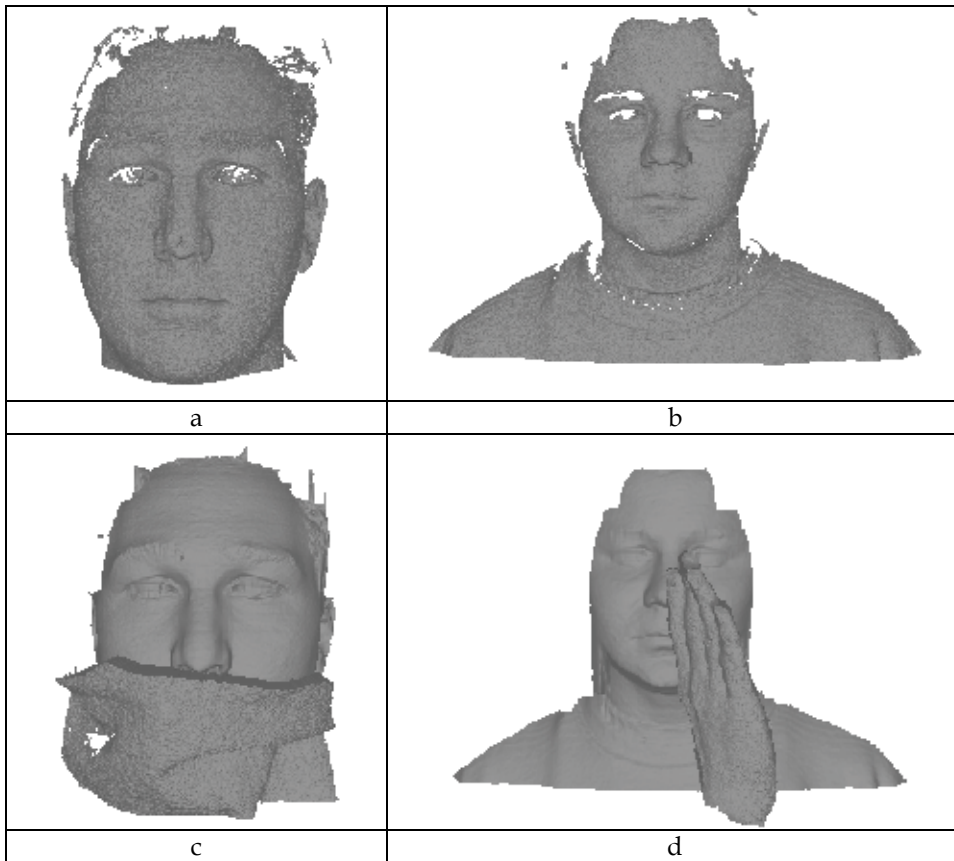
Fig. 12. Some examples of artificially occluded faces. (a,c) The original UND acquisitions; (b,d) the occluded faces. Note that automatic preprocessing is applied to the occluded faces, i.e. smoothing and simple hole closure via linear interpolation.

Figure 13 shows the receiver operator characteristic curves for the GPCA classifier at different values of the valid pixel threshold T_v. The curves have been computed considering only the candidate faces produced by the hypothesis generation phase of the algorithm. As can be seen, performance deteriorates as T_v decreases. This is an obvious consequence due to the lack of information. Depending on the application requirements, T_v must be chosen accordingly.

Table 1 reports the results obtained when choosing a value for the classifier threshold T_f aimed at reducing false positives and a value of T_v of 0.5 (i.e. at least half of the face image must be non-occluded). In this case, tests were also performed on the original non-occluded test set. A fraction of 83% of the total number of occluded faces have been successfully detected, generating 43 false alarms. The results are satisfactory considering the toughness of the problem and the fact that a large number of the acquisitions would be missed using conventional 3D approaches. The detector performs very well on non-occluded faces, reaching 100% of the detected faces and just one false alarm.

Figure 14 shows some examples of correctly detected faces while Fig. 15 shows a subset of missed faces.
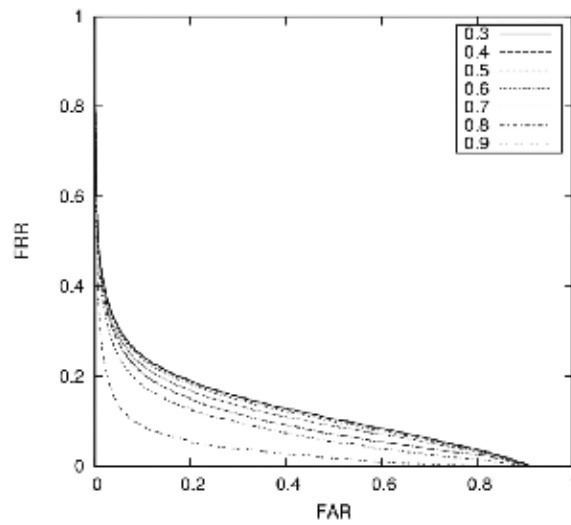
Fig. 13. The GPCA classifier ROC curves (FAR-False Acceptances Rate vs. FRR-False Rejections Rate) varying the value of the threshold T_v. and considering all the candidate faces from the 476 acquisitions of the test set.

Test Set	False Positives	False Negatives	Detected Faces
UND (non-occluded)	1	0	476/476 (100%)
UND (occluded)	43	77	399/476 (83.8%)

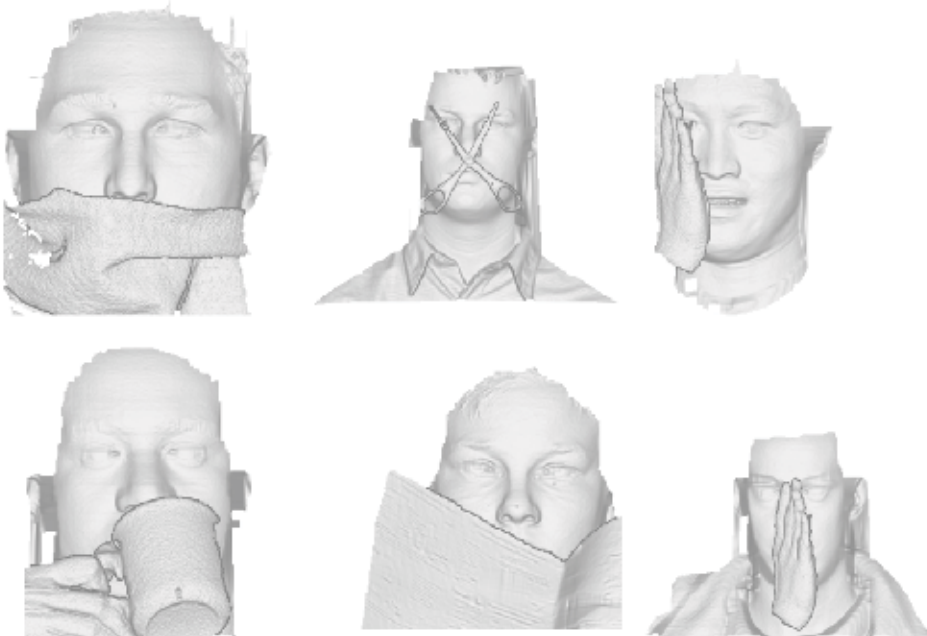Table 1. Face detection results obtained on 476 acquisitions.

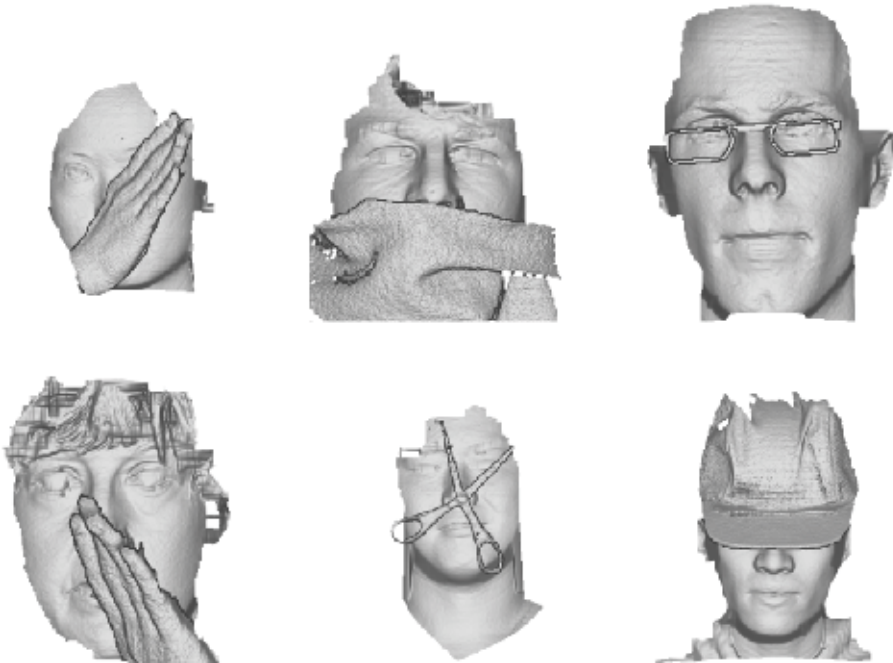Fig. 14. Examples of detected faces taken from the artificially occluded UND dataset.

Fig. 15. Examples of missed faces taken from the artificially occluded UND dataset.

4.3 Face recognition results

The occlusion detection, face restoration and the following recognition modules have been tested considering the set of detected faces. For feature extraction and matching we used the popular Fisherfaces approach.

First, an analysis has been made of how the values of the thresholds T_ρ (Equation 13) and T_τ (Equation 14) influence the accuracy in the detection of occlusions. The training set has been used to estimate the mean face $\vec{\mu}$ and to evaluate the distribution of the differences between the pixels of non-occluded faces and those of the mean face. Taking into account such a distribution, the threshold T_ρ has been set to be 15 mm, which is more than five times the estimated standard deviation of the distribution ($\sigma = 2.849$ mm). Less than 0.01% of the pixels of the training images exceed the mean face by that value.

To evaluate the accuracy of the occlusion detection method the procedure has been run varying the threshold T_τ. Since it is known exactly which parts of the faces are occluded, it is possible to compute the precision (fraction of true positives among the pixels detected as occluded) and the recall (fraction of occluded pixels which have been detected). The results, obtained with both automatic as well as manual normalized dataset, are reported in Fig. 16. As can be seen, though the normalization error degenerates the occluded pixel detection performances, the approach is still effective in the case of automatic systems.

On the basis of these measures the authors considered the value of 1.9 mm for the threshold T_τ to be a good compromise which has been set to this value for the rest of the experiments.

As a measure of the accuracy of the restoration method, the pixel by pixel absolute difference between the original (non-occluded) and the restored faces has been considered.
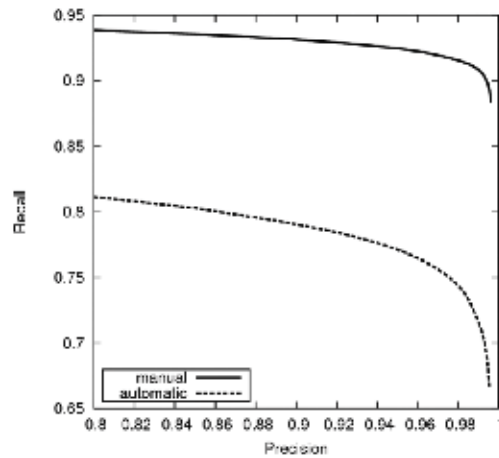
Fig. 16. Precision vs. recall corresponding to the pixels detected as occluded, obtained varying the value of the threshold T_r. Manual (476 acquisitions) and automatic (399 acquisitions) normalization.

Since the restoration accuracy is expected to be highly correlated to the extent of the occlusions, the results obtained as a function of the area of occluded regions are reported (see Fig. 17).
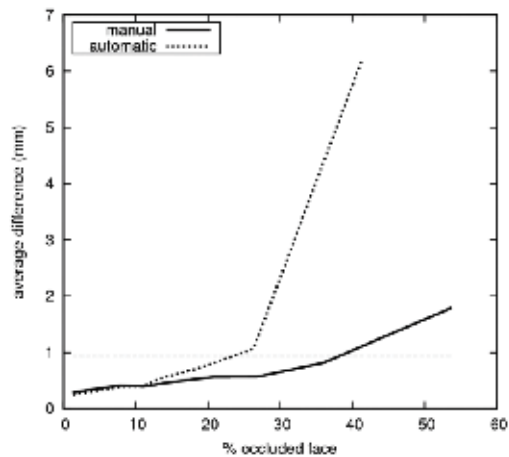
Fig. 17. Average absolute difference of the **z** coordinate between original and restored pixels, as a function of the fraction of the occluded face. Manual (476 images) and automatic (399 images) normalization. The horizontal line represents the average difference between two non-occluded acquisitions of the same subject, computed on manually normalized faces of the training set.

In Figure 18 shows two example of successful occlusion detection and restoration and an example of incorrect restoration.

However, since the error may be unevenly distributed over the faces, recognition performance could be affected more than expected. In order to ensure that a restored face may be reliably recognized, the same holistic recognition method has been applied to the original, the occluded, and the restored faces, and the performance has been compared in

the three cases. The Fisherfaces method (Belhumeur et al., 1997) has been adopted here, but any other method may be applied. Fisherfaces has been chosen for its popularity, and being a holistic approach, it is quite sensitive to normalization errors.
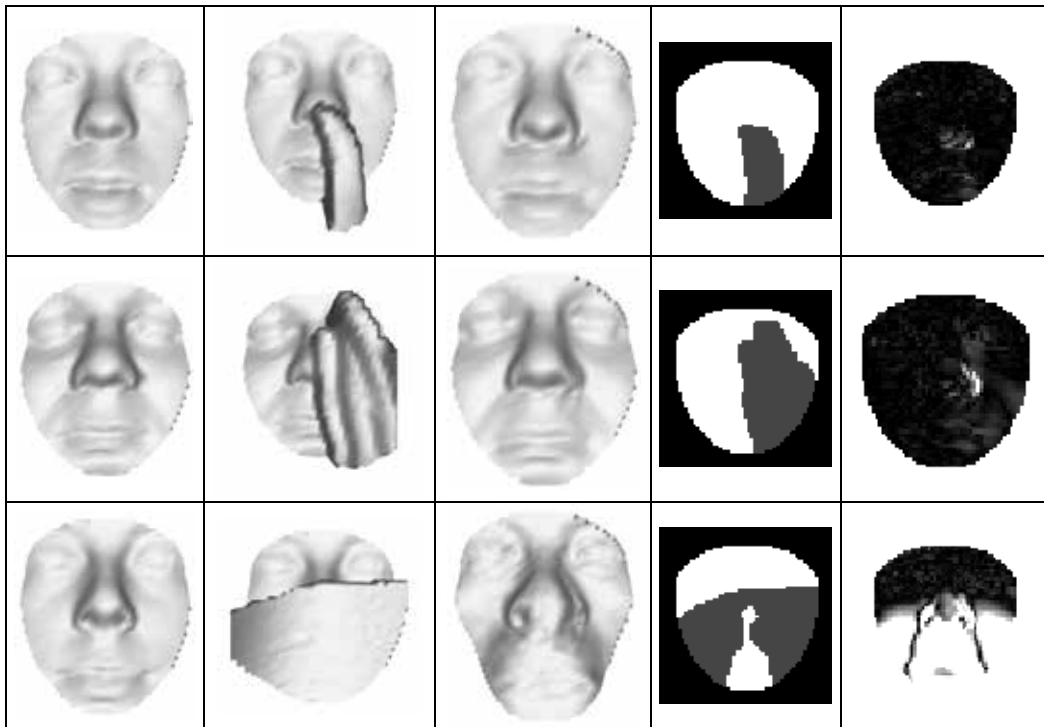

Fig. 18. First column: original faces. Second column: artificially occluded faces. Third column: restored faces. Fourth column: occlusion mask. Fifth columns: restoration error (brighter pixels indicate higher errors). The face in the third row represents a case of restoration failure.

We considered the identity verification scenario: the set of the remaining 475 non-occluded UND DB images not included in the test set have been used for building the known subject database and for training the Fisherfaces. Each test image has been proposed to the system, every time claiming a different identity (once for each subject included in the system DB).

Fig. 19 shows the ROC curves obtained on non-occluded, occluded, and restored faces in case of automatic detection and normalization. As expected, the recognition of occluded faces is very difficult: the Equal Error Rate (ERR) for occluded faces is 0.488, which is slightly better than random guessing. The application of the restoration strategy significantly improves the EER to a more acceptable 0.147, which is quite close to the 0.062 EER obtained on the original non-occluded faces.

In order to understand how much normalization errors degenerate results, we normalized the test set manually. Fig 20 shows the performances obtained with the manual procedure. As can be seen, normalization errors influence the system performances but the comparison of these results with those in Fig 19 shows the robustness of the automatic procedure.
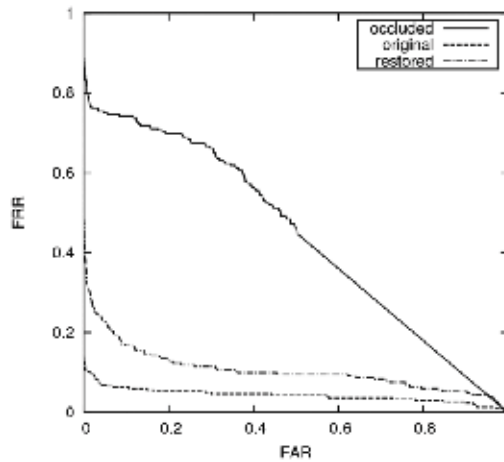
Fig. 19. ROC curves for the identity verification scenario. The curves refer to the performances obtained on the original (476 acquisitions), the artificially occluded and the restored test set (399 acquisitions). Images have been normalized automatically.
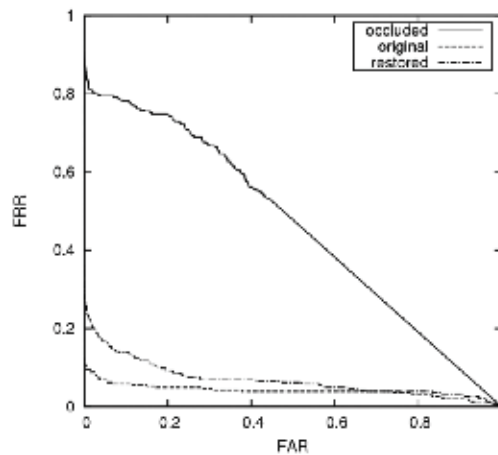
Fig. 20. ROC curves for the identity verification scenario. The curves refer to the performances obtained on the original (476 acquisitions), the artificially occluded and the restored test set (399 acquisitions). Images have been normalized manually.

To better understand the relationship between the occlusion extension and the recognition performance, the test set has been divided into three groups: 0-20%, 20-40% and over 40% (small, medium and large). Fig. 21 reports the ROC curves obtained using the three groups as test set. In the first case the EER is 0.07, which is comparable to the error rate measured on non-occluded faces. In the case of large occlusions, results are unacceptable (EER=0.5). For medium occlusions we obtained an EER of 0.27. In Fig 17 the restoration error begins to diverge approximatively around 30% of occluded face.

In conclusion the proposed face detection and normalization approach combined with the restoration module obtains promising results with occlusions smaller than approximatively 30% of the face.

Fig. 21. ROC curves for the identity verification scenario obtained on the set of 399 detected faces, subdivided into "small", "medium" and "large" subsets.

5. Conclusions and future works

The presence of occluding objects in face recognition and more generally in object recognition tasks, is a "far from solv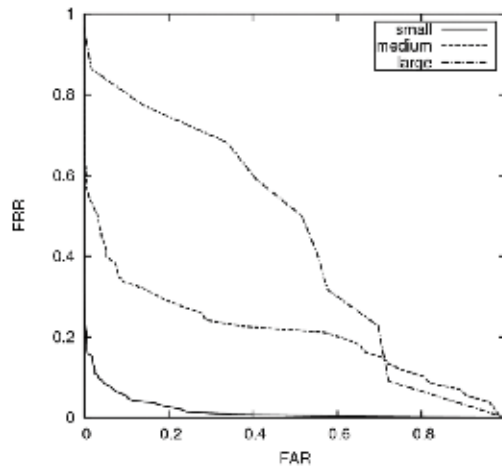ed" problem. Here a solution has been presented which is composed of three core modules (detection, normalization and occlusion detection/face restoration) that could be employed in any 3D recognition system in order to improve its robustness. The results are quite promising and indicate that the use of 3D data may simplify the problem.

However, there are still a lot of open issues. First of all, is it possible to reduce the errors introduced by the automatic normalization of faces? Secondly, how does the proposed algorithm perform in the presence of emphasized facial expressions? How could the proposed algorithm be integrated with a facial expression tolerant solution?

Another important issue regards the comparison between face restoration approaches versus partial matching approaches. It is not clear at present if the missing or covered parts of the face should be reconstructed or simply detected and ignored, letting a partial matching strategy perform recognition.

Future studies must take into account all these aspects. A large database containing real occlusions and various types and degrees of facial expressions should be adopted in order to allow an in-depth study of the problem and the proposed solutions.

8. References

Belhumeur, P.N., Hespanha, J.P. & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on PAMI*, pp. 711-720, 1997.

Besl, P. J. & McKay N. D. (1992). A method for registration of 3-d shapes. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14:239-256, 1992.

Chang, K, Bowyer, K. W. & Flynn, P. J. (2003). Face recognition using 2D and 3D facial data. *ACM Workshop on Multimodal User Application*, pp. 25-32, 2003.

Colombo, A., Cusano, C. & Schettini, R. (2006). 3D Face Detection using Curvature Analysis. *Pattern Recognition*, vol. 39, no. 3, pp. 444-455, 2006.

Colombo, A., Cusano, C. & Schettini, R. (2008a). Recognizing Faces In 3D Images Even In Presence Of Occlusions. *IEEE Second International Conference on Biometrics (BTAS08)*, submitted, 2008.

Colombo, A., Cusano, C. & Schettini, R. (2008b). Gappy PCA Classification for Occlusion Tolerant 3D Face Detection. *IEEE Transactions on systems, man and cybernetics–Part B*, submitted, 2008.

Colombo, A., Cusano, C. & Schettini, R. (2008b). Three-dimensional Occlusion detection and Restoration Of Partially Occluded Faces. *IEEE Transactions on multimedia*, submitted, 2008.

De Smet, M., Fransens, R. & Van Gool, L. (2006). A Generalized EM Approach for 3D Model Based Face Recognition under Occlusions. IEEE *Computer Society Conference on Computer Vision and Pattern Recognition* vol.2, no., pp. 1423-1430, 2006.

Everson, R. & Sirovich, L. (1995). Karhunen-Loève Procedure for Gappy Data. *J. Optical Soc. of America A*, vol. 12, no. 8, pp. 657-1664, 1995.

Gordon, G. (1991). Face recognition based on depth maps and surface curvature. *In Proceedings of SPIE, Geometric Methods in Computer Vision,* volume 1570, pages 234-247, 1991.

Hotta, K (2004). A robust face detector under partial occlusion. *Proceedings of ICIP 2004,* pp. 597-600, 2004.

Kim, J, Choi, J.,Yi, J., & Turk, M. (2005). Effective representation using ICA for Face Recognition Robust to Local Distortion and Partial Occlusion. *IEEE Trans. PAMI*, vol 27, no. 12, pp. 1977-1981, 2005.

Kirby, M. & Sirovich, L. (1990). Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Transactions on PAMI*, vol.12, no.1, pp. 103-108, 1990.

Lin, Yen-Yu, Liu, Tyng-Luh, & Fuh, Chiou-Shann (2004). Fast Object Detection with Occlusions. *The 8th European Conference on Computer Vision (ECCV-2004)*, Prague, May 2004.

Martinez, A. M. (2000). Recognition of Partially Occluded and/or Imprecisely Localized Faces using a Probabilistic approach. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 712-717, 2000.

Martinez, A. M. (2002). Recognizing Imprecisely Localized, Partially Occluded and Expression Variant Faces from a Single Sample per Class. *IEEE Trans. PAMI*, vol. 24, on. 6, pp. 748-763, 2002.

Park, J. S., Oh, Y. H., Ahn, S. C., & Lee, S. W (2005). Glasses Removal from Facial Image Using Recursive Error Compensation. *IEEE Trans. PAMI*, vol. 27, no. 5, pp. 805-811, 2005.

Rusinkiewicz, S., Levoy, M. (2001). Efficient Variants of the ICP Algorithm. *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, 2001.

A Model-based Approach for Combined Tracking and Resolution Enhancement of Faces in Low Resolution Video

Annika Kuhl, Tele Tan and Svetha Venkatesh
Curtin University of Technology
Australia

1. Introduction

Wide area surveillance situations require many sensors, thus making the use of high-resolution cameras prohibitive because of high costs and exponential growth in storage. Small and low cost CCTV cameras may produce poor quality video, and high-resolution CCD cameras in wide area surveillance can still yield low-resolution images of the object of interest, due to large distances from the camera. All these restrictions and limitations pose problems for subsequent tasks such as face recognition or vehicle registration plate recognition. Super-Resolution (SR) offers a way to increase the resolution of such images or videos and is well studied in the last decades (Farsiu et al., 2004; Park et al., 2003; Baker & Kanade 2002). However, most existing SR algorithms are not suitable for video sequences of faces because a face is a non-planar and non-rigid object, violating the underlying assumptions of many SR algorithms (Baker & Kanade, 1999).

A common SR algorithm is the super-resolution optical flow (Baker & Kanade, 1999). Each frame is interpolated to twice its size and optical flow is used to register previous and consecutive frames, which are then warped into a reference coordinate system. The super-resolved image is calculated as the average across these warped frames. However, the first step of interpolation introduces artificial random noise which is difficult to remove. Secondly, the optical flow is calculated between previous and consecutive frames preventing its use as an online stream processing algorithm. Also, accurate image registration requires precise motion estimation (Barreto et al., 2005) which in turn affects the quality of the super-resolved image as reported in (Zhao & Sawhney, 2002). Optical flow in general fails in low textured areas and causes problems in registering non-planar and non-rigid objects in particular. Recent techniques like (Gautama & van Hulle, 2002) calculate sub-pixel optical flow between several consecutive frames (with non-planar and non-rigid moving objects) however they are unable to estimate an accurate dense flow field, which is needed for accurate image warping.

Although solving all the issues in a general case is difficult, as the general problem of super-resolution is numerically ill-posed and computationally complex (Farsiu et al., 2004), we address a specific issue: Simultaneous tracking and increasing super-resolution of known object types, in our case faces, acquired by low resolution video. The use of an object-specific 3D mesh overcomes the issues with optic flow failures in low textured images. We avoid the

use of interpolation, and use this 3D mesh to track, register and warp the object of interest. Using the 3D mask to estimate translation and rotation parameters between two frames is equivalent to calculating a dense sub-pixel accurate optical flow field and subsequent warping into a reference coordinate system. The 3D mesh is subdivided, such that each quad is smaller than a pixel when projected into the image, which makes super-resolution possible (Smelyanskiy et al., 2000). It also allows for sub-pixel accurate image registration and warping and in addition, such a fine mesh improves the tracking performance of low-resolution objects. Each quad then accumulates the average colour values across several registered images and a high resolution 3D model is created online during tracking. This approach differs from classical SR techniques as the resolution is increased at the model level rather than at the image level. Furthermore only the object of interest is tracked and super-resolved rather than the entire scene which reduces computation costs. Lastly, the use of a deformable mask mesh allows for tracking of non-rigid objects.

The novelty of our approach is the use of an object-specific 3D model within a combined tracking and super-resolution framework which means that the super-resolved image is created during tracking. The 3D mesh model allows for accurate tracking of non-planar and low-resolution objects, and unlike optical flow based super-resolution methods our methods does not need initial resolution increase by interpolation, thus results in less blurred images. The resulting high-resolution 3D models can be used for a number of applications such as generating the object under different views and different lighting conditions.

2. Background and related work

Super-Resolution methods increase the resolution of a single image or a whole video sequence and can be formally treated as single frame or multi-frame approaches using spatial or frequency information (Huang & Tsai, 1984; Borman & Stevenson, 1998).

The simplest way of increasing the resolution of a single image is by interpolation using techniques like nearest neighbour or spline interpolation. However interpolation alone is not able to recover high-frequency details of the image or video and is therefore not truly regarded as `formal' SR (Park et al., 2003). More complex methods model the image formation process as a linear system (Bascle et al., 1996)

$$Y = AX + z \tag{1}$$

where X and Y is the high and low-resolution image respectively. The degrading matrix A represents image warp, blur and image sampling; z models the uncertainties due to noise. Restoring the high-resolution image X involves inverting the imaging process but this is computationally complex and numerically ill-posed, even though different constraints have been proposed in the last years (Baker & Kanade 2002).

SR methods that reconstruct super-resolved images from video sequences apply the image formation process of Equation 1 to several frames (Farsiu et al. 2004) or use a Bayesian approach to estimate images of higher resolution (Baker & Kanade 2002). Super-Resolution optical flow is another approach for combining several frames of a video sequence. But according to (Baker & Kanade 1999) most existing super-resolution algorithms are not suitable for video sequences of non-planar and/or non-rigid objects.

Pre-requisite for increasing the resolution of video sequences by combining several frames are sub-pixel shifts between consecutive frames. Typical SR techniques assume that the

A Model-based Approach for Combined Tracking and Resolution Enhancement
of Faces in Low Resolution Video
175

camera is moving as the scene is recorded. But a moving camera results in motion blur which decreases the quality of the images. Even though special cameras are able to increase the resolution of motion blurred images (Agrawal & Raskar 2007), according to (Ben-Ezra et al. 2005) traditional cameras should avoid motion blur as much as possible. However, in surveillance applications cameras are generally fixed and the object of interest is moving. We will capitalise on this arrangement to define the tracking and super-resolution requirements. This way the amount of motion blur can be avoided or is reduced to a minimum depending on the speed of the object and the frame rate of the camera.

Combining several frames to increase resolution requires accurate motion estimation techniques (Barreto et al. 2005) to register and warp consecutive frames into a reference coordinate system. Accurate image registration is important and does affect the quality of the SR results as shown in (Zhao & Sawhney 2002). SR optical flow uses optical flow to register consecutive frames and typically comprises the following five main steps

1. *Image Interpolation* - interpolate each frame to twice its size
2. *Image Registration* - estimate the motion field between consecutive frames
3. *Image Warping* - warp images into a reference coordinate system
4. *Image Fusing* - fuse images using mean, median or robust mean
5. *Deblurring* - apply standard deconvolution algorithms to super-resolved image

While this approach is well suited to increase the resolution of images of rigid and planar scenes, image registration is more difficult for non-rigid and non-planar low-resolution objects that are more subjected to occlusion and lighting changes. The first step of the SR optical flow algorithm interpolates each frame to twice its resolution using standard interpolation techniques like nearest neighbour or bilinear. But interpolation cannot recover high-frequency details in images. In addition it introduces artificial random noise that is difficult to remove in the deblurring step. Image warping, the third step, also involves interpolation of pixels which introduces further noise.

A similar approach which obtains the super-resolved texture during tracking is proposed by (Dellaert et al. 1998). This method tracks planar objects and predicts the super-resolved texture using a Kalman filter. In (Yu & Bhanu 2006) SR optical flow is extended and planar patches are used to track different parts of the face individually to account for non-rigidity. The resolution of the face is increased for these different facial parts individually but again no three-dimensional object-specific mask mesh is used. The authors of (Smelyanskiy et al., 2000) use a Bayesian approach for high-resolution 3D surface construction of low-resolution images given a user provided 3D model. Synthetic images are rendered using the surface model, compared with the real low-resolution images and the difference is minimised.

Recent studies involve the use of special cameras to capture super-resolved video sequences. The so called jitter camera is used in (Ben-Ezra et al. 2005) and creates sub-pixel offsets between frames during recording. They also show that motion blur degrades the result of super-resolution algorithms, even if the motion blur itself is known, and should therefore be avoided. The authors of (Agrawal & Raskar 2007) use a special so called flutter shutter camera. This camera preserves the high frequencies by opening and closing the shutter frequently during exposure. A single camera is extended to capture super-resolved stereo images in (Gao & Ahuja 2006). Our work, however, uses images captures by a standard digital camera.

3. Method overview

The image formation process and basic outline of our approach is illustrated in Figure 1. The degrading matrix A in Equation 1 models image warp, blur due to the optical system and motion as well as the down-sampling process caused by the finite and discrete imaging chip. We neglect the optical blur and keep the motion blur down to a minimum.
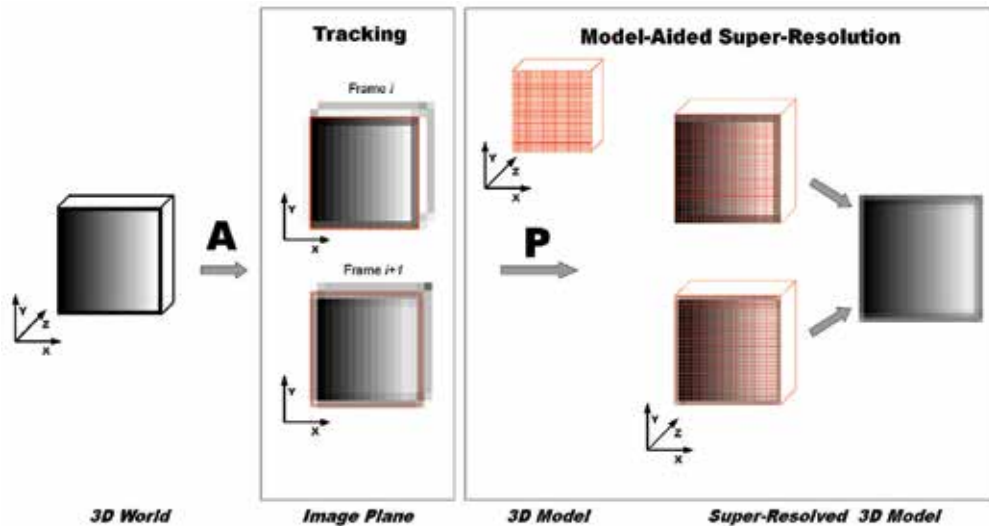

Fig. 1. Image formation process and basic outline of our approach. The matrix A degrades the images according to Equation 1. The 3D model of the object is textured by projecting every frame onto the model using the projection matrix P. The super-resolved texture is calculated as the mean across several frames.

During the image formation process the object is projected from the 3D world onto the image plane (Faugera 1993) and results, after being degraded by matrix A, in a finite number of discrete pixels as shown in Figure 1. The number of pixels that an object covers within the image depends on the size of the imaging chip, the optical lens, the size of the object itself and the distance between object and camera. As the camera or the object moves it may be projected onto different pixels of the imaging chip in different frames.

The black edges surrounding the gradient on the front side of the cube in Figure 1 are projected nearly exact into pixel centres resulting in 14 black pixels on either side of the cube in the image plane in frame *i*. A movement of the cube in front of the camera results in sub-pixel movements on the image plane. The black edges of the cube then may fall between pixels of the imaging chip resulting in grey edge pixels in frame *i+1* in Figure 1.

The tracking algorithm uses an object-specific deformable 3D model mesh to estimate translation and rotation parameters between consecutive frames which allows for accurate tracking of non-planar and non-rigid objects. Computer graphic techniques are used to subdivide the 3D model such that every quad of the mesh is smaller than a pixel when projected into the image, which makes super-resolution possible. By projecting the 3D model in several frames each quad accumulates different colour values over time. The super-resolved 3D model is then calculated as the mean colour value for each quad. Without loss of generality Figure 1 only shows the projection of one side of the planar cube; the same would be true for a non-planar and/or non-rigid object.

A Model-based Approach for Combined Tracking and Resolution Enhancement
of Faces in Low Resolution Video
177

The super-resolved 3D model is created online during tracking and improves with every frame, whereas super-resolution optical flow incorporates consecutive and previous frames which prohibits its usage as an online stream processing algorithm. Furthermore, using an object-specific 3D model in a combined tracking and super-resolution approach inverses the image formation process in Equation 1. The subdivided 3D mesh represents the high-resolution object X that is down-sampled by projection into the image plane. The finer the mesh the higher the resolution of X and the larger the possible increase in resolution. Thus, interpolation, the first step of the optical flow algorithm, is unnecessary and the resulting super-resolved 3D model is less blurred by achieving the same resolution increase. This in turn makes deblurring (the last step of the optical flow algorithm) unnecessary. Lastly, using the 3D mesh for tracking equals image registration, warping and the estimation of a dense flow field, comprising steps 2 and 3 of the optical flow algorithm.

Furthermore we present an extended tracking approach that allows for the non-rigidity of objects during tracking. By incorporating a deformable mask mesh that allows for deformations during the tracking process super-resolution is possible despite non-planarity and non-rigidity.

4. Combined tracking and super-resolution

4.1 3D object tracking

A pre-requisite for building our super-resolved 3D model is the need to track the object in low-resolution video. We utilise an object-specific 3D mesh model which is manually fitted in the first frame. For fully automatic tracking, a combined appearance and geometric based approach similar to that in (Wen & Huang 2005) is used. We differ in that we use a subdivided fine mesh for the appearance approach and a constrained template matching algorithm for the geometric tracking as opposed to minimising a linear combination of action units. For each frame we apply both methods, i.e. the appearance based and the geometric based tracking. The one that results in the smallest tracking error will be used for the current frame. We describe each tracking approach in detail.

The **appearance based approach** is similar to the one in (Wen & Huang 2005). We differ in that we use a subdivided mask mesh which achieves better tracking results than a mesh that is coarser with respect to the pixel size. The warping templates b_i are created from this subdivided mask as

$$b_i = I_0 - Q\big(\mathrm{P}\big(T_0 + n_i, X\big)\big) \text{ with } I_0 = \mathrm{P}\big(T_0, X\big) \tag{2}$$

where function P is the projection of 3D object points X to image coordinates using the initial transformation T_0; Q then maps RGB values to these coordinates. X is a vector containing the centre of gravity of each mask triangle, n_i is the transformation parameter displacement and I_0 is a vector of the concatenated RGB-values of each projected triangle. The required intrinsic camera transformation parameters are obtained using camera calibration techniques as in (Zhang, 2000).

Objects are tracked by using the pose parameters of the previous frame as initialisation and solving for q and c for each frame i

$$I_0 - Q\big(\mathrm{P}\big(T_i^{app}, X\big)\big) \approx Bq + Uc \tag{3}$$

where the columns of B are the warping templates b_i. T_i^{app} contains the transformation parameters for the appearance-based tracking at frame i and U are the illumination templates. Like in (Wen & Huang 2005) we use the first nine spherical harmonic bases, given the 3D mask mesh, for modelling lighting changes during tracking. Please refer to (Cascia et.al., 2000) for more details on the appearance based approach.

The **geometric based approach** uses a standard template matching approach that is restricted by the object-specific mask. We do not use action units to track facial movements like in (Wen & Huang 2005). Objects are tracked by projecting every vertex V of the mask into the previous image, using perspective projection. Around each projected vertex a rectangular template is cropped and matched with the current frame. The size of the patch is set to 1/6th of the whole object. Normalised cross-correlation is used to match this patch in the current frame within a window that is double the size of the template. In order to minimise the effect of outliers, the entire mask is fitted to the retrieved new vertex points v_i in the current frame utilising the Levenberg-Marquardt algorithm

$$T_i^{geo} = \min_{T^{geo}} \sum_{j=1}^{l} \left(P\left(T_i^{geo}, V_j\right) - v_j \right)^2 \qquad (4)$$

where l is the number of mask vertices V and T_i^{geo} contains the transformation parameters of the geometric-based approach at frame i. Again the transformation parameters are initialised with the previous frame.

During tracking each method is applied individually and a texture residual as the root mean squared error (RMSE) for the current frame i with respect to the first frame is calculated

$$RMSE(T_i) = \sqrt{\frac{1}{k} \sum_{j=1}^{k} \left(P\left(T_0, X_j\right) - P\left(T_i, X_j\right) \right)^2} \qquad (5)$$

where k is the number of mask triangles X. The pose parameters T_i of the method with the smallest RMSE will be used for the current frame i as

$$T_i = \min_{T_i} \left[RMSE\left(T_i^{app}\right), RMSE\left(T_i^{geo}\right) \right] \qquad (6)$$

where $RMSE\left(T_i^{app}\right)$ and $RMSE\left(T_i^{geo}\right)$ is the texture residual for frame i of the appearance-based and the geometric-based approach respectively. The tracking runs automatically once the mesh mask is manually fitted to the first frame of the sequence.

4.2 3D model-aided super-resolution

During tracking the resolution of the low-detailed object is gradually increased. To achieve this, every triangle of the object-specific mask is projected into the video using perspective projection. But in order to increase the resolution of the object, every triangle needs to be smaller than a pixel (Smelyanskiy et al., 2000).

As each mask triangle is projected into different frames of the sequence it is eventually assigned with different colour values for each frame as shown in Figure 1. Therefore the super-resolved mask I_{SR} is calculated as the mean of the last k frames that result in an RMSE below a certain threshold ε

$$I_{SR} = \frac{1}{k} \sum_{i=1}^{k} \left(P\left(T_i, X\right) \right) \text{ with } RMSE\left(T_i\right) < \varepsilon \qquad (7)$$

Small tracking errors (RMSE) allow for an exact alignment of the 3D mask across frames whereas high RMSE result in blurring and distortion. The threshold ε depends on the initial object resolution. Low-resolution objects usually result in higher RMSE during tracking as image pixels are more likely to change due to the down-sampling process of the imaging chip. Furthermore the quality of the super-resolved mask I$_{SR}$ also depends on the total number of frames k used. But a larger number of frames increases the probability of introducing artificial noise as frames might not be aligned perfectly. The issue of choosing the appropriate number of frames k versus the quality of the super-resolved mask is experimentally evaluated in Section 4.4.

4.3 Extension to non-planar and non-rigid objects

In order to increase the resolution of various non-planar and non-rigid objects the tracking algorithm also needs to allow for deformations, i.e. the mask mesh representing the three-dimensional object needs to be deformable. This is especially an issue when tracking non-rigid objects like faces. We therefore propose an extended tracking and super-resolution algorithm and apply it to faces, as faces are a major field of interest especially in wide area surveillance systems.
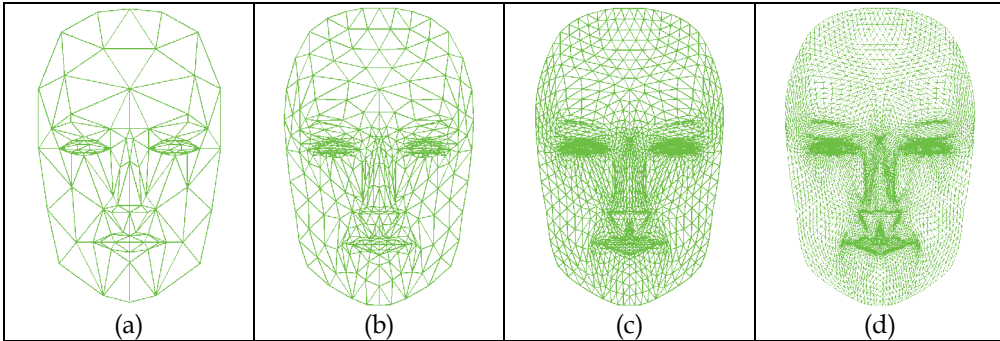

Fig. 2. (a) CANDIDE-3 face mask with 184 triangle, (b), (c) and (d) are subdivided masks after 1, 2 and 3 subdivision steps resulting in 736, 2944 and 11776 triangles. Source: (Kuhl et al.,2008) © 2008 IEEE

For tracking faces the CANDIDE-3 face model is used (Ahlberg, 2001). As shown in Figure 2 this triangular mesh consists of 104 vertices and 184 triangles and it is subdivided three times using the Modified Butterfly algorithm (Zorin & Schröder, 2000) to finally produce 5984 vertices and 11776 triangles. To allow for the non-rigidity of faces we use the CANDIDE-3 expression parameters for tracking mouth and eyebrow movements in low-detailed faces. More complex facial expressions often require a more detailed face model and high-resolution images, such as in (Goldenstein et al., 2003; Roussel & Gagalowicz, 2005; Wang et al., 2005).

The expression tracking is performed after the actual tracking for each frame. Using the expression parameters of the last frame the combined geometric and appearance based tracking approach is used to determine the position of the mesh model in the current frame. After that a global random search (Zhigljavsky, 1991) is performed to improve the RMSE around the mouth and the eyebrow region. A normal distribution with respect to the last

expression parameters is used to sample 10 to 20 different values for each expression parameter. The parameter that results in the smallest RMSE is chosen for the current frame.

5. Experiments

5.1 Combined geometric and appearance-based 3D object tracking

(a) 230x165   (b) 115x82   (c) 57x41   (d) 28x20

Fig. 3. Cropped faces for each face size used. Source: (Kuhl et al.,2008) © 2008 IEEE

In order to evaluate the tracking accuracy of the combined geometric and appearance based tracking algorithm, a video sequence of a face with translation and rotation movements is recorded at 15 frames per second and an initial resolution of 640x480 pixels. The face within one frame has an average size of 230x165 pixels. This resolution is divided into halves three times, resulting in face sizes of 115x82, 57x41 and 28x20 pixels with corresponding frame sizes of 320x240, 160x120 and 80x60 respectively. A cropped face for each face size used is shown in Figure 3.

(a)   (b)
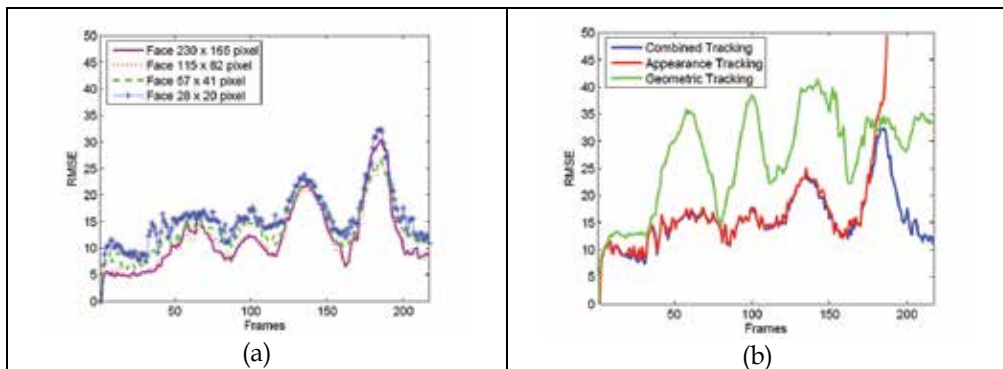
Fig. 4. Tracking results for different size faces (a) and each tracking method applied individually to the video with the smallest resolution (b). Source: (Kuhl et al.,2008) © 2008 IEEE

For tracking faces we use the CANDIDE-3 face model (Ahlberg, 2001) as shown in Figure 2. In order to initialise this mask in the first frame of the sequence, we use the shape parameters of the CANDIDE-3 model to adjust the mask according to the persons individual face. After this initialisation, the mask is tracked automatically over more than 200 frames using the combined geometric and appearance-based approach as described in Section 4.1.

The results of the combined tracking algorithm applied to different face sizes is shown in Figure 4(a). The RMSE for measuring the tracking accuracy with respect to the first frame is defined in Equation 5. The variation in RMSE in frames 1 to 100 are due to translation and rotation around the horizontal x-axis, whereas the peaks at frames 130 and 175 respectively are mainly due to rotation around the vertical y-axis.

Faces between 230x165 and 115x82 result in similar RMSE, whereas faces with a resolution down to 57x41 result in a slightly increased tracking error, that is 22% larger on average. Even though the RMSE increased by about 41% when tracking faces with a resolution of 28x20, the algorithm is still able to qualitatively track the face to the end of the sequence.

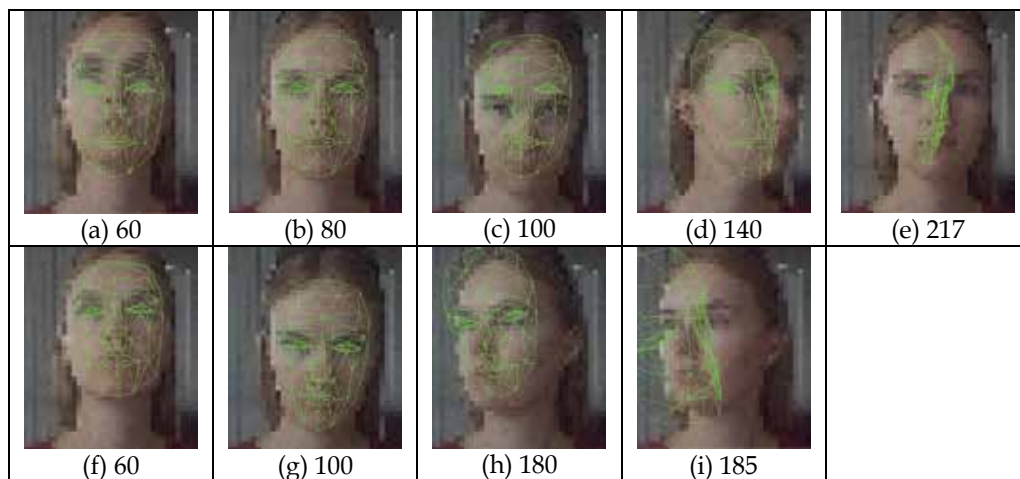| (a) 60 | (b) 80 | (c) 100 | (d) 140 | (e) 217 |
| (f) 60 | (g) 100 | (h) 180 | (i) 185 | |

Fig. 5. Sample frame shots of geometric based tracking (a)-(e) and appearance based tracking (f)-(i). Numbers denote different frames.

In comparison, Figure 4(b) shows the result of the geometric and appearance based tracking approach operating individually on the same video sequence, with the smallest face size of 28x20, as this face size is the most difficult to track. The geometric approach loses track just after 40 frames and as shown in Figure 5(a) this is due to small inter frame movements which causes the mask to stay in the initial position instead of following the face. The mask then recovers in frame 80 and loses track immediately afterwards as shown in Figure 5(b) and Figure 5(c). The geometric approach finally loses track around frame 140, from which it can not recover, as shown in Figure 5(d) and Figure 5(e). This shows that the geometric approach is not able to handle small inter frame movements due to image noise and low resolution.

The appearance based approach on the other hand results in small tracking errors from frame 1 through to frame 170 as shown in Figure 5(f) and Figure 5(g). But as the face turns the appearance approach loses track from which it cannot recover, as shown in Figure 5(h) and Figure 5(i). This is mainly due to large inter frame movements and the rotation of the face, resulting in partial occlution of the face.

Figure 4(b) shows that by combining the geometric and appearance based approach tracking is improved. Both approaches complement one another resulting in smaller RMSE than either of them individually. While the appearance based method tends to be more precise for small inter-frame movements, the geometric method is better for larger displacements. Furthermore the geometric approach applies template matching between the current and

the previous frame, while the appearance approach is based on the comparison of the current frame with the first frame. Thus, the combination is more stable and precise and able to track even small size faces down to a size of 28x20 pixels.

5.2 Combined tracking and super-resolution

We tested the performance of the combined geometric and appearance-based tracking algorithm with different mask sizes. As described in Section 3, super-resolution is only possible when the mask mesh is subdivided such that every quad or triangle is smaller than a pixel when projected into the image. The following experiment evaluates the effect of the mesh size on the tracking performance.

(a) Example Frame   (b) Object Size 31x31

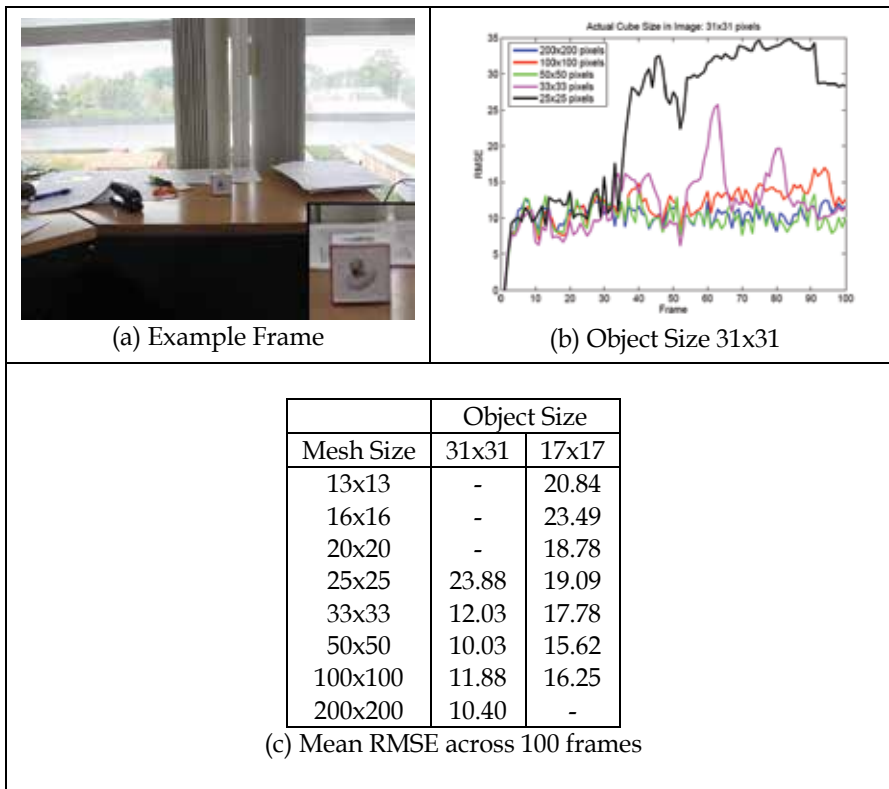|	Object Size	
Mesh Size	31x31	17x17
13x13	-	20.84
16x16	-	23.49
20x20	-	18.78
25x25	23.88	19.09
33x33	12.03	17.78
50x50	10.03	15.62
100x100	11.88	16.25
200x200	10.40	-

(c) Mean RMSE across 100 frames

Fig. 6. Mean tracking RMSE across 100 frames for a planar objects, the front side of a cube (a). Using an object mask mesh that is finer than the object results in smaller tracking errors.

The front side of a cube as shown in Figure 6(a) is tracked across 100 frames of size 640x480. This planar patch covers an area of 31x31 pixels within the image. For tracking this patch we used a 3D model mesh similar to the one in Figure 1. This mesh is equally subdivided into 25x25, 33x33, 50x50, 100x100 and 200x200 quads. Figure 6(b) shows the result of the combined tracking approach when different mesh sizes were used.

For tracking a planar patch of size 31x31, the best result is achieved with mesh sizes larger than 33x33, whereas further subdivision does not improve the tracking. Using a mesh that is coarser with respect to the pixel size loses track easily and results in higher RMSE during tracking as shown in Figure 6(b). Such a coarse mesh is a under representation of the object,

A Model-based Approach for Combined Tracking and Resolution Enhancement
of Faces in Low Resolution Video
183

resulting in higher tracking errors. By using a larger number of quads the mesh is able to account better for appearance changes due to sub-pixels movements.

Another video of the same object was recorded at greater distances, resulting in a cube of size 17x17 pixels. The mask mesh used for tracking consists of 13x13, 16x16, 20x20, 25x25, 33x33, 50x50 or 100x100 quads. The corresponding mean tracking errors are shown in Table 6(c). Again the best tracking results are achieved with a mask mesh that contains more quads than the pixels covered by the object within the image.

This shows that not only the super-resolution benefits from the tracking algorithm but also the super-resolution benefits tracking. The combined geometric and appearance-based tracking approach achieves best results when a fine model mesh is used. This fine mesh should ideally be subdivided such that every quad or triangle is smaller than a pixel when projected into the image. In practice a mesh that is double the size has proven to be the best trade-off between accuracy and speed.

5.3 Expression tracking

In order to evaluate the performance of the expression tracking approach, we recorded a video of a face with a resolution of 230x165 with mouth and eyebrow movements. One frame of this sequence is shown in Figure 7(a). The graph in Figure 7(b) compares the result of the expression tracking with the combined geometric and appearance based tracking approach without expression tracking. Frames 10 to 22 contain mouth openings and frames 28 to 38 contain eyebrow movements. The graph shows clearly that the expression tracking improves the result of the combined tracking approach by reducing the RMSE for each frame.

(a)

(b)

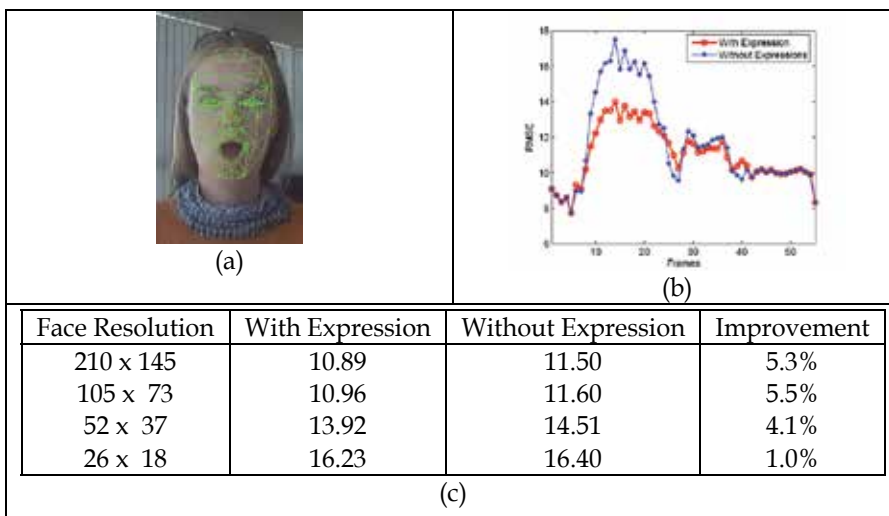| Face Resolution	With Expression	Without Expression	Improvement
210 x 145	10.89	11.50	5.3%
105 x 73	10.96	11.60	5.5%
52 x 37	13.92	14.51	4.1%
26 x 18	16.23	16.40	1.0%

(c)

Fig. 7. Results of the expression tracking, (b) compares the tracking RMSE with and without expression tracking, (c) shows the mean tracking RMSE of 56 frames for different face resolutions.

Furthermore we again cut the resolution of the video in halves three time resulting in face resolutions of 210x145, 105x73, 52x37 and 26x18 pixels in size. The mean tracking error across 56 frames for each resolution is shown in Table 7(c). While the difference in RMSE

amounts to about 0.60 (between 5.5% and 4.1% improvement factor) for the first three resolution levels, a face resolution of 26x18 only results in a RMSE difference of 0.17 compared to the tracking approach without expressions. This equals an improvement factor of only 1.0%.

The smaller the resolution of the face, the larger the RMSE as shown in Figure 4(a) and Figure 7(c). A face that is captured in high-resolution, results in a large number of pixels representing this face. But the smaller the number of representative pixels, the more likely they are to change over time. Due to the discretisation of the imaging sensor certain face regions, e.g. the eyes, result in very few pixels being allocated to them. The colour value of these pixels is most likely to change over time as the camera or the face moves and these areas fall on or between different image pixels. Therefore, tracking expressions of low-resolution faces does not improve the overall RMSE significantly.

5.4 3D model-aided super-resolution

In order to increase the resolution of the object of interest the object-specific 3D model must be subdivided into a fine mesh. Each quad or triangle must me smaller than a pixel when projected into the image to make super-resolution possible as illustrated in Figure 1. The possible increase in resolution depends on the size of the 3D model mesh. The finer the mesh, the larger the possible increase in resolution but however more frames are needed.

The following experiments quantitatively evaluate the number of frames needed to achieve different resolutions. We therefore tracked a little cube across more than 200 frames of a video sequence recorded at a resolution of 320x240 with the cube of size 95x95. This video is sub-sampled three times resulting in resolutions of 160x120, 80x60, 40x30 and corresponding cube sizes of 48x48, 24x24, and 12x12 respectively. In order to diminish the effect of tracking errors the cube is tracked at the highest resolution of 95x95. The estimated pose parameters are then used for the cube of size 24x24 and 12x12. An example frame of both these resolutions is shown in Figure 8(a) and Figure 8(c) respectively. For comparison Figure 8(b) and Figure 8(d) show these frames after their resolution is doubled using bilinear interpolation.

(a)   (b)   (c)   (d)

Fig. 8. Example frame of resolution (a) 80x60 and (c) 40x30 with cube sizes of 24x24 and 12x12 respectively. Figure (b) and (d) show these frames after they have been doubled in size using bilinear interpolation.

For increasing the resolution of the cube of size 24x24 we used a 3D model as shown in Figure 1 and subdivided it into 25x25, 50x50 and 100x100 equal size quads. This mesh is projected into every frame of the sequence and the super-resolved 3D model I_{SR} is created according to Equation 7 by combining 1, 10, 20, 50, 100 or 200 frames with the results shown in Figure 9.

Using a mesh of size 25x25 cannot increase the resolution of a cube of size 24x24. Although, calculating the mean across about 20 frames removes the noise of the camera and partially

recovers the eyes of the duck that are not visible in the first frame as shown in Figure 9(c). Using a mesh that is double the size (50x50) results in a more detailed image but after about 20 to 50 frames the maximal possible resolution is achieved and adding more frames does not improve the resolution further as shown in Figure 9(i) and Figure 9(j) respectively.
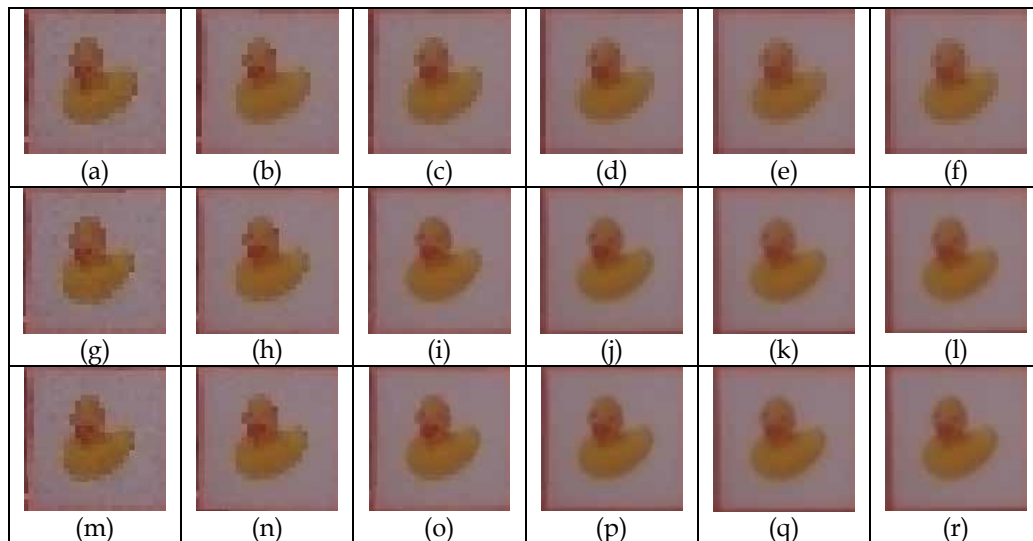

Fig. 9. Super-Resolution results of a cube of resolution of 24x24 pixels using mesh sizes of 25x25 (top), 50x50 (middle) and 100x100 (bottom) after 1, 10, 20, 50, 100 and 200 frames respectively.

Using a mesh of size 100x100 equals a possible resolution increase of four times in each dimension. But to achieve this increase at least 50 frames are needed as shown in Figure 9(p). But taking the mean across such a large number of frames also introduces noise as shown in Figure 9(r). This noise results from slightly misaligned frames and from the averaging process itself.

The same experiment is done for the cube of size 12x12 using mesh sizes of 20x20, 40x40 and 80x80. The result after combining 1, 10, 20, 50, 100 and 200 frames is shown in Figure 10. Using the mesh of size 20x20, which equals a resolution increase of 166%, requires about 20 to 50 frames (Figure 10(c) and Figure 10(d)). Adding more frames does not improve the result further.

Using a mesh of 40x40 results in a possible resolution increase of 3.3 times in each dimension and about 100 frames are needed to achieve this increase as shown in Figure 10(k). Trying to increase the resolution 6.6 times requires a mesh size of 80x80 and about 200 frames as shown in Figure 10(r). Even though the outer shape of the duck is recovered in greater detail, the overall result is very noisy and blurry as a result of taking the mean across 200 frames.

In practice it is therefore not recommended to increase the resolution of an object by more than 2 to 3 times. The higher the increase in resolution the higher the number of frames needed to achieve this resolution which in turn results in more noise. It is therefore a trade-off between possible resolution increase and number of frames. Furthermore the tracking error ε in Equation 7 also influences the resulting super-resolution image. Large tracking errors lead to a misalignment of frames resulting in noisy and blurred super-resolution images.
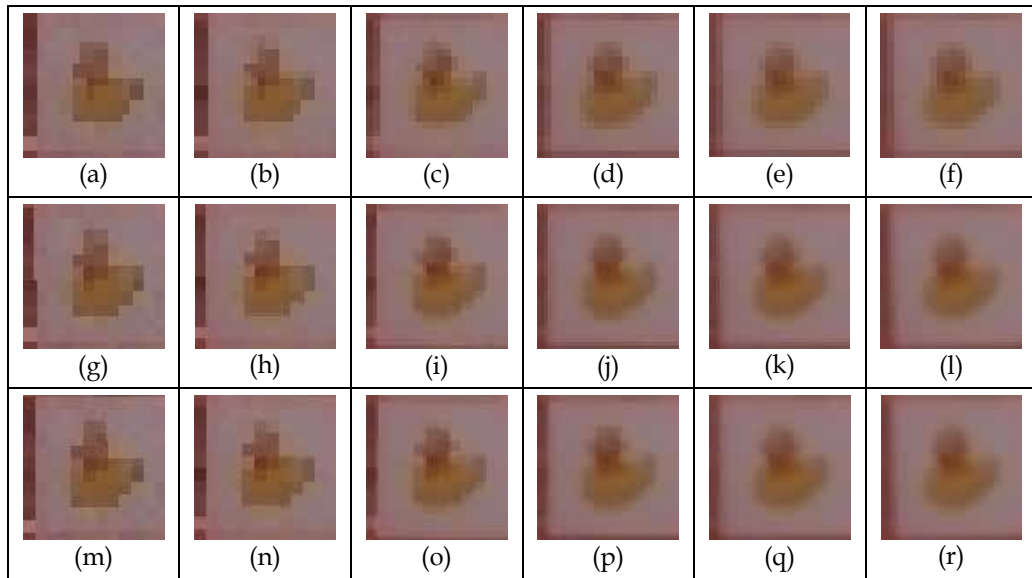
Fig. 10. Super-Resolution results of a cube of resolution of 12x12 pixels using mesh sizes of 20x20 (top), 40x40 (middle) and 80x80 (bottom) after 1, 10, 20, 50, 100 and 200 frames respectively.

Simple objects like a cube allow for an equal subdivision of the 3D model mesh. However, more complex objects may require a 3D model that consists of different size quads or triangles, thus resulting in a varying resolution increase across the mask mesh. For increasing the resolution of faces, the CANDIDE-3 mask as shown in Figure 2 is used. This mask is finely sampled around the eyes, the mouth and the nose region. These areas are also the most textured and the most important part of the face and therefore a finer sampled mask allows for a larger increase in resolution in these areas.

Evaluating the number of frames needed to achieve a certain resolution is more difficult using such a complex mask as the resolution increase varies across the entire mask due to different size triangles. In order to evaluate the minimum number of frames needed to create a face mask of higher resolution, faces are tracked in videos with minimal head movements in order to keep the tracking error to a minimum. The average sizes of the faces are 60x40 and 30x20 using videos of resolution 160x120 and 80x60 respectively. These resolutions are sub-sampled from the original video sequence recorded with 640x480. The mask mesh is manually fitted to the first frame of each sequence and then tracked fully automatically across more than 200 frames.

The super-resolved mask I_{SR} is calculated using between 1 to 200 frames with the smallest tracking RMSE according to Equation 7. The mean tracking RMSE amounted to 10.9 and 14.9 for faces of size 60x40 and 30x20 respectively. We use the CANDIDE-3 mask that is subdivided three times as shown in Figure 2(d) for both face sizes. The high-resolution mask created from the 30x20 faces is then compared with a single frame of double the resolution (60x40) and the mask created from the 60x40 faces is compared with the face of 120x80 respectively. We use the mean colour difference E_{colour} to compare two face masks I_1 and I_2 consisting of k triangles each

$$E_{colour} = \frac{1}{k} \sum_{i=1}^{k} \left| I_1(i) - I_2(i) \right|_2 \tag{1}$$

Figure 11 shows the results for six different persons. Common to all persons and face sizes is the strong error decrease within the first 20 to 30 frames. Within the first 20 frames faces of size 60x40 increase resolution most significantly with respect to a face of double the size. Faces of size 30x20 are smaller and therefore more frames are needed to achieve the same resolution increase using a mask mesh with the same number of triangles. After about 20 to 30 frames the resolution increases most significantly. These results are comparable to the results of the cube shown in Figure 9 and Figure 10.
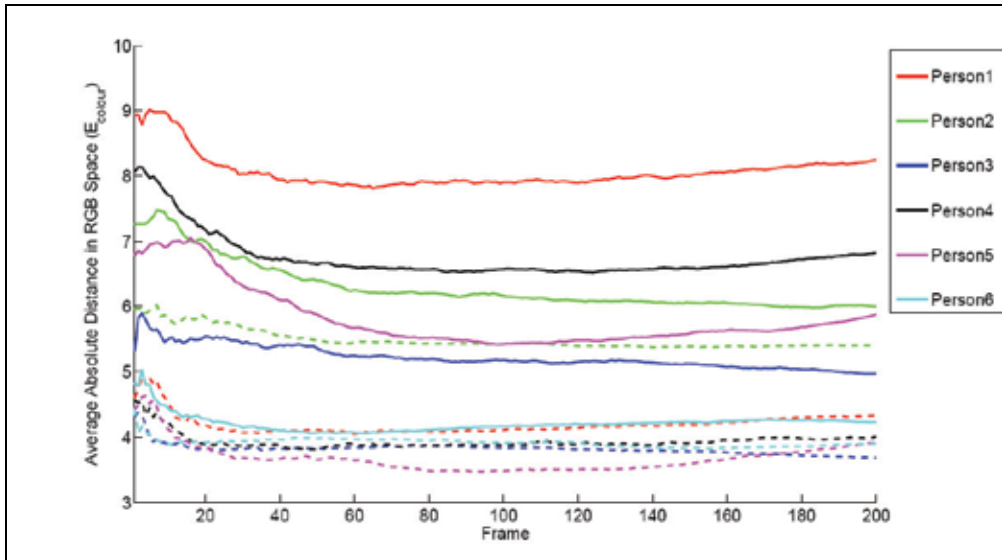

Fig. 11. Quality of the super-resolved 3D face mask using different resolutions (60x40, dotted line and 30x20 solid line) and number of frames (x-aches). Source: (Kuhl et al.,2008) © 2008 IEEE

For a qualitative comparison the super-resolution result of person 4 is shown in Figure 12. The faces on the left show the facial mask that is textured with a single frame of face size 60x40 (Figure 12(a)) and 30x20 (Figure 12(f)) respectively. The second, third and fourth collumn show the increase in resolution after 20 (Figure 12(b) and Figure 12(g)), 50 (Figure 12(c) and Figure 12(h)) and 200 (Figure 12(d) and Figure 12(i)) frames have been added to the super-resolved mask.

As the first step of the super-resolution optical flow algorithm is to double the size of the input images using interpolation techniques (see Section 2), we used bilinear interpolation to increase the size of the input video. The result after combing 200 frames of the interpolated input frames is shown in Figure 12(e) and Figure 12(j) for face sizes of 60x40 and 30x20 respectively. Even though the input images are doubled in size the resulting super-resolved faces show less detail and are more blurred. Interpolation does not recover high-frequencies and on the contrary introduces further noise; it should therefore be avoided during the super-resolution process.
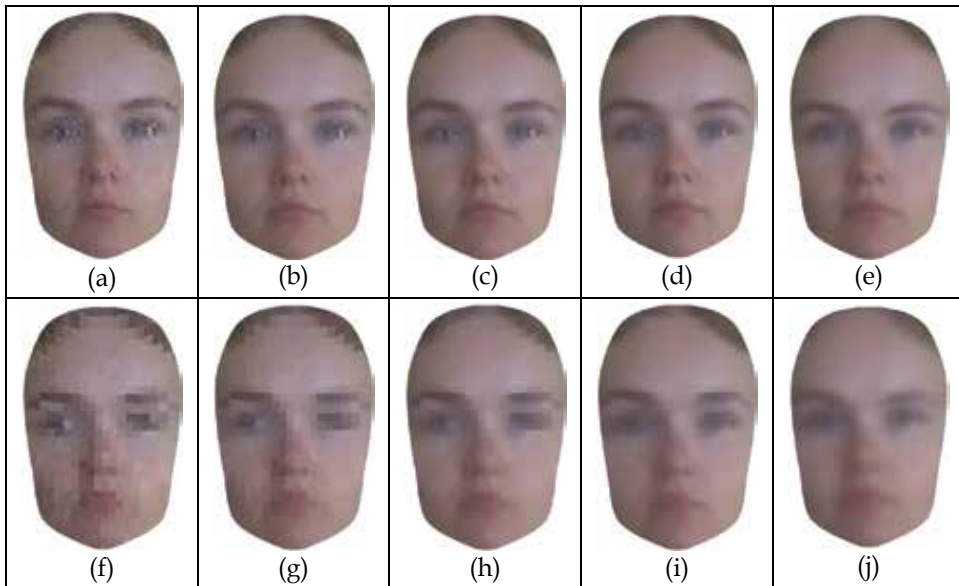
Fig. 12. Results of the combined tracking and super-resolution approach for face sizes 60x40 (top) and 30x20 (bottom) after 1, 20, 50 and 200 frames respectively. The last column shows the result after 200 frames when the input images are interpolated to double the size using bilinear interpolation. Source: (Kuhl et al.,2008) © 2008 IEEE

5.5 Comparison with super-resolution optical flow

We are comparing our approach with super-resolution optical flow by applying both algorithms to video sequences recorded in our lab as well as surveillance video of faces. Our implementation of the super-resolution optical flow follows the first four steps as outlined in Section 2. We abstained from using deconvolution techniques as this is an additional option for both our approach and the optical flow method to further increase the quality of the super-resolved images. The optical flow between consecutive frames is calculated using (Gautama & van Hulle, 2002) and the mean is used to calculate the super-resolved image.

At first we recorded a video sequence of a cube at 15 frames per second and with a resolution 320x240, where the cube is about 100x100 pixels in size. The proposed combined geometric and appearance based approach is used to track this cube across more than 100 frames resulting in a mean tracking error of 9.16. The pose parameters are then used to project the 3D model into every image and k frames with the smallest tracking error are used to calculate the super-resolved image I_{SR} according to Equation 7.

The 3D model of the cube is subdivided into 400x400 quads before projected into the image which, under ideal conditions, equals a resolution increase of 400%. The result is shown in Figure 13. After about 20 frames (Figure 13(d) and Figure 13(j)) the maximum resolution increase is reached and further added frames result in increased blur due to tracking errors. For comparison we doubled the size of each frame using bilinear interpolation before creating the super-resolution image. Again a cube with 400x400 quads is used and the result is shown in Figure 13(m) to Figure 13(r). Even though the input images are doubled in size the resulting super-resolution images after about 20 frames (Figure 13(p)) do not show a significant resolution increase compared to the one without initial interpolation (Figure

13(j)). On the contrary, like in Figure 12, the resulting super-resolution images show a greater amount of blur as interpolation cannot recover high-frequency details.
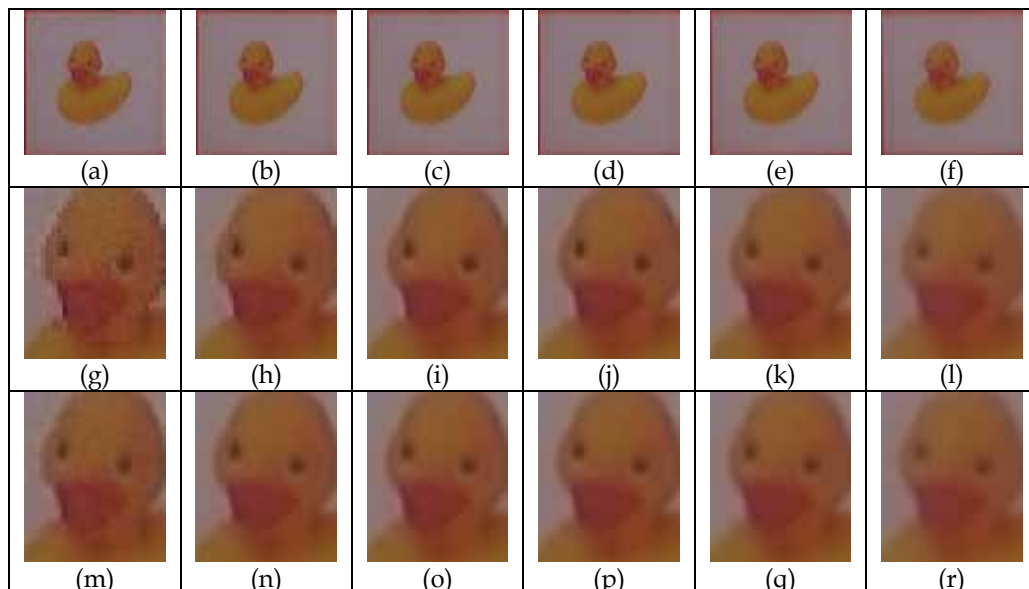

Fig. 13. Result of the proposed combined tracking and super-resolution approach after combining (a) 1, (b) 5, (c) 10, (d) 20, (e) 50, (f) 100 frames. Figure (g)-(l) show cropped parts of every Figure (a)-(f) respectively and Figure (m)-(r) show the result after each input image was doubled in size using bilinear interpolation.
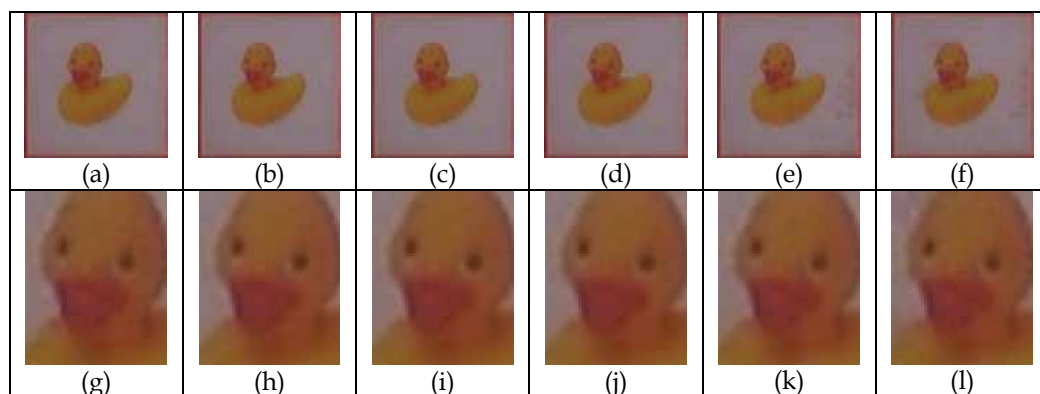
Fig. 14. Result of the super-resolution optical flow algorithm after taking the mean across (a) 1, (b) 5, (c) 10, (d) 20, (e) 50 and (f) 100 frames. A cropped part of each image is shown in (g)-(l) respectively.

Super-resolution optical flow increase the resolution of each frame by interpolation, combining several frames afterwards enhances the quality of the super-resolved image but does not further increase the resolution. The increase in resolution is usually fixed at 200%. This is contrary to our approach as we avoid interpolation to allow for less blurred images. But optical flow based methods require less frames as shown in Figure 14. The quality of the

optical flow super-resolved image increases most significantly within the first 5 to 10 frames as shown in Figure 14(b) and Figure 14(c). This corresponds to the number of frames most optical flow based super-resolution methods use, like in (Baker & Kanade, 1999). The addition of more frames results in more blurred and noisy images as estimating an accurate dense flow field across a large number of frames is difficult and erroneous especially in low-resolution images. This is clearly visible in Figure 14(e) and Figure 14(f). Estimating a dense optical flow field across 50 to 100 frames is erroneous and the calculation of the mean across such a large number of frames results in artefacts and distortion.
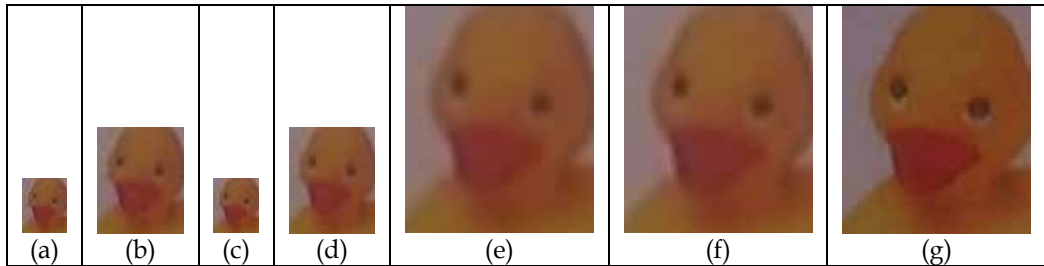

Fig. 15. (a) Cropped original frame with a cube of 100x100 pixels, (b) result after applying bilinear interpolation, (c) result of the optical flow after 10 frames without interpolation, (d) result of the optical flow after 10 frames with interpolation, (e) result of our approach after 20 frames with interpolation (f) result of our approach after 20 frames without interpolation, and (g) cropped cube of size 400x400 pixels.

Figure 15 summarises the results of both methods. The original video is recorded at a resolution of 320x240 and the cube is of size 100x100. A single cropped frame is shown in Figure 15(a). The simplest way of increasing the resolution is by interpolation, the result of bilinear interpolation is therefore shown in Figure 15(b). First we applied the super-resolution optical flow algorithm without an initial resolution increase by interpolation. The result after taking the mean across 10 frames is shown in Figure 15(c). The quality of the image is improved, i.e. the image noise is reduced, but the resolution remained unchanged. Interpolation is needed to increase the resolution, thus Figure 15(d) shows the result of the optical flow algorithm using bilinear interpolation to double the resolution of the input frames.

We also applied our approach to the video sequence that has been doubled in size using interpolation. The resulting super-resolved image, as shown in Figure 15(e), is more blur red and shows less detail, as a result of the interpolation, compared to the super-resolved image in Figure 15(f) that is calculated from the original input sequence. Furthermore the subdivision of the 3D model into a fine mesh allows for a greater increase in resolution compared to optical flow based methods. Lastly, Figure 15(g) shows a cropped image of the cube of size 400x400 pixels, which equals a resolution increase of 400% compared to Figure 15(a).

Furthermore, we tested our approach on non-planar and non-rigid objects, in this case that of surveillance video of people entering a bus. The video is recorded with a resolution of 640x480 at 23 frames per second due to dropped frames. Each frame is sub-sampled to half the resolution resulting in 320x240 pixels. The face within one frame is about 32x25 pixels, a single cropped frame of two different persons is shown in Figure 16(a) and Figure 16(f). We applied the combined geometric and appearance based approach to track these faces across about 40 frames.
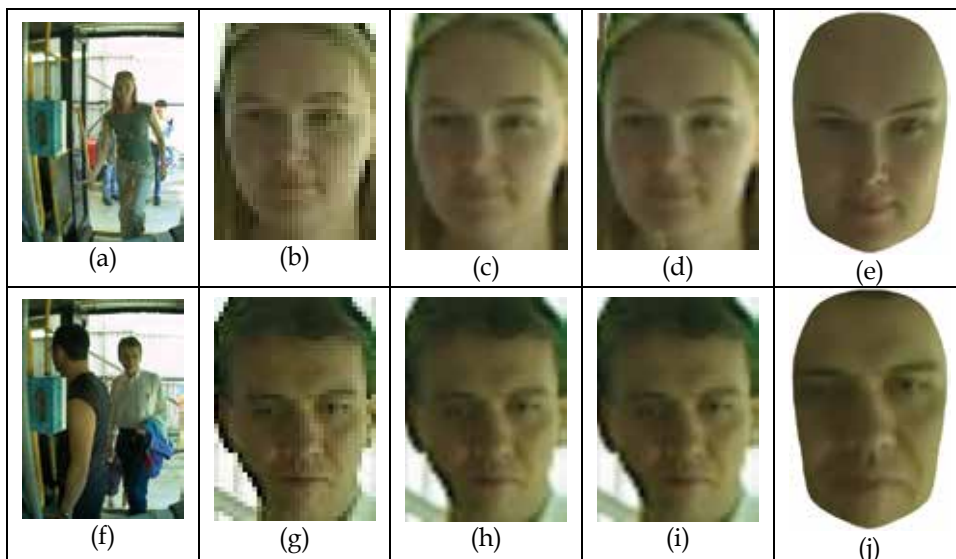
Fig. 16. Original image (a) and (f), cropped face (b) and (g), bilinear interpolation (c) and (h), optical flow result combining 20 frames (d) and (i) and our approach combining 20 frames (e) and (j).

We omitted the use of the extended expression tracking approach because we assume that people have a fairly neutral expression when entering the bus. Furthermore the low resolution of the face does not justify the runtime overhead of the proposed expression tracking. Also the difference in tracking error when tracking with and without expressions decreases with the resolution as shown in Section 4.3. If expressions occur during tracking nonetheless the tracking error will increase. But as the super-resolved image is created by using only frames below the threshold ε (Equation 7), frames with expressions will not be used and therefore not affect the result.

Optical flow is feasible for tracking planar objects like the front side of a cube in the last experiment but tracking non-planar and non-rigid objects like faces poses more challenges, especially when trying to estimate a dense flow field across a large number of frames. The result of the optical flow combining 20 frames is shown in Figure 16(d) and Figure 16(i). The chin area in Figure 16(d) shows slight distortion due to erroneous flow vectors. Figure 16(b) and Figure 16(g) show the initial frames after they have been doubled in size using bilinear interpolation.

A comparative result of the combined tracking and super-resolution method after 20 frames is shown in Figure 16(e) and Figure 16(j). The super-resolved faces of our approach are less blurred and shows more detail compared to the result of the optical flow. Again the initial interpolation introduces artificial random noise, whereas our approach does not need an initial interpolation; but the achieved resolution increase after 20 frames is equal or slightly higher.

Another advantage of our approach is the further use of the created super-resolved 3D model. In case of faces these models can be used to generate various face images under different pose and lighting in order to improve the training of classifiers or the super-resolved 3D model itself can be used for 3D face recognition.

6. Conclusion

We proposed a combined tracking and super-resolution algorithm that increases the resolution online during the tracking process. An object-specific 3D mask mesh is used to track non-planar and non-rigid objects. This mask mesh is then subdivided such that every quad is smaller then a pixel when projected into the image. This makes super-resolution possible and in addition improves tracking performances. Our approach varies from traditional super-resolution as the resolution is increased on mask level and only for the object of interest rather than on image level and for the entire scene.

We demonstrated our combined geometric and appearance based tracking approach on sequences of different size faces and showed that our approach is able to track faces down to 28x20 pixels in size. The combination of these two tracking algorithms achieves better results than each method alone.

The tracking algorithm is further extended to allow for the non-rigidity of objects. We applied this to faces and expressions. Experiments showed that the proposed method for expression tracking reduces the mean tracking error and thus allows for a better alignment of consecutive frames, which is needed to create super-resolution images.

The proposed 3D model-aided super-resolution allows for a high increase in resolution; the finer the 3D mesh the higher the possible increase in resolution. Therefore we experimentally estimated the number of frames needed to achieve a certain resolution increase. In practice about 20 to 30 frames are needed to double the resolution. Increasing the resolution further is limited by the number of frames used as well as the tracking error. Large tracking errors and the averaging process across a large number of frames introduces noise that decreases the quality of the super-resolved image.

We demonstrated our method on low resolution video of faces that are acquired both in the lab and in a real surveillance situation. We show that our method outperforms the optical flow based method, and performs consistently better for longer tracking durations in video that contain non-planar and non-rigid low-resolution objects. The combined tracking and super-resolution algorithm increases the resolution on mask level and makes interpolation, the first step of the optical flow algorithm, redundant. The resulting super-resolved 3D model is less blurred by achieving the same or a higher resolution increase. This in turn makes deblurring, the last step of the optical flow algorithm, unnecessary. Furthermore the super-resolved 3D model is created online during tracking and improves with every frame, whereas super-resolution optical flow incorporates consecutive and previous frames which prohibits its usage as an online stream processing algorithm.

7. References

Agrawal, A. & Raskar, R.. (2007). Resolving Objects at Higher Resolution from a Single Motion-blurred Image. *IEEE Conference on Computer Vision and Pattern Recognition*, pp.1-8, June 2007.

Ahlberg, J. (2001). CANDIDE-3 - An updated parameterized face. *Technical Report LiTH-ISY-R-2326*, Department of Electrical Engineering, Linkping University, Sweden.

Baker, S. & Kanade, T. (1999). Super-resolution optical flow. *Technical Report CMU-RI-TR-99-36*, Robotics Institute, Carnegie Mellon University, October, 1999.

Baker, S. & Kanade, T. (2002). Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (9), 1167 - 1183.

Barreto, D.; Alvarez, L. D. & Abad, J. (2005). Motion Estimation Techniques in Super-Resolution Image Reconstruction. A Performance Evaluation. *Virtual Observatory: Plate Content Digitization, Archive Mining and Image Sequence Processing, iAstro workshop*, Sofia, Bulgaria, p. 254-268

Bascle, B.; Blake, A. & Zisserman, A. (1996). Motion deblurring and super-resolution from an image sequence. *Proceedings of the 4th European Conference on Computer Vision*, Volume II. Springer-Verlag, London, UK, pp. 573-582.

Ben-Ezra, M.; Zomet, A. & Nayar, S. K. (2005). Video super-resolution using controlled subpixel detector shifts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (6), 977-987.

Borman, S. & Stevenson, R. L. (1998). Super-resolution from image sequences - A review. *Proceedings of the 1998 Midwest Symposium on Systems and Circuits*. IEEE Computer Society, Washington, DC, USA, pp. 374-378, August 1998.

Cascia, M. L.; Sclaroff, S. & Athitsos, V. (2000). Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Computer Society*, 2000, 22, 322-336

Chiang, M. & Boult, T. (2000). Efficient super-resolution via image warping. *Image and Vision Computing*, Elsevier, Vol. 18 (10), July 2000, pp. 761-771.

Dellaert, F.; Thorpe, C. & Thrun, S. (1998). Super-resolved texture tracking of planar surface patches. *IEEE/RSJ International Conference on Intelligent Robotic Systems*, October 1998, pp. 197-203.

Farsiu, S.; Robinson, D.; Elad, M. & Milanfar, P. (2004). Advances and challenges in super-resolution. *International Journal of Imaging Systems and Technology* 14 (2), Wiley, August 2004, 47-57.

Faugera, O. (1993). Three-Dimensional Computer Vision - A Geometric View-point. *MIT Press*, Cambridge, MA, USA.

Gao, C. & Ahuja, N. (2006). A refractive camera for acquiring stereo and super-resolution images. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2), pp. 2316-2323.

Gautama, T. & van Hulle, M. (2002). A phase-based approach to the estimation of the optical flow field using spatial filtering. *IEEE Transactions on Neural Networks* 13 (5), September 2002, pp. 1127-1136.

Goldenstein, S. K.; Vogler, C. & Metaxas, D. (2003). Statistical cue integration in DAG deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (7)*, pp. 801-813, July 2003.

Huang, T. S. & Tsai, R. Y. (1984). Multi-frame image restoration and registration. *Advances in Computer Vision and Image Processing: Image Reconstruction from Incomplete Observations*, Thomas S. Huang (Ed.), London, 1984, Vol. 1, pp. 317-339, JAI Press.

Kuhl, A.; Tan, T. & Venkatesh, S. (2008). Model-based combined tracking and resolution enhancement. *Proceedings of the 2008 IEEE International Conference on Pattern Recognition*.

Lin, Z. & Shum, H.-Y. (2004). Fundamental limits of reconstruction-based super-resolution algorithms under local translation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (1), pp. 83-97, Jan 2004

Park, S. C.; Park, M. K. & Kang, M. G. (2003). Super-resolution image reconstruction: a technical overview. *IEEE Signal Processing Magazine* 20 (3), May 2003, pp. 21-36.

Roussel, R. & Gagalowicz, A. (2005). A hierarchical face behavior model for a 3D face tracking without markers. *Computer Analysis of Images and Patterns*. Vol. 3691. Springer, pp. 854-861.

Smelyanskiy, V.; Cheeseman, P.; Maluf, D. & Morris, R. (2000). Bayesian super-resolved surface reconstruction from images. *Proceedings. IEEE Conference on Computer Vision and Pattern Recognition*, vol.1, pp.375-382

Tanaka, M. & Okutomi, M. (2005). Theoretical analysis on reconstruction-based super-resolution for an arbitrary PSF. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2. IEEE Computer Society, Washington, DC, USA, pp. 947-954, June 2005.

Wang, Y.; Gupta, M.; Zhang, S.; Wang, S.; Gu, X.; Samaras, D. & Huang, P. (2005). High resolution tracking of non-rigid 3D motion of densely sampled data using harmonic maps. *Tenth IEEE International Conference on Computer Vision*, pp. 388-395, Oct. 2005

Wen, Z. & Huang, T. (2005). Enhanced 3D geometric-model-based face tracking in low resolution with appearance model. *IEEE International Conference on Image Processing*, vol.2, pp. II-350-3, September 2005

Yu, J. & Bhanu, B. (2006). Super-resolution restoration of facial images in video. *Proceedings of the 18th International Conference on Pattern Recognition*. IEEE Computer Society, Washington, DC, USA, pp. 342-345.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), pp. 1330-1334, Nov 2000

Zhao, W. & Sawhney, H. S. (2002). Is super-resolution with optical flow feasible?, *Proceedings of the 7th European Conference on Computer Vision-Part I*. Springer-Verlag, London, UK, pp. 599-613.

Zhigljavsky, A. A. (1991). Theory of global random search. *Dordrecht Netherlands : Kluwer Academic Publishers*.

Zorin, D. & Schröder, P. (2000). Subdivision for modeling and animation. *ACM Siggraph Course Notes*

Face Recognition Based on Human Visual Perception Theories and Unsupervised ANN

Mario I. Chacon M. and Pablo Rivas P.
Chihuahua Institute of Technology
Mexico

1. Introduction

The face recognition problem has been faced for more than 30 years. Although a lot of research has been done, much more research is and will be required in order to end up with a robust face recognition system with a potential close to human performance. Currently face recognition systems, FRS, report high performance levels, however achievement of 100% of correct recognition is still a challenge. Even more, if the FRS must work on non-cooperative environment its performance may decrease dramatically. Non-cooperative environments are characterized by changes on; pose, illumination, facial expression. Therefore FRS for non-cooperative environment represents an attractive challenge to researchers working on the face recognition area.

Most of the work presented in the literature dealing with the face recognition problem follows an engineering approach that in some cases do not incorporate information from a psychological or neuroscience perspective. It is our interest in this material, to show how information from the psychological and neuroscience areas may contribute in the solution of the face recognition problem. The material covered in this chapter is aimed to show how joint knowledge from human face recognition and unsupervised systems may provide a robust alternative compared with other approaches.

The psychological and neuroscience perspectives shows evidence that humans are deeply sensible to the face characteristic configuration, but the processing of this configuration is restricted to faces in a face-up position (Thompson, 1980), (Gauthier, 2002). This phenomenon suggests that the face perception process is a holistic configurable system.

Although some work has been done in these areas, it is still uncertain, how the face feature extraction processes is achieved by a human being. An interesting case is about newborn face feature extraction. Studies on newborns demonstrate that babies perceive a completely diffuse world, and their face perception and recognition is based on curves and lines from the face (Bower, 2001), (Johnson, 2001), (Nelson, 2001), (Quinn et al., 2001) and (Slater A. & Quinn, 2001).

Nowadays, there exists some research work on face recognition that has intended to incorporate psychological and neuroscience perspectives (Blanz & Vetter, 2003), (Burton et

al., 1999). However, the solution to the face recognition problem is stated only on bases of matrix operations and general pattern recognition methodologies, without considering other areas as visual perception.

On the engineering area, patter recognition systems approaches offer a large variety of methods. In recently years, unsupervised systems have provided a new paradigm for the pattern recognition problem (Chacon & Ramirez, 2006a). These systems allow data mining or data discovering information that traditional pattern recognition systems do not incorporate. This feature makes it possible to find information in the feature vectors that may not be considered in traditional pattern recognition approaches.

Based on these points, we present in this chapter a new face recognition approach taking into account recently face perception theories and an unsupervised classifier in order to improve the performance of the FRS in non-cooperative environments.

2. Literature analysis

This section presents a survey of 30 representative papers published in recently years. The purpose of this analysis is to provide the reader with a flavor of the variety of paradigms used in the face recognition problem, and to propose a method to compute an index performance of such methods. Table 1 shows the 30 published works analyzed. The numbers on the column No. are used later as references in figures and tables.

The first analysis shown in Table 2 is the robustness of the method with respect to variations on face; **P**ose, **I**llumination, **E**xpression and / or **R**otation. We can observe from Table 2 that only one method assumes tolerance to PIE/R, five of the methods are tolerant to PIE, eight only consider robustness to two variations. Eight methods are designed to be invariant to only one variation, and eight methods are not tolerant to any variation. The most considerable change in the works is **E**, followed by **P**, **I**, and the less is **R**. The performances reported vary from good, No. 1, to very poor No.5.

Figure 1 illustrates the feature extraction methods used in these papers, and Figure 2 shows the type of classifier used. The feature extraction methods are 3D models, Fisher´s Linear Discriminant FLD, Discrete Cosine Transform DCT, Linear Discriminant Analysis LDA, Principal Component Analysis PCA, wavelet based, Bayesian and other methods. It was observed that feature extraction methods that represent data in sub-spaces are the most commonly used. Among the classifier methods the Euclidean distance is the most used, followed by other methods, and the artificial neural network method approach.

With respect to the data bases, ORL, YALE, AR and MIT, are among the most used data bases. The ORL data base presents variations on pose, illumination, and expression (Li & Jain, 2004), (Samaria & Harter, 1994), (Olivetti, 2006). YALE has face images with individuals in different conditions, with and without glasses, changes in illumination, and expression (Li & Jain, 2004), (Yale, 2002). The AR data base includes changes on facial expression, illumination, and occlusion (Li & Jain, 2004), (Martinez & Benavente, 1998). The MIT data base is composed of face images involving variations on pose, illumination and facial expression (Weyrauch et al., 2004). Some examples of these data bases are shown in Figure 3.

No.	Publication
1	Deformation analysis for 3d face matching (Lu & Jain, 2005)
2	Discriminative common vectors for face recognition (Cevikalp et al., 2005)
3	Face recognition using laplacianfaces (He et al., 2005)
4	High-Speed face recognition based on discrete cosine transform and rbf neural networks (Er et al., 2005)
5	Locally linear discriminant analysis for multimodal distributed classes for face recognition with a single model image (Kim & Kittler, 2005)
6	Wavelet-based pca for human face recognition (Yuen, 1998)
7	Real-time embedded face recognition for smart home (Zuo & de With, 2005)
8	Acquiring linear subspaces for face recognition under variable lighting (Lee & Kriegman, 2005)
9	Appearance-Based face recognition and light-fields (Gross et al., 2004)
10	Bayesian shape localization for face recognition using global and local textures (Yan et al., 2004)
11	A unified framework for subspace face recognition (Wang & Tang, 2004)
12	Probabilistic matching for face recognition (Moghaddam & Pentland, 1998)
13	Face recognition based on fitting a 3d morphable model (Blanz & Vetter, 2003)
14	Face recognition using artificial neural network group-based adaptive tolerance (gat) trees (Zhang & Fulcher, 1996)
15	Face recognition by applying wavelet subband representation and kernel associative memory (Zhang et al., 2004)
16	Face recognition using kernel direct discriminant analysis algorithms (Lu et al., 2003)
17	Face recognition using fuzzy integral and wavelet decomposition method (Kwak & Pedrycz, 2004)
18	Face recognition using line edge map (Gao & Leung, 2002)
19	Face recognition using the discrete cosine transform (Hafed & Levine, 2001)
20	Face recognition system using local autocorrelations and multiscale integration (Goudail et al., 1996)
21	Face recognition using the weighted fractal neighbor distance (Tan & Yan, 2005)
22	Gabor-Based kernel pca with fractional power polynomial models for face recognition (Liu, 2004)
23	Gabor wavelet associative memory for face recognition (Zhang et al., 2005)
24	N-feature neural network human face recognition (Haddadnia & Ahmadi, 2004)
25	GA-Fisher: a new lda-based face recognition algorithm with selection of principal components (Zheng et al., 2005)
26	Kernel machine-based one-parameter regularized fisher discriminant method for face recognition (Chen et al., 2005)
27	Generalized 2d principal component analysis (Kong et al., 2005)
28	Face detection and identification using a hierarchical feed-forward recognition architecture (Bax et al., 2005)
29	Nonlinearity and optimal component analysis (Mio et al., 2005)
30	Combined subspace method using global and local features for face recognition (Kim et al., 2005)

Table 1. List of analyzed references.

No.	1	2	3	4	5	6	7	8	9	10
P	✓		✓		✓				✓	✓
I	✓		✓	✓				✓		
E	✓		✓	✓						
R										
%Rec.	91.00	98.34	91.96	91.20	53.00	-	95.0	98.82	70.65	93.66
No.	11	12	13	14	15	16	17	18	19	20
P			✓					✓		
I			✓				✓	✓	✓	
E			✓				✓	✓		✓
R									✓	
%Rec.	96.0	89.5	82.70	-	91.20	-	99.24	73.30	84.58	95.00
No.	21	22	23	24	25	26	27	28	29	30
P	✓	✓			✓		✓	✓		
I	✓				✓	✓			✓	
E	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
R	✓									
%Rec.	67.40	95.30	99.30	99.50	89.63	92.03	95.48	79.00	93.00	97.00

Table 2. Robustness analysis with respect to Pose, Illumination, Expression, and Rotation.
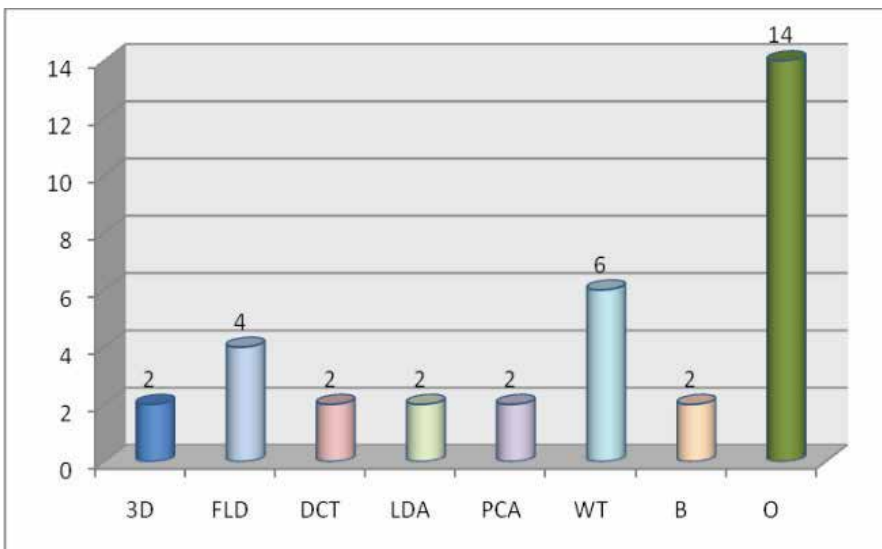

Fig. 1. Feature extraction methods. 3D models, Fisher´s Linear Discriminant, Discrete Cosine Transform, Linear Discriminant Analysis, Principal Component Analysis, Wavelet Transform, Bayesian methods, other methods.
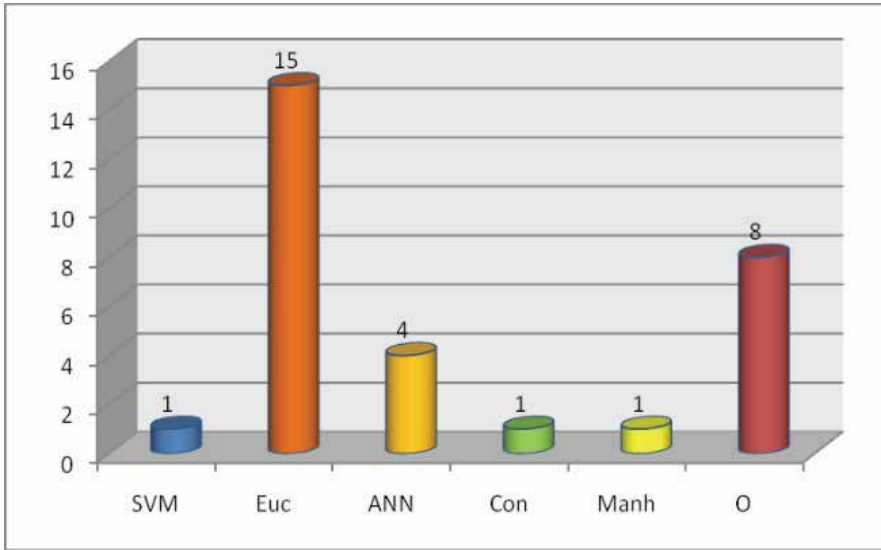
Fig. 2. Analysis by classifier scheme. Support Vector Machine, Euclidean distance, Artificial Neural Networks, Convolution , Manhattan, Other.

Since the experimentation to obtain the performance of the proposed method in the analyzed works is different, it is difficult to achieve a comparison among the methods. Therefore, in order to obtain a comparable performance index we proposed one. The proposed performance index assigns a weight of 10 to the number of individual that can be recognized by the system and a weight of 90 to the recognition performance. These weights were assumed arbitrarily and can be adjusted to a particular criterion. The performance index is defined as follows. Let p_{max} be the maximum recognition performance of FRS and $prec_k$ the recognition of the $k-th$ method in its different performances. Let also n_{max} be the maximum number of faces that the $k-th$ method can recognize. $nfaces_k$ is the number of individuals that the $k-th$ method can recognize for $k = \{1,2,3,...,N\}$. Then the performance index is given by

$$p_{max} = \max(prec_k), \qquad p_{max} < 100 \tag{1}$$

$$n_{max} = \max(nfaces_k) \tag{2}$$

$$r_k = \frac{prec_k \times p_{max}}{w_{prec}} + \frac{nfaces_k \times n_{max}}{w_{nfaces}} \tag{3}$$

where r_k is the proposed performance index of the $k-th$ method. w_{prec} is the facial recognition performance weight, and w_{nfaces} the weight for the number of individuals that can be recognized. Using this performance index, the best method is the $k-th$ method that maximizes

$$d_{max} = \max(r_k) \tag{4}$$
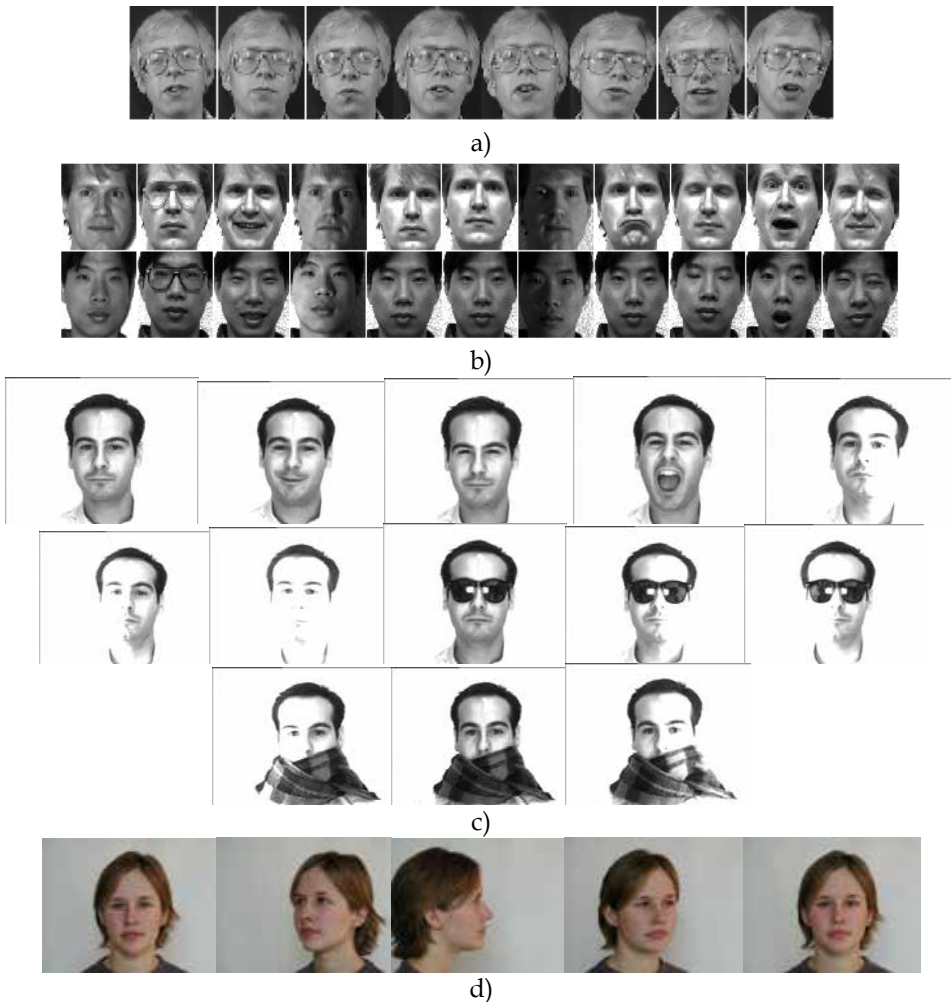
a)

b)

c)

d)

Fig. 3. Example of data base images. a) ORL, b) YALE, c) AR, and d) MIT.

Fig. 4. shows the summary of the performance of the best three methods for each combination of robustness. In Figure 4, the bars indicate the recognition performance for each method, and the lines indicate the number of faces that each method can recognize.

It can be noticed that the performance of the method increases as the number of faces decreases and vice versa. It can also be observed that the methods No. 22 (Liu, 2004), No. 17 (Kwak & Pedrycz, 2004) and No. 3 (He et al., 2005) appear like best methods in more than one robustness category.

A summary of the two best methods is shown in Table 3. It shows the classifier type, and the face feature extraction method used. The best method tolerant to PIE has a performance of 91.96%. From Table 3 it is observed that methods that appear more frequently among the best face feature extraction are based on wavelets. It is also noticed that most of the methods are baseed on simple classifiers like nearest-neighbor, which open the opportunity to investigate with other classifiers like support vector machine, and artificial neural networks in order to improve their performance.
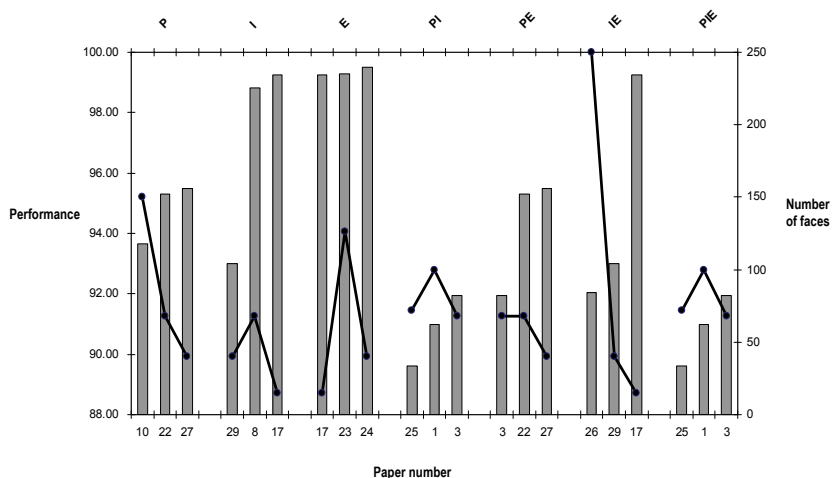
Fig. 4. General results of the evaluation. Bars indicate the percentage of performance. Black lines indicate the number of individual that the method can recognize.

Tolerant to	No.	Classifier	Characteristics
P	10	Euclidian distance	Gabor wavelet representation y Bayesian Shape Localization
P	22	Nearestneighbor using Euclidean distance	Gabor wavelet representation of face images and the kernel PCA method
I	8	9 illumination points(9PL)	9 illumination points(9PL)
I	17	Euclidian distance and fuzzy integral	Wavelet decomposition, Fisherface method
E	23	GWAM	Gabor wavelet associative memory GWAM
E	24	RBF Neural Network	Ellipse of a facial image
PI	1	SVM	Deformation analysis of a 3D figure
PI	3	Nearestneighbor using Euclidean distance	The Laplacianfaces are obtained by finding the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the face manifold
PE	22	Nearestneighbor using Euclidean distance	Gabor wavelet representation of face images and the kernel PCA method
PE	27	Nearest-neighborhood	K2DPCA
IE	17	Euclidian distance and fuzzy integral	Wavelet decomposition, Fisherface method
IE	26	RBF kernel function	K1PRFD algorithm
PIE	3	Nearestneighbor using Euclidean distance	The Laplacianfaces are obtained by finding the optimal linear approximations to the eigen functions of the Laplace Beltrami operator on the face manifold
PIE	25	Optimal projection matrix of GA-Fisher	GA-Fisher

Table 3. Summary of the two best methods.

At this point, we have presented an analysis of reported work on FRS considering, robustness, features used and type of classifiers. In the next section, we present important results on human face perception theory that can be considered in the design of new FRS.

3. Human facial perception theory

Notwithstanding the tremendous effort to solve the face recognition system, it is not possible yet, to have a FRS that deploys effectively in unconstrained environments, tolerant to illumination, rotation, pose, expression, noise, and viewing distance. The most efficient system without any doubt is the human system, therefore, it makes sense to try to discover the secret of this system.

There are some studies in the fields of psychology and neuroscience related to face recognition (Thompson, 1980), (Gauthier & Tanaka, 2002), (Haxby et al., 2001), (Kalocsai, 1998), (Bower, 2001), (Mareschal & Thomas, 2006). However, aside from the diversity of experiments and approaches, it is notorious that there is not a final conclusion about questions like; What features does the brain use to recognize a face?, Is there a specialized region of the brain for face recognition?. From the psychological and neuroscience point of view there exists evidence that humans are very sensible to face configuration, that is, relationship among the face constituents, nose, mouth, eyes, etc. But, the process is related only to upright faces (Thompson, 1980), (Gauthier & Tanaka, 2002). This phenomenon is known as the "Margaret Thatcher Illusion", (Sinha, 2006) and (Thompson, 1980). Figure 6 illustrates this phenomenon. Apparently the brain does not perceive differences between a modified upright face and a normal face, Figures 6a and 6b. This is because for the brain it is easier to process an inverted face. However, when the face that apparently does not have differences in comparison with the normal face is rotated, we can perceive the differences between these two faces, Figure 6c and 6d. This phenomenon indicates that it is possible that face perception is a holistic – configural system, or configuration based.
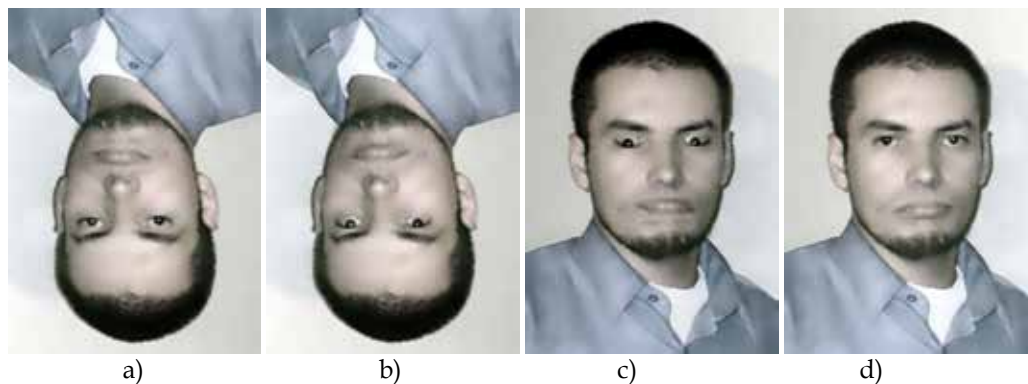

| a) | b) | c) | d) |

Fig. 6. Examples of the "Margaret Thatcher Illusion". a) Upright face with eyes inverted, b) normal upright face, c) and d) corresponding right faces of a) and b)

The work in (Gauthier & Tanaka, 2002) distinguish between two concepts; holistic – inclusive and holistic – configural. Holistic – inclusive is defined as the use of a part of an object even though it is said to be ignored. On the other hand, holistic – configural is defined as the use of relations among the parts of the object. Therefore, under a holistic – configural approach it is important to consider those relationships to achieve a good recognition performance.

In other studies, the area of neuroscience has suggested that the face recognition process is achieved in the brain by a specialized region (Thompson, 1980), (Haxby et al., 2001). These studies are based on analysis of PET and MRI images during object and face recognition experiments by humans. The *laterar fusiform gyrus* or *Fusiform Face Area*, region shows activity during the face recognition task, but this activity is not detected during object recognition (Kanwisher, 1997), (Tong et al., 2000).

Other evidence suggests a specific region in the brain related to the face identification that is related to the disease called *prosopagnosia*. People with this problem can recognize objects and face expressions but not faces.

Notwithstanding all this research it is not possible yet, to define a coherent theory for the face recognition process in humans. Nevertheless, some works, (Haxby et al., 2001), (Kanwisher, 2006), (Kalocsai, 1998), can state guidelines that can improve the performance of computer FRS. For example the work in (Bower, 2001) indicates that newborns perceive a fuzzy world and they resort to line and curve face shapes for face recognition. Figure 7 illustrates the perception and visual sharpness of newborns during their development (Brawn, 2006). The sequence is Figure 7a newborn, 7b four weeks, 7c eight weeks, 7d three months, and 7e six months. The work in (Peterson & Rhodes, 2003) demonstrates that lines are better features in the holistic configuration to provide discrimination among other type of geometric objects. This could lead to the fact of why newborns have the ability to recognize people using diffuse lines features.
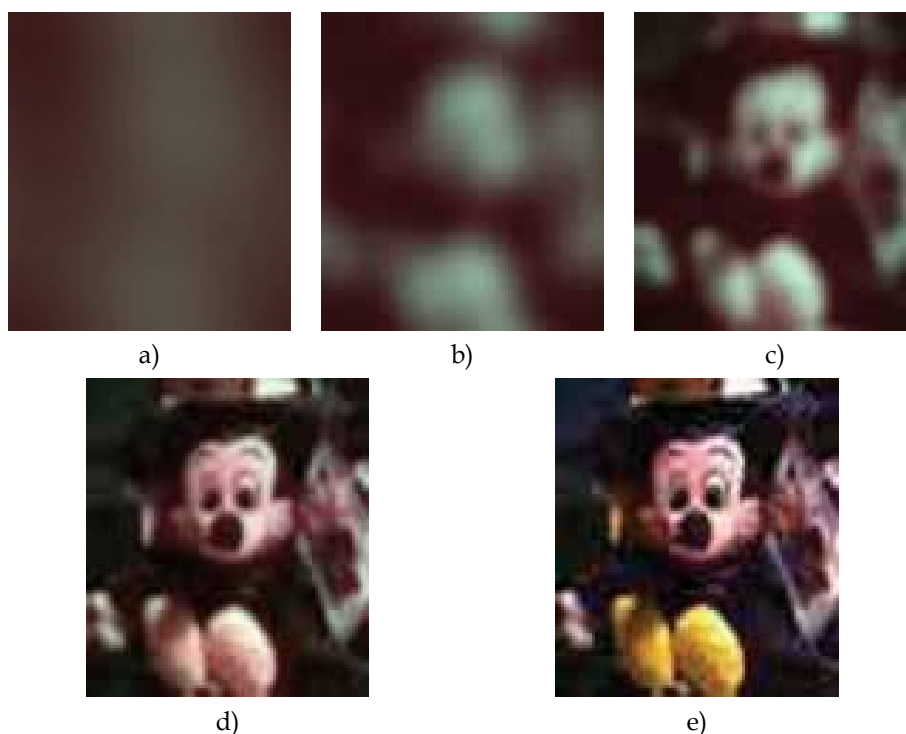

Fig. 7. Newborn visual perception variation. a) Newborn, b) 4 weeks, c) 8 weeks, d) 3 months, e) 6 months. From professor Janice Brown's class presentations (Brawn, 2006).

As a conclusion of this section we recommend the work reported by Sinha (Sinha et al., 2006), where the reader could find nineteen important points a computer vision research interested on face recognition should consider in FRS design. Among those points we can mention; human recognize familiar faces in very low resolution images, high – frequency information by itself is insufficient for good face recognition performance, facial features are processed holistically, pigmentation cues are at least as important as shape cues.

These points are results of human visual perception experiments that are related to the main idea presented in this section, and more details can be found in that reference.

4. Face feature lines, hough-KLT human facial perception theory

Considering the theories presented in section 3 we decided to implement a face recognition scheme based on facial features containing information of the most prominent lines in low resolution faces. Besides the perceptual justification of this theory, an engineering justification to use features lines are the works reported in (Zhao et al., 2003) where lines are combined with PCA and (Konen , 1996), where the ZN-Face algorithm is able to compare drawing faces against gray scale faces using lines. Besides, other works show the advantages, like representation simplicity (Singh et al., 2003), low computational cost (Singh et al., 2002), invariance tolerance of face recognition algorithms based on lines, (Aeberhard & del Vel, 1998) and (Aeberhard & de Vel, 1999).

4.1 Face feature lines

The proposed method is based on the features that we will denominate, face feature lines, FFL. FFL are prominent lines in low resolution face images, and can be extracted using the Hough transform. The Hough transform is a transformation that allows to detecting geometric patterns in images, like lines, circles, and ellipses. The Hough transform, HT, works on a parametric space to define a line by

$$\rho = x\cos\theta + y\sin\theta \tag{5}$$

where x and y represent the coordinate of a pixel, ρ is the distance of the line to the origin, and θ is the angle of the line with respect the horizontal axis. FFL can be extracted from the HT by obtaining the maximum values in the transformation. We consider that four face feature lines are enough to represent a face, based on the experiments related to the newborns vision system. These four FFL have shown significant improvement in the performance of fuzzy face recognition systems (Chacon et al., 2006b). The information of these four FFL will be included as components of the feature vector which is defined with detail on further subsections. Figure 8 illustrates how the FFL are obtained.

The feature vector including the FFL is generated as follows:

Step 1. Find the four maximum peak of the HT.

Step 2. Obtain the four characteristic lines coordinates.

Step 3. Encode the coordinates information by taking the value of the first coordinate of the

  i-th line, x_{1i} and add it to $\dfrac{y_{1i}}{1000}$, and include the result to l_{i_1} .

Step 4. Take the value of the second coordinate of the i-th line, x_{2i} and add it to $\dfrac{y_{2i}}{1000}$, and

  include the result to l_{i_2} .

The FFL feature vector can be defined by

$$\mathbf{z}_i = [l_{i_1} \quad l_{i_2}]$$

$$\mathbf{z}_i = \begin{bmatrix} x_{11} + \dfrac{y_{11}}{1000}, x_{21} + \dfrac{y_{21}}{1000} \cdots \\[2em] x_{1i} + \dfrac{y_{1i}}{1000}, x_{2i} + \dfrac{y_{2i}}{1000} \end{bmatrix} \qquad (6)$$

a)   b)   c)   d)
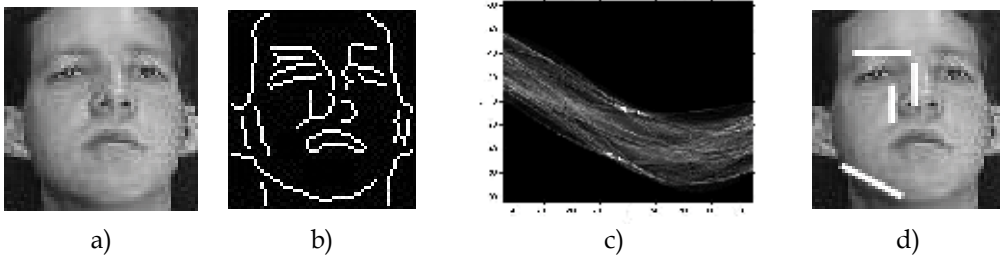
Fig. 8. FFL extraction: a) Original image, b) Face edges, c) Accumulator of the HT. d) Original image plus its four FFL.

The $\mathbf{z}_i$ vector must be concatenated with the original image $I(x, y)$, in a canonical form (vector column) $\mathbf{i}_{xy}$, to construct the final feature vector

$$\mathbf{X}_{i+xy} = [\mathbf{z}_i \quad \mathbf{i}_{xy}] \qquad (7)$$

The vector $\mathbf{z}_i$ is linked to the information of the original image in order to contribute and complement the face information representation before the transformation via KLT.

4.2 Principal component analysis and Karhunen-Loeve transformation

The feature vector is now processed by Principal Component Analysis, PCA, in order to reduce the features dimensionality. This reduction is achieved by the PCA that transforms the representation space X into a new space Y, in which the data are uncorrelated. The covariance matrix in this space is diagonal. The PCA method leads to find the new set of orthogonal axis to maximize the variance of the data. The PCA transformation is accomplished by

Step 1. The covariance matrix $\text{Cov}_{\mathbf{X}}$ is calculated over the input vectors set $\mathbf{X}_i$ that corresponds to i facial images represented as vectors $\mathbf{x}$. The covariance is defined as

$$\text{Cov}_{\mathbf{X}} = \frac{1}{n-1} \sum_{i=1}^{n} \left(\mathbf{x}_i - \overline{\mathbf{x}} \right) \left(\mathbf{x}_i - \overline{\mathbf{x}} \right)^T \qquad (8)$$

where $\overline{\mathbf{x}}$ denotes the mean of each variable of the vector $\mathbf{X}$, and n is the amount of input vectors.

Step 2. The n eigenvalues of $\text{Cov}_{\mathbf{X}}$ are extracted and defined as $\lambda_1, \lambda_2, \ldots \lambda_n$, where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$.

Step 3. The n eigenvectors are $\Phi_1, \Phi_2, ... \Phi_n$ and are associated to $\lambda_1, \lambda_2, ... \lambda_n$.

Step 4. A transformation matrix, $\mathbf{W}_{PCA}$, is created $\mathbf{W}_{PCA} = [\Phi_1, \Phi_2, ... \Phi_n]$.

Step 5. The new vectors $\mathbf{Y}$ are calculated using the following equation

$$\mathbf{Y} = \mathbf{W}_{PCA}^T \mathbf{X} \tag{9}$$

where T denotes the transpose of $\mathbf{W}_{PCA}$, and $\mathbf{X}$ denotes the matrix containing all the input vectors.

The Karhunen-Loeve transformation, KLT, is similar to the PCA (Li & Jain, 2004), however in the KLT the input vectors $\mathbf{X}_i$ are normalized to the interval [0,1] before applying the PCA steps.

4.3 SOM- *k*means face recognition system

The face recognition system is based on a combination of the *k*-means and SOM methods. Its description is presented next.

The system is designed to recognize 10 persons. The design samples considered are the first 8 samples of each individual. This approach generates a training matrix size of 3408x80. The face databases used were the ORL and the YALE.

The classifier design is performed in two steps. First the SOM is trained with the trained samples. The parameters of the SOM using the Kohonen algorithm are; input dimension 3488, map grid size 15X13, lattice type hexagonal, shape sheet, neighborhood Gaussian. Once the SOM has detected the possible classes, Figure 9 a, they are reinforced through the *k*-means algorithm. The *k*-means is applied trying to find 10 clusters, one for each class. Graphical representations of the clusters generated are shown in Figure 9b. Each hexagon in Figure 9b includes the label corresponding to the subject that has been assigned to a specific neuron on the map. The color scale represents the clusters found when the SOM is trained with 8 samples per subject. The U-matrix is a class distribution for graphic representation.
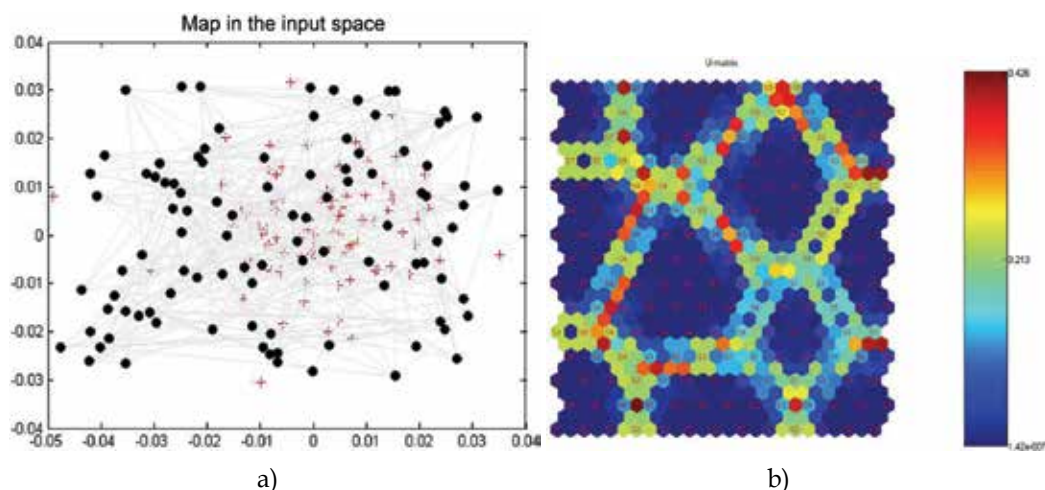

a)   b)

Fig. 9. The final map after Kohonen's training algorithm over ORL is shown in a). b) U-matrix of the SOM map when the k-means is applied over ORL.

The performance achieved for the ORL was 100% and 90% for design and testing respectively. For the YALE database the performance achieved was 100% and 70% for design and testing respectively. The use of the *k*-means-clustering algorithm, that reinforces the grouping, may justify this higher recognition rate. As expected, the performance has lower rates on the YALE database because of the variations in lighting conditions of the YALE database. However the performance is comparable with current face recognition systems based on PCA which achieves 77%.

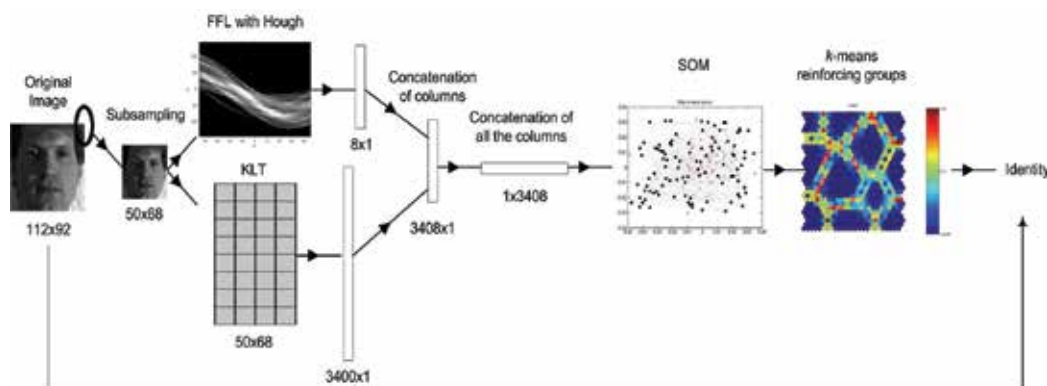The general scheme for the SOM-Hough-KLT proposed method is shown in Figure 10.

Fig. 10. General scheme for the SOM-Hough-KLT face recognition method.

The propose FRS based on the SOM was compared against a feedforward back propagation scheme for face recognition called FFBP-Hough-KLT. The highest recognition rate on testing reaches 60% on the YALE database, and 92% on the ORL. This result indicates an advantage of unsupervised over supervised systems.

The results obtained in this work are comparable to PCA, LDA, FLDA methods. For the YALE database the highest performance reported in the literature analyzed is 80% and for ORL database is 97%. Another important result is that the SOM network improved with the *k*-means performed better than the FFBP network. This leads us to think that hybrid systems will offer new alternatives to design robust face recognition systems.

5. Comparison of the proposed method with other classifiers

This section presents a set of experiment results where we compare the performance of several FRS by evaluating the classifiers and the feature vectors. The comparison is considering the average performance of each classifier with respect to the data bases AR, YALE, MIT, and ORL. Table 4 shows the average performance by classifier by data base for the different data bases. ED stands for Euclidean distance, NFL is nearest feature line, FFBP1 and FFBP2 are feedforward neural networks, GG is the fuzzy clustering algorithm Gath-Geva modified by Abonyi – Szeifert (Abonyi et al., 2002), N-D corresponds to a fuzzy neural system using RBF. It can be noted that the SOM has the best performance in two of the four data bases, and it is not far from the best performance of the best FRS in the other two cases. Besides the SOM has the higher performance reached of all FRS, with 92.86%. The highest performance is in the ORL data base as expected because it has less variation with respect to PIE. Contrary to the AR and YALE data bases that have more PIE variations.

With respect to the type of classifier system, the systems based on Artificial Neural Networks and Fuzzy Logic resulted to be the most consistent on their performance over a set of 14 recognition experiments taking different sets of faces as shown in Figure 11.The experiment number 8 had the lowest performance because some of the worst face images were included on that testing set. We can also observe that the SOM approach was the best FRS among the fuzzy and ANN classifiers.

Data base	FRS						
	ED	NFL	FFBP 1	FFBP 2	SOM	GG	N-D
AR	85.49	87.00	83.25	85.15	88.59	78.82	75.61
YALE	89.79	90.73	84.79	86.76	89.76	80.17	75.61
MIT	90.52	91.29	86.48	88.11	91.11	81.66	79.24
ORL	84.49	85.38	88.78	90.05	92.86	83.62	83.27

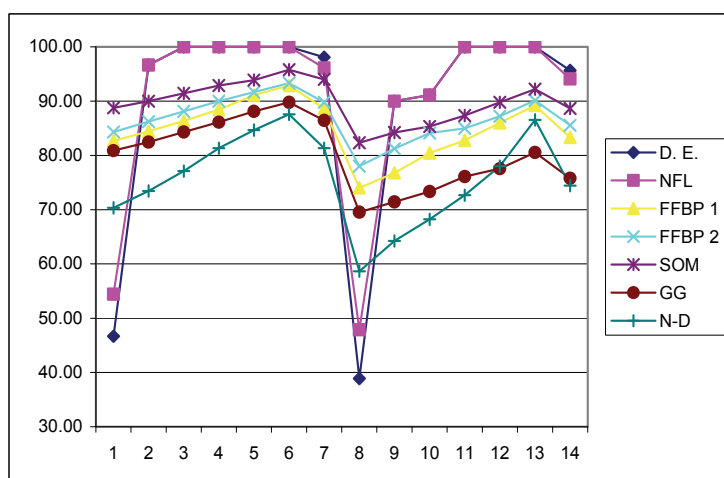Table 4. Classifier performance by data base.

Fig. 11. Performance by type of classifier in the 14 experiments.

Another important finding in these experiments is that the Hough-KLT feature yielded the best performance compared with the other features as shown in Table 5 over the different data bases. This result may reinforce the use of the face feature line feature. In this table DDWT stands for Wavelet features, and dB is the wavelet level used.

6. Results, conclusion and future work

The chapter presented an analysis of 30 works on FRS. The works were analyzed for tolerance to image variations, feature extraction methods and classifier system used. Results indicate that the most considered variation in the works is related to face expression, one of the most used method for feature extraction turned to be wavelet based methods, and the classifiers with more use are based on Euclidean distance. In order to compare the performance of the different works a new performance index was proposed. The best performance for a FRS assuming PIE tolerance reached 91.96%.

Feature	Data base			
	AR	YALE	MIT	ORL
DDWT-KLT db1	82.12	85.18	87.42	86.56
DDWT-KLT db2	82.82	85.13	87.36	86.98
DDWT-KLT db4	83.42	86.24	87.74	86.66
KLT	82.09	83.95	85.52	86.01
Gabor-KLT	81.81	84.18	85.10	86.02
Hough-KLT	86.69	88.57	89.32	89.43
Eigenfaces	80.51	82.52	84.59	85.12

Table 5. Performance by features.

The chapter also covered an alternative point of view for FRS based on human facial perception theories. This part of the chapter, presented some neurophysiologic theories that may be useful to design more robust FRS. In fact, one of these theories, newborn visual perception, is proposed to be incorporated in a novel face recognition approach that is described in the chapter. The newborn visual perception idea led us to consider low resolution features extracted with the Hough transform, FFL, to design a FRS.

The FRS designed is based on a SOM- $k$means classifier. The performance achieved during testing were, 90% for the ORL, 70% for the YALE database. As expected, the performance has lower rates on the YALE database because of the variations in lighting conditions of the database. However the performance is comparable with current face recognition systems based on PCA which achieves 77%.

In a final experiment, the proposed method was compared against 6 other methods using the AR, YALE, MIT, and ORL data based. The proposed method turned to achieve the best performance in two of the test, with 88.59% and 92.86%, and it was the second best in the other two, 89.76% and 91.11%. Another important result in this experiment is that with respect to the type of classifier system, the systems based on Artificial Neural Networks and Fuzzy Logic resulted to be the most consistent on their performance over a set of 14 recognition experiments. Besides, the SOM approach was the best FRS among the fuzzy and ANN classifiers.

Still another important finding in these experiments is that the Hough-KLT feature, that incorporates the FFL, yielded the best performance compared with the other features.

Based on the previous results, we can conclude that incorporation of neurophysiologic theories into the design of FRS is a good alternative towards the design of more robust systems. We also may conclude that FRS based on ANN, specially with unsupervised systems, represent a good alternative according to the results of the experiments reported in this chapter.

As future work, we propose to achieve a more complete research towards the integration of the results presented in (Sinha et al., 2006) into FRS design in order to evaluate the real impact of these theories in real world applications.

7. Acknowledgment

8. References

Abonyi J.; Babuska R. & Szeifert F. (2002). Modified Gath-Geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models, *IEEE Transaction on Systems, Man and Cybernetics Part B*, Vol. 32, Issue: 5, October 2002, pp. 612 – 621,ISSN 1083-4419.

Aeberhard S. & del Vel O. (1998). Face recognition using multiple image view line Segments, *1998 Proceedings of the Fourteenth International Conference on Pattern Recognition*, Vol. 2, Augusto 1998, pp. 1198 – 1200.

Aeberhard S. & de Vel O. (1999). Line-based face recognition under varying pose, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 10, October 1999, pp. 1081–1088, ISSN 0162-8828.

Bax I.; Heidemann G., Ritter H. (2005). Face Detection and Identification Using a Hierarchical Feed-forward Recognition Architecture, *IEEE International Joint Conference on Neural Networks (IJCNN)*, Montreal, Canada, August 2005,pp. 1675-1680.

Blanz V. & Vetter T. (2003). Face Recognition Based on Fitting a 3D Morphable Model, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 9, September 2003, pp. 1063–1074, ISSN 0162-8828.

Brawn J. (2006). Development of Brain and Behavior in Infancy. Lecture 4: Perceptual Development, On line, http://www.lsbu.ac.uk/psycho/teaching/ppfiles/dbbi-l4-05-06.ppt.

Bower B. (2001). Faces of Perception, *Science News*, July 2001, Vol. 160, No. 1, pp. 10-12, ISSN 0036-8423.

Burton A. M.; Wilson S., Cowan M. & Bruce V. (1999). Face recognition in poor quality video, *Psychological Science*, Vol. 10, No. 3, May,1999, pp 243-248,ISSN 0956-7976.

Cevikalp H.; Neamtu M., Wilkes M. & Barkana A. (2005). Discriminative Common Vectors for Face Recognition, *IEEE Transactions on Pattern Analysis and Machinery Intelligence*, Vol. 27, No. 1, January 2005, pp. 4-13,ISSN 0162-8828.

Chacon M. I. & Ramirez G. (2006a), Wood Defects Classification Using A Som-ffp Approach with Minimum Dimension Feature Vector, *Lecture Notes in Computer Science*, Vol. 3973/2006,Springer, 2006, pp 1105-1110, ISSN 0302-9743.

Chacon M. I.; Rivas P. & Ramirez G. (2006b). A fuzzy clustering approach for face recognition based on face feature lines and eigenvectors, *Proceeding of the International Seminar on Computational Intelligence 2006*, October 2006.

Chen W.; Yuen P. C., Huang J. & Dai D. (2005). Kernel Machine-Based One-Parameter Regularized Fisher Discriminant Method for Face Recognition, *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, Vol. 35, No. 4, August 2005, pp. 659-669, ISSN 1083-4419.

Er M. J.; Chen W. & Wu S. (2005). High-Speed Face Recognition Based on Discrete Cosine Transform and RBF Neural Networks, *IEEE Transactions on Neural Networks*, Vol. 16, No. 3, May 2005, pp. 679-698,ISSN 1045-9227.

Gao Y. & Leung M. K. H. (2002). Face Recognition Using Line Edge Map, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 6, June 2002, pp. 764–779, ISSN 0162-8828.

Gauthier I. & Tanaka J. W. (2002). Configural and holistic face processing: The Whole story, *Journal of Vision*, November 2002, Vol. 2, No. 7, pp. 616-616, ISSN 1534-7362.

Goudail F.; Lange E., Iwamoto T., Kyuma K. & Otsu N. (1996). Face Recognition System Using Local Autocorrelations and Multiscale Integration, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 10, October 1996, pp. 1024–1028, ISSN 0162-8828 .

Gross R.; Matthews I. & Baker S. (2004). Appearance-Based Face Recognition and Light-Fields, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 4, April, 2004, pp. 449–465, ISSN 0162-8828.

Haddadnia J. & Ahmadi M. (2004). N-feature neural network human face recognition, *Image and Vision Computing*, Vol. 22,No. 22. October 2004, pp. 1071–1082, ISSN 0262-8856.

Hafed Z. M. & Levine M. D. (2001). Face Recognition Using the Discrete Cosine Transform, *International Journal of Computer Vision*, Vol. 43, No. 3, July 2001, pp. 167–188, ISSN 0920-5691.

Haxby J. V.; Gobbini M. I., Furey M. L., Ishai A., Schouten J. L. & Pietrini P. (2001). Distributed and Overlapping Representations of Faces and Objects in Ventral Temporal Cortex, *Science*, September 2001, Vol 293, No. 5539, pp. 2425 - 2430,ISSN 0036-8075.

He X.; Yan S., Hu Y., Niyogi P. & Zhang H. J. (2005). Face Recognition Using Laplacianfaces, *IEEE Transactions on Pattern Analysis and Machinery Intelligence*, Vol. 27, No. 3, March, 2005,pp. 328-340, ISSN 0162-8828.

Johnson M. H. (2001). The development and neural basis of face recognition: Comment an speculation, *Infant and Child Development*, Vol. 10, No. 1-2, March-June 2001,pp. 31-33, ISSN 1522-7219.

Kalocsai I. P. (1998). Neural and Psychophysical analysis of object and face recognition, *Face Recognition: From Theory to Applications*, Springer-Verlag, Berlin, Germany, 1998, pp. 3-25, ISBN 3-540-64410-5.

Kanwisher, J.; McDermott J. & Chun M. (1997). The fusiform face Area: A module in human Extrastriate Cortex Specialized for the Perception of Faces, *Journal Neurosciences*,Vol. 17, No. 11, June 1997,pp. 4302-4311, ISSN 0270-6474.

Kanwisher N. (2006). The Fusiform Face Area: A Module in Human Extrastriate Cortex Specialized for Face Perception, On line, http://www.arts.uwaterloo.ca /~jdancker /fMRI/oriet%20FFA.ppt

Kim T. & Kittler J. (2005). Locally Linear Discriminant Analysis for Multimodal Distributed Classes for Face Recognition with a Single Model Image, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 3, March, 2005, pp. 318-327,ISSN 0162-8828.

Kim C.; Oh J. & Choi C. (2005). Combined Subspace Method Using Global and Local Features for Face Recognition, *IEEE International Joint Conference on Neural Networks (IJCNN)*, Montreal, Canada, August 2005,pp.2030-2035.

Konen W. (1996). Neural information processing in real-world face recognition applications, *IEEE Expert*, Vol. 11, No. 4, Augusto 1996, pp. 7 – 8, ISSN: 0885-9000.

Kong H.; Li X., Wang L., Teoh E. K., Wang J. & Venkateswarlu R. (2005). Generalized 2D Principal Component Analysis, *IEEE International Joint Conference on Neural Networks (IJCNN)*, Montreal, Canada, August 2005,pp.108-113.

Kwak K. C. & Pedrycz W. (2004). Face Recognition Using Fuzzy Integral and Wavelet Decomposition Method, *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, Vol. 34, No. 4, August 2004, pp. 1666–1675, ISSN 1083-4419.

Lee K.; Ho J. & Kriegman D. J. (2005). Acquiring Linear Subspaces for Face Recognition under Variable Lighting, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 5, May 2005, pp. 684-698,ISSN 0162-8828.

Li S. Z. & Jain A. K. (2004). *Handbook of face recognition*, Springer, USA, 2004.

Liu C. (2004). Gabor-Based Kernel PCA with Fractional Power Polynomial Models for Face Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 5, May 2004, pp. 572–581, ISSN 0162-8828.

Lu J.; Plataniotis K. N. & Venetsanopoulos A. N. (2003). Face Recognition Using Kernel Direct Discriminant Analysis Algorithms, *IEEE Transactions on Neural Networks*, Vol. 14, No. 1, January 2003. pp. 117–126, ISSN 1045-9227.

Lu X. & Jain A. K. (2005). Deformation Analysis for 3D Face Matching, *Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision*, January 2005, Brekenridge, CO, USA ,Vol. 1,pp. 99-104.

Mareschal D. & Thomas M. S. C. (2006), How Computational Models Help Explain the Origins of Reasoning, *IEEE Computational Intelligence Magazine*, August 2006, Vol. 1, No. 3, pp 32-40, ISSN: 1556-603X.

Martinez A.M. & Benavente R. (1998). The AR Face Database, http://cobweb.ecn.purdue.edu/~aleix/ar.html.

Mio W.; Zhang Q. & Liu X. (2005). Nonlinearity and optimal component analysis, *IEEE International Joint Conference on Neural Networks (IJCNN)*, Montreal, Canada, August 2005,pp.220-225.

Moghaddam B. & Pentland A. (1998). Probabilistic Matching for Face Recognition, *IEEE Southwest Symposium on Image Analysis and Interpretation*, April 1998, pp. 186– 191.

Nelson C. A. (2001). The development and neural bases of face recognition, *Infant and Child Development*, March 2001, Vol. 10, pp. 3-18, ISSN 1522-7219.

Olivetti Research Lab (ORL) Database, On line (2006): http://www.uk.research.att.com/facedatabase.html

Quinn P. C.; Eimas P. D., & Tarr M. J. (2001). Perceptual categorization of cat and dog silhouettes by 3- to 4-month-old infants, *Journal of Experimental Child Psychology*, Vol. 79, No. 1, May 2001, pp. 79-78, ISSN: 0022-0965.

Peterson M. A. & Rhodes G. (2003). Perception of Faces, Objects, and Scenes, *Advances in visual cognition*, Oxford University Press, New York, USA. 2003.

Samaria F. & Harter A. (1994). Parameterization of a stochastic model for human face identification, *2nd IEEE Workshop on Applications of Computer Vision*, Sarasota (Florida), 1994.

Singh S.K.; Vatsa M., Singh R. & Chauhan D.S. (2002). A comparison of face recognition algorithms neural network based & line based approaches, *2002 IEEE International Conference on Systems, Man and Cybernetics*, Vol. 6, October 2002, pp. 6.

Singh S.K.; Vatsa M., Singh R. & Shukla K.K. (2003). A comparative study of various face recognition algorithms (feature based, eigen based, line based, neural network approaches), *2003 IEEE International Workshop on Computer Architectures for Machine Perception*, May 2003, pp. 160 – 171.

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*, Vol. 94, No. 11, November 2006, pp. 1948-1962.

Slater A. & Quinn P. C. (2001). Face recognition in the newborn infant, *Infant and Child Development*, March 2001, Vol. 10, No. 1, pp. 21-24, ISSN 1522-7219.

Tan T. & Yan H. (2005). Face Recognition Using the Weighted Fractal Neighbor Distance, *IEEE Transactions on Systems, Man, And Cybernetics—Part C: Applications And Reviews*, Vol. 35, No 3, November 2005, pp 1 – 7, ISSN 1094-6977.

Thompson P. (1980). Margaret Thatcher: A new illusion, *Perception*, 1980, Vol. 9, No.4, pp 483-484, ISSN 0301-0066.

Tong F., Nakayama K., Moscovitch M., Weinrib O. & Kanwisher N. (2000). Response properties of human fusiform area, *Cognitive Neuropsychol*, Vol. 17, No. 1, February 2000, pp. 257-279, ISSN: 1464-0627.

Yale University. (2002). Face Database. On line http://cvc.yale.edu/projects/yalefaces/yalefaces.html

Yan S.; He X., Hu Y., Zhang H., Li M. & Cheng Q. (2004). Bayesian Shape Localization for Face Recognition Using Global and Local Textures, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 14, No. 1, January 2004, pp. 102-113, ISSN: 1558-2205.

Yuen P. C.; Dai D. Q. & Feng G. C. (1998). Wavelet-based PCA for Human Face Recognition, *Proceeding of IEEE Southwest Symposium on Image Analysis and Interpretation*, pp 223–228, April 1998.

Wang X. & Tang X. (2004). A Unified Framework for Subspace Face Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 9, September 2004, pp. 1222–1228, ISSN 0162-8828.

Weyrauch B.; Huang J., Heisele B. & Blanz V. (2004). Component-based Face Recognition with 3D Morphable Models, *First IEEE Workshop on Face Processing in Video*, Washington, D.C., 2004.

Zhang M. & Fulcher J. (1996). Face Recognition Using Artificial Neural Network Group Based Adaptive Tolerance (GAT) Trees, *IEEE Transactions on Neural Networks*, Vol. 7, No. 3, May 1996, pp. 555 – 567, ISSN 1045-9227 .

Zhang B.; Zhang H. & Ge S. S. (2004). Face Recognition by Applying Wavelet Subband Representation and Kernel Associative Memory, *IEEE Transactions on Neural Networks*, Vol. 15, No. 1, January 2004, pp. 166–177, ISSN 1045-9227.

Zhang H.; Zhang B., Huang W. & Tian Q. (2005). Gabor Wavelet Associative Memory for Face Recognition, *IEEE Transactions on Neural Networks*, Vol. 16, No. 1, January 2005, pp. 275–278, ISSN 1045-9227.

Zhao W.; Chellappa R., Rosenfeld A. & and Phillips P. J.(2003). Face recognition: A literature survey, ACM Computing Surveys, Vol. 35, No. 4, pp. 399–458, 2003, ISSN 0360-0300.

Zheng W.; Lai J. & Yuen P. C. (2005). GA-Fisher: A New LDA-Based Face Recognition
    Algorithm With Selection of Principal Components, *IEEE Transactions on Systems,
    Man, and Cybernetics – Part B: Cybernetics*, Vol. 35, No. 5, October 2005, pp. 1065–
    1078, ISSN 1083-4419.
Zuo F. & de With P. H. N. (2005). Real-time Embedded Face Recognition for Smart Home,
    IEEE Transactions on Consumer Electronics, Vol. 51, No. 1, February 2005, pp. 183-190
    ISSN 0098-3063.

Gender Classification by Information Fusion of Hair and Face

Zheng Ji, Xiao-Chen Lian and Bao-Liang Lu

Department of Computer Science and Engineering, Shanghai Jiao Tong University 800 Dong Chuan Road, Shanghai 200240, China

1. Introduction

Various gender classification methods have been reported in the literature. These existing methods fall into two categories. The first kind of method is the appearance-based approach. Golomb *et al.* [1] used a two-layer neural network with 30 × 30 inputs and directly fed the scaled image pixels to the network without dimensionality reduction. Their database contains only 90 images with half male and half female facial images. Gutta *et al.* [2] used the mixture of experts combining the ensembles of radial basis functions (RBF) networks and a decision tree. Xu *et al.* [3] applied Adaboost to gender classification problem with the feature pools composed of a set of linear projections utilizing statistical moments up to second order. Wu *et al.* [4] also adopted Adaboost. Instead of using threshold weak classifiers, they used looking-up table weak classifiers, which are more general and better than simple threshold ones due to stronger ability to model complex distribution of training samples. Moghaddam and Yang [5] demonstrated that support vector machines (SVMs) work better than other classifiers such as ensemble of radial basis function (RBF) networks, classical RBF networks, Fisher linear discriminant, and nearest neighbor. In their experiments, the Gaussian kernel works better than linear and polynomial kernels. However, they did not discuss how to set the hyper-parameters for Gaussian kernel, which affect the classification performance. Kim *et al.* [6] applied Gaussian process technique to gender classification. The advantage of this approach is that it can automatically determine the hyper-parameters.Wu *et al.* [7] presented a statistical framework based on the 2.5D facial needle-maps which is a shape representation acquired from 2D intensity images using shape from shading (SFS). Saatci and Town [8] used an active appearance model (AAM) of faces to extract facial features and developed a cascaded structure of SVMs for gender classification. Lian and Lu applied min-max modular support vector machine to gender classification and developed a method for incorporating age information into task decomposition [9]. They also proposed a multi-resolution local binary pattern for dealing with multi-view gender classification probelms [10].

The second kind of approach is the geometrical feature based approach. The idea is to extract from faces geometric features such as distances, face width, and face length. Burton *et al.* [11] extracted point-to-point distances from 73 points on face images and used discriminant analysis as a classifier. Brunelli and Poggio [12] computed 16 geometric

features, such as pupil to eye brow separation and eye brow thickness, from the frontal images of a face and used HyperBF network as a classifier.

Fig. 1. Illustration of the important role of hair information for gender classification. The upper row denotes three facial images of female, whose hair regions have been discarded, and the lower row denotes the same three images with hair.

Most of the existing methods mentioned above, however, use only facial information. As we know, external information such as hair and clothing, also provide the discriminant evidence. Fig. 1 illustrates the benefit of incorporating hair information in gender classification task. The upper row shows the facial images whose hair regions were discarded. The lower row shows the corresponding original pictures. It is difficult for us to judge the gender when we only see the images in the upper row since their neutral-like faces. But, when both the facial information and the external information of hair as shown in the lower row are presented, we can easily make the decision.

Although external features are useful, their detection, representation, analysis, and application have seldom been studied in the computer vision community. Considering the important role of hair features in gender classification, we study hair feature extraction and the combination of hair classifier and face classifier in this Chapter. Given a facial image containing both hair and face, we first locate hair region and face region. We construct a geometric hair model (GHM) to extract hair features and use local binary pattern (LBP) to extract facial features. After performing these feature extraction, we train two different classifiers for each kind of features and then apply a classifier fusion model. The key issue of classifier fusion is to determine how classifiers interact with each other. In this study, we adopt fuzzy integral [13], which has the advantage of its automatical adaptation of degree of classifier interaction.

We conduct experiments on three popular facial image databases: AR, FERET and CAS-PEAL. The experimental results demonstrate that the combination of hair and face classifiers achieves much higher classification rate than hair classifier or face classifier along.

The rest of this Chapter is organized as follows: Section 2 describes the feature extraction process of hair and face. Section 3 introduces the fuzzy integral method. Section 4 gives experimental details. Section 5 is the conclusions.

2. Feature extraction

2.1 Hair feature extraction

Hair is a highly variable feature of human appearance. It perhaps is the most variant aspect of human appearance. Until recently, hair features have often been discarded in most of the gender classification systems. To our best knowledge, there are two different algorithms in the literature about hair feature representation. Yacoob *et al.* [14] developed a computational model for measuring hair appearance. They extracted several attributes of hair including color, split location, volume, length, area, symmetry, inner and outer hairlines, and texture. These attributes are organized as a hair feature vector. Lapedriza *et al.* [15] learned a model set composed by a representative set of image fragments corresponding to hair zones called building blocks set. The building blocks set is used to represent the unseen image as it is a set of puzzle pieces and the unseen image is reconstructed by covering it with the most similar fragments. By using this approach, the hair information is encoded and used for classification. We adopt the former method and modify it in this study. The overall process of hair feature extraction is illustrated in Fig. 2.

Fig. 2. The overall process of hair feature extraction.

Geometric Hair Model The basic symbols used in the geometric hair model are depicted in Fig. 3. Here G is the middle point between the left eye point L and the right eye point R, I is the point on the inner contour, O is the point on the outer contour, and P is the lowest point of hair region.

Hair Detection In this work, we assume that the facial images used are in frontal view. The hair detection process is illustrated in Fig. 4. The detection algorithm consists of the following four steps:

1. Locate three landmarks on each facial image shown in the second picture of Fig. 4. Two are centers of eyes and the other one is middle of hair. These three points facilitate hair region extraction. Currently we label them manually. These points can be easily located in an automated manner, providing that the locations of the eyes are given.
2. Obtain binary facial image. The pixels around the landmark in hair region form the seeds to separate hair from face and background.
3. Get the edge image of the hair by Laplace operator to hair edge detection. The 2D Laplace operator is $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$, and the edge extraction result is shown in the lower right image of Fig. 4.
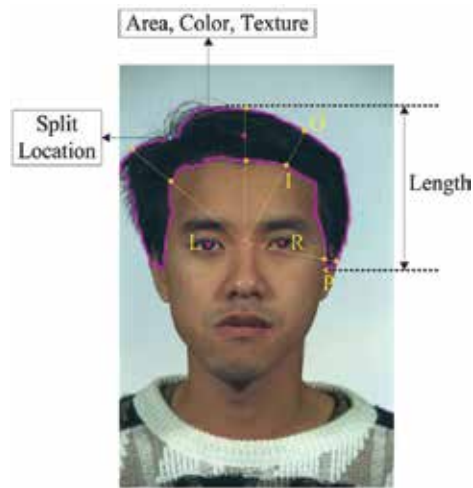
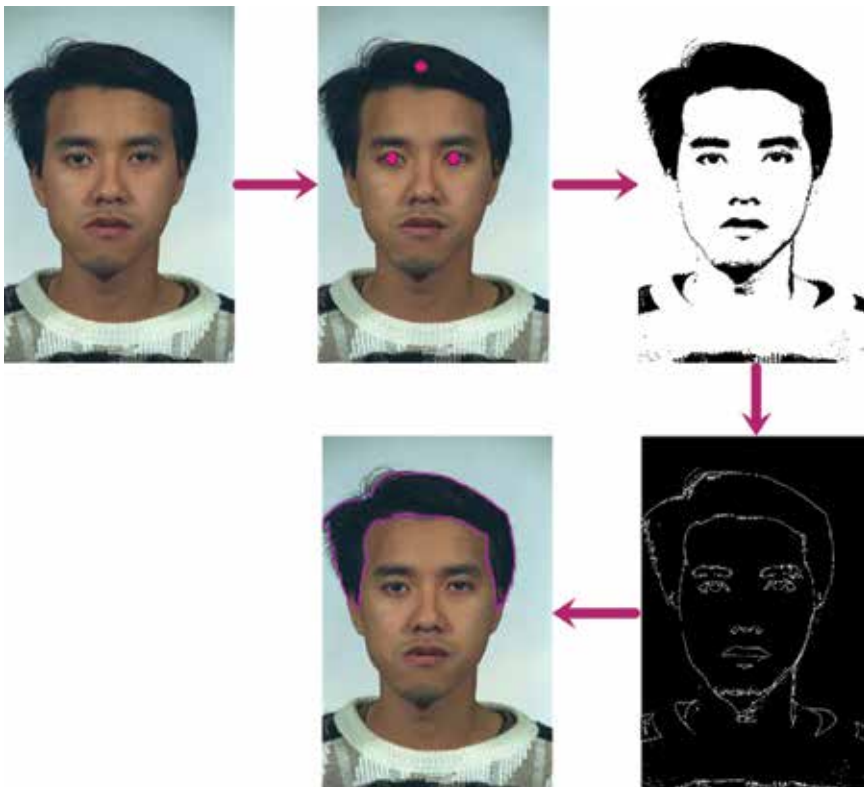Fig. 3. Some key parameters in geometric hair model.

Fig. 4. The process of hair detection.

4. Extract the inner and outer contour of the hair. Given an edge image from step 3), a ray denoted by the lines (yellow) in Fig. 3 is emitted from the mid-point of left and right eyes. In the ideal situation, the ray will meet exactly two edge points, *I* and *O* depicted in Fig. 3, respectively. By making a full rotation of the ray, we can determine the

contours of hair. In practice, however, step 2) may produce some holes in hair region, which make the detected contour inaccurate. To overcome this problem, we notice that the holes that will greatly affect the contour accuracy are those far away from the real contour and the distance between points *I* and *O* of consecutive rays will not change sharply. Based on this observation, we select the last edge point that the ray meets as the outer contour point. By using this technique, the holes can be removed. As a result, the detected contour of hair becomes more accurate. This improvement is illustrated in Fig. 5.

Fig. 5. The original detected contour of hair (left) versus the detected contour in which the holes have been removed (right).

Hair Length and Area We define the largest distance between a point on the outer contour and *P* as the hair length. The normalized distance L_{hair} is defined as

$$L_{hair} = max(dist(O_y, P_y))/Girth_{face} \tag{1}$$

where $Girth_{face}$ is the girth of the face region.
We define the area covered by hair as the hair surface. Based on the hair model, the normalized hair area is defined as

$$Area_{hair} = R\_Area_{hair}/R\_Area_{face} \tag{2}$$

where R_Area_{hair} is the real area of hair and R_Area_{face} is the area of face.
Hair Color To obtain the color in the hair region, we follow the method described in [16]. Based on this color model depicted in Fig. 6, the measured color results from the brightness and surface spectral reflectance. The averaged color distortion is calculated by

$$\overline{CD} = \frac{\sum_{i \in H} \|I_i - \alpha_i E_i\|}{|H|} \tag{3}$$

where *H* is the pixel set of hair region, and I_i and E_i denote the actual RGB color and the expected RGB color at pixel *i*, respectively, as follows:

$$I_i = (I_r(i), I_g(i), I_b(i)) \tag{4}$$
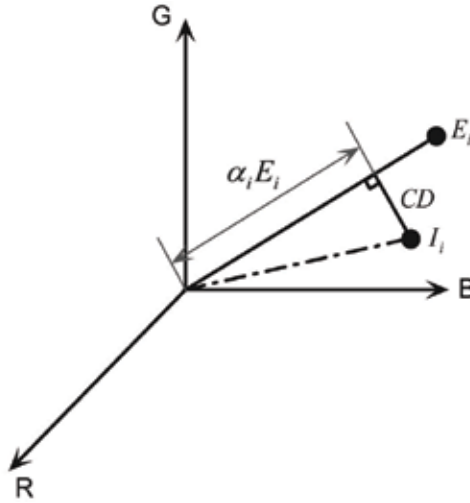
$$E_i = (E_r(i), E_g(i), E_b(i)) \tag{5}$$

Fig. 6. Color model of hair.

According to the definitions mentioned above, the color distortion CD_i at pixel i can be computed by

$$CD_i = \sqrt{(\frac{I_r(i) - \alpha_i \mu_r}{\sigma_r})^2 + (\frac{I_g(i) - \alpha_i \mu_g}{\sigma_g})^2 + (\frac{I_b(i) - \alpha_i \mu_b}{\sigma_b})^2} \tag{6}$$

where α_i represents the current brightness with respect to the brightness of the model

$$\alpha_i = \frac{(I_r(i)\frac{\mu_r}{\sigma_r})^2 + (I_g(i)\frac{\mu_g}{\sigma_g})^2 + (I_b(i)\frac{\mu_b}{\sigma_b})^2}{(\frac{\mu_r}{\sigma_r})^2 + (\frac{\mu_g}{\sigma_g})^2 + (\frac{\mu_b}{\sigma_b})^2}, \tag{7}$$

and (μ_r, μ_g, μ_b) and $(\sigma_r, \sigma_g, \sigma_b)$ are the mean and standard deviation of color in the training set, respectively.

Hair Texture We employ Gabor wavelets to compute the hair feature attributes that characterize hair texture. The following two-dimensional Gabor function $g(x, y)$ under 6 directions and 4 scales and its Fourier transform $G(u, v)$ are used,

$$g(x, y) = (\frac{1}{2\pi\sigma_x\sigma_y})exp[-\frac{1}{2}(\frac{x^2}{\sigma_x} + \frac{y^2}{\sigma_y}) + 2\pi j W_x] \tag{8}$$

and

$$G(u, v) = exp-\frac{1}{2}[(\frac{u - W}{\sigma_u})^2 + (\frac{v}{\sigma_u})^2] \tag{9}$$

where $\sigma_u = \frac{1}{2\pi\sigma_x}$ and $\sigma_v = \frac{1}{2\pi\sigma_y}$.

The discrete form of Gabor wavelet transformation is defined as

$$W_{mn} = \iint I(x_l, y_l) g_{mn}^*(x - x_l, y - y_l) dx_l dy_l \tag{10}$$

where g_{mn}^* represents the conjugate operation in complex area of the *m*-th orientation and *n*-th scale.

We use the mean value and standard deviation of Gabor parameters to represent the texture shown in Fig. 7. The mean value and standard deviation are, respectively, calculated by

$$\mu_{mn} = \iint | W_{mn}(xy) | \tag{11}$$

and

$$\sigma_{mn} = \sqrt{\iint (| W_{mn}(x, y) - \mu_{mn} |)^2 dx dy}. \tag{12}$$
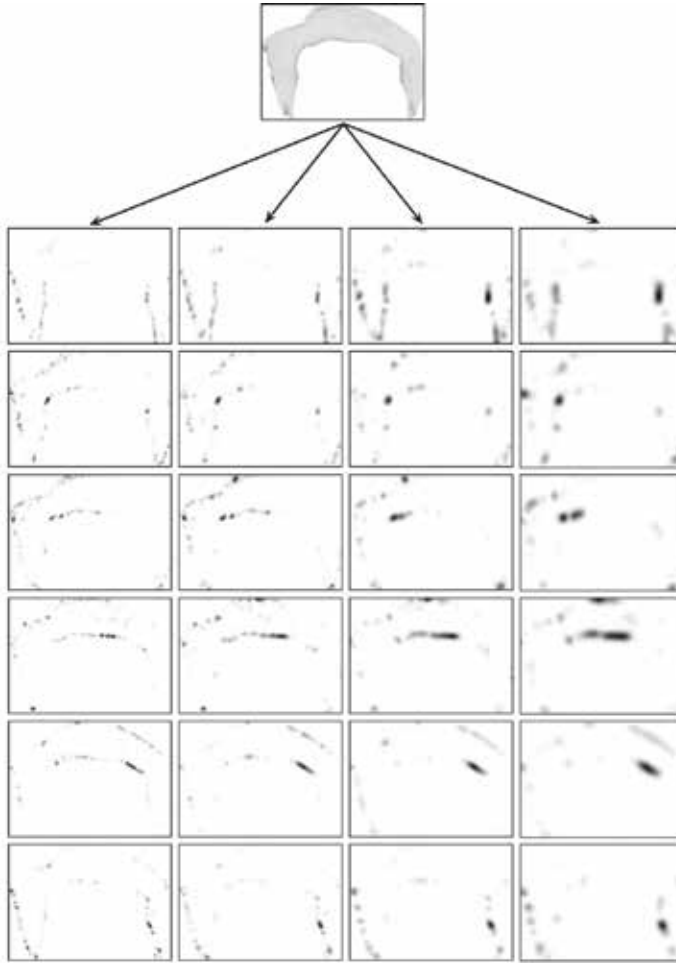

Fig. 7. Extracting hair texture using wavelet transformation with 4 scales and 6 orientations.

From Eqs. 11 and 12, we have the following feature attributes of hair texture:

$$V = [\mu_1, \sigma_1, \mu_2, \sigma_2, \mu_3, \sigma_3, \cdots, \mu_{24}, \sigma_{24}] \tag{13}$$

Hair-Split Location The hair split location is commonly accompanied by a concavity point at the outer hairline. The split angle is defined as the angle of the concavity point with respect to the horizontal axis. Whether a point P is a concavity point can be judged by

$$P = \begin{cases} concavity, & \overrightarrow{P_{n-1}P_n} \times \overrightarrow{P_nP_{n+1}} < 0 \\ non-concavity, & \overrightarrow{P_{n-1}P_n} \times \overrightarrow{P_nP_{n+1}} > 0 \end{cases} \tag{14}$$

We define *concavity*[P] as the concavity of point P, which can be calculated by

$$concavity[P] = -\frac{\overrightarrow{P_{n-1}P_n} \times \overrightarrow{P_nP_{n+1}}}{|\overrightarrow{P_{n-1}P_n}| |\overrightarrow{P_nP_{n+1}}|} \tag{15}$$

With this definition, we scan all the edge points on the outer contour, compute the average concavity and select the one with largest *concavity*[P] as the split point.

By concatenating all the hair feature attributes mentioned above, we obtain a feature vector of hair as follows:

$$\begin{aligned} Hair_{Vector} &= [length, area, color, split\_location, texture] \\ &= [L_{hair}, Area_{hair}, E_r, E_g, E_b, concavity[P], \\ &\quad \mu_1, \sigma_1, \mu_2, \sigma_2, \mu_3, \sigma_3, \cdots, \mu_{24}, \sigma_{24}] \end{aligned} \tag{16}$$

2.2 Face feature extraction

We use LBP [17] to characterize the face feature. The overall process is illustrated in Fig. 8. LBP is a simple and efficient approach for texture description. The operator labels the pixels of an image by thresholding 3 × 3-neighbourhoods of each pixel with the center value and considering the result as a binary number. The histogram of the labels is used as a texture descriptor. The basic LBP operator is illustrated in Fig. 9.

To achieve rotation invariance, An extension to the original operator is to use so called uniform pattern. A local binary pattern is called a uniform pattern if it contains at most two bitwise transitions from 0 to 1 or vice versa when the binary string is considered circular. For example, 00000000, 00011110, and 10000011 are three uniform patterns.

We use the notation of $LBP_{P,R}^{u2}$ for the LBP operator. Here, $LBP_{P,R}^{u2}$ means using the operator in a neighborhood of P sampling points on a circle of radius R, and the superscript $u2$ represents using uniform patterns and labeling all remaining patterns with a single label. In our experiment, $\mathbf{LBP}_{8,1}^{u2}$ operator is used to quantify the total of 256 LBP values into the histogram of 59 bins according to the uniform strategy.

This histogram contains information about the distribution of the local micro-patterns over the whole image such as edges, spots and flat areas. For efficient face representation, one should also retain spatial information. For this purpose, an image is spatially divided into m small regions, $\mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_m$, and the spatially enhanced histogram for region $\mathbf{R}_j$ is defined as

$$\mathbf{h}_{i,j} = \sum \mathbf{I}\{\mathbf{f}_l(\mathbf{x}, \mathbf{y}) = i\}\mathbf{I}\{(\mathbf{x}, \mathbf{y}) \in \mathbf{R}_j\},\ i = 1, 2, ..., L,\ j = 1, 2, ..., m \qquad (17)$$

where L is the number of different labels produced by LBP operator, m is the number of blocks of the divided image, and $I\{A\}$ is 1 or 0 depending on whether A is true or false. According to Eq. 17, we obtain the following face feature vector:

$$\mathbf{H} = (\mathbf{h}_{1,1}, ..., \mathbf{h}_{L,1}, ..., \mathbf{h}_{1,m}, ..., \mathbf{h}_{L,m}) \qquad (18)$$
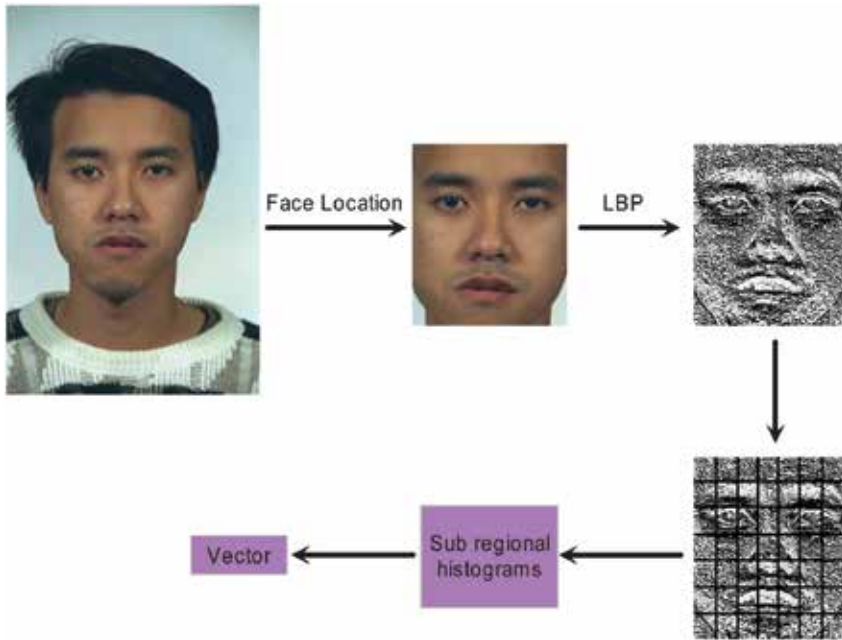
Fig. 8. The overall process of LBP feature extraction.

Binary: 10001010
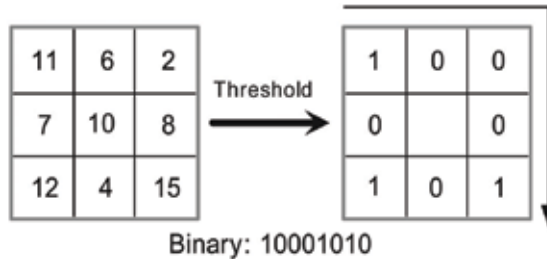
Fig. 9. The 3 × 3-neighborhood LBP operator.

3. Fuzzy integral fusion of support vector machine classifiers

The ultimate goal of gender classification systems is to achieve the best possible classification performance. This objective traditionally led to the development of different classification schemes. Although the common way then is to choose one of the design that would yield the best performance, the sets of patterns misclassified by the different classifiers would not necessarily overlap. This suggested that different classifier designs

potentially offered complementary information about the patterns to be classified which could be harnessed to improve the performance of the selected classifier [18].

Here, we introduce a classifier fusion method based on fuzzy integral proposed by Sugeno [13]. The distinguish characteristic of fuzzy integral is that it is able to represent a certain kind of interaction between criteria, which is always avoided by making classifiers independent. Given a set of classifiers and their importance, fuzzy integral evaluates the interaction of these classifiers by computing fuzzy measure, a real function defined on the power set of classifiers. Based on such function and considering each classifier's decision, fuzzy integral will give a final decision.

Before describing the proposed fusion method, we present a note on notation. Let $\mathcal{C} = \{c_1, \ldots, c_M\}$ be the set of classes and we have K classifiers, $f_1, \ldots, f_K$. Each of the K classifiers provides for an unknown sample X a degree of confidence in the statement 'X belongs to class c_j', for all c_j. We denote by $f_i^j(X)$ the confidence degree delivered by the classifier i of X belonging to c_j.

3.1 Probabilistic output of SVMs

We choose support vector machine as the basic classifier. Two SVMs are trained on hair and face features, respectively. Since fuzzy integral requires each classifier to give confidence value, we need to convert the binary output to probabilistic output. As the task of gender classification is a two-class problem, we assume that the training set is

$$\mathcal{T} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\} \tag{19}$$

where $x_i \in R_n$, $y_i \in \{-1, 1\}$, and $i = 1, 2, \cdots, N$.

A classifier $f(x)$ in the margin form of SVMs is equivalent to solving the following convex quadratic programming problem [19–21]

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \\ \text{s.t. } \forall 1 &\leqslant i \leqslant N, y_i(\mathbf{w}^T\mathbf{x}_i + b) \geqslant 1 - \xi_i, \\ & \xi_i \geqslant 0 \end{aligned} \tag{20}$$

and its dual form:

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i y_i \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{N}\alpha_i \\ \text{s.t. } & \forall 1 \leqslant i \leqslant N, \quad 0 \leqslant \alpha_i < C, \\ & \sum_{i=1}^{N}\alpha_i y_i = 0 \end{aligned} \tag{21}$$

SVM classifier $f(x)$ finds the optimal hyperplane that correctly separates the training data while maximizing the margin. Therefore, there is the following discriminant hyperplane:

$$f(x) = \sum_{i=1}^{N}\alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{22}$$

where $K(\cdot, \cdot)$ is a kernel function and b is a bias.
Now $f(x)$ can be projected onto [0, 1] by sigmoid function

$$P_{+1}(x) = P(y = 1|x) = \frac{1}{1 + e^{Af(x)+B}}$$

$$P_{-1}(x) = 1 - P_{+1}(x)$$

(23)

The parameters A and B [22] can be estimated by solving the following maximum likelihood problem [23]:

$$\min_{A,B} F(A, B) = -\sum_{j=1}^{N}(t_i log(P_{+1}(x_i)) + (1 - t_i)log(P_{-1}(x_i)))$$

$$\text{s.t. } \forall 1 \leqslant i \leqslant N, t_i = \begin{cases} \frac{N_{+}+1}{N_{+}+2}, & \text{if } y_i = 1 \\ \frac{1}{N_{-}+2}, & \text{if } y_i = -1 \end{cases}$$

(24)

where N_+ is the number of training samples with $y = 1$ and N_- is the number of training samples with $y = -1$.

3.2 Fuzzy integral theory

Fuzzy measures are the generalization of classical measures. The notion of a measure in an Euclidean space is a natural generalization of such elementary notions as the length of a line segment, the area of a rectangle and the volume of a parallelepiped. Still a more general concept of a measure in an arbitrary abstract set can be defined [24–26] as

Definition 1: By a measurable space we mean a pair (X,Ω) consisting of a set X and a σ-algebra of subsets of X. A subset A of X is called measurable (or measurable with respect to Ω) if $A \in \Omega$.

Definition 2: A measure μ on a measurable space (X,Ω) is a real non-negative set function defined for all sets of Ω such that $\mu(\emptyset) = 0$, and if $\{A_i\}_{i=1}^{\infty}$ is a disjoint family of sets with $A \in \Omega, i \geq 1$, then

$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mu(A_i).$$

(25)

It can be shown that a measure μ has the following properties [24]:
1. $\mu(A) \leq \mu(B)$ if $A \subset B$.
2. If $\{A_i\}_{i=1}^{\infty}$ is an increasing sequence of measurable sets, then

$$\lim_{i \to \infty} \mu(A_i) = \mu(\lim_{i \to \infty} A_i).$$

(26)

An important example of such a measure is the probability measure P, where $P(X) = 1$.
In the seventies of the twentieth century, alternative models were proposed by different researchers [27–30], who all share the following intuitively reasonable axioms:
Definition 3: Let $g : \Omega$ [0, 1] be a set function with
1. $g(\emptyset) = 0, g(X) = 1,$

2. $g(A) \leq g(B)$ if $A \subset B$,
3. If $\{A_i\}_{i=1}^{\infty}$ is an increasing sequence of measurable sets, then

$$\lim_{i \to \infty} g(A_i) = g(\lim_{i \to \infty} A_i). \tag{27}$$

Such a function is called a fuzzy measure [29].

By the nature of the definition of a fuzzy measure g, the measure of the union of two disjoint subsets cannot be directly computed from the component measures. In light of this, Sugeno [29] introduced the so-called g_λ-fuzzy measures satisfying the property as follows:

$$\forall A, B \subset X, \ \ A \cap B = \emptyset,$$
$$\text{s.t.} \ \ g(A \cup B) = g(A) + g(B) + \lambda g(A)g(B), \ \ \lambda > -1. \tag{28}$$

A g_λ-fuzzy measure is indeed a fuzzy measure, and the g_λ-fuzzy measure for $\lambda = 0$ is a probability measure [26, 31].

The constant λ can be determined by solving the following equation:

$$\lambda + 1 = \prod_{i}^{n} (1 + \lambda g^i) \tag{29}$$

In which, for a fixed set of $\{g^i = g_\lambda(s_i), 0 < g^i < 1\}$ where s_i is the ith classifier and g^i is a fuzzy density, there exists a unique root of $\lambda > -1$, $\lambda \neq 0$.

Let $S = \{s_1, s_2, \cdots, s_K\}$ be a finite set of individual SVM classifiers and $0 \leq h(s_1) \leq h(s_2) \leq \cdots \leq h(s_K) \leq 1$, where $h(s_i)$ is the probabilistic output of SVM classifier s_i $(1 \leq i \leq K)$ and the Choquet integral can be computed by

$$\int_A h(s) dg(\cdot) = \sum_{i=1}^{K} [h(s_i) - h(s_{i-1})] g_\lambda(A_i) \tag{30}$$

where $A_i = \{s_1, s_2, ..., s_i\}$, $i = 1, 2, ...,K$, $(h(s_0) = 0)$. Therefore, the value of $g_\lambda(A_i)$ can be computed recursively by

$$g_\lambda(A_1) = g_\lambda(s_1) = g^1$$
$$g_\lambda(A_i) = g^i + g_\lambda(A_{i-1}) + \lambda g^i g_\lambda(A_{i-1}) \tag{31}$$

3.3 Fusion of SVM classifiers

Now for each f_k, the degree of importance g^k, of how important f_k is in the recognition of the class c_i must be determined. Hence λ can be calculated using Eq. (29) and the g_λ-fuzzy measure can be determined by a confusion matrix (CM) in Eq. (32) of the SVM classifier s_k which can be computed by means of n-flod validation method,

$$CM^k = \begin{bmatrix} n_{11}^k & n_{12}^k & \cdots & n_{1M}^k \\ n_{21}^k & n_{22}^k & \cdots & n_{2M}^k \\ \vdots & \vdots & \ddots & \vdots \\ n_{M1}^k & n_{M2}^k & \cdots & n_{MM}^k \end{bmatrix} \tag{32}$$

where n_{ij}^k is the number of the samples which are the class c_i assigned to the class c_j by the classifier s_k. Therefore, the $\boldsymbol{g}_\lambda$-fuzzy measure can be calculated by the following formulation:

$$g^k = \frac{\sum_{i=1}^m n_{ii}^k}{\sum_{i=1}^m \sum_{j=1}^m n_{ij}^k} \tag{33}$$

and

$$g_i^k = \frac{n_{ii}^k}{\sum_{j=1}^m n_{ji}^k}, i = 1, 2, \cdots, m, k = 1, 2, \cdots, K \tag{34}$$

where $\boldsymbol{g}^k$ is the fuzzy density for the SVM classifier s_k and g_i^k represent the fuzzy density for the class c_i and the SVM classifier s_k. According to the above statement, we conclude the following algorithm for fuzzy integral fusion of SVM (FIF-SVM) classifiers.

> **Algorithm:** Fuzzy integral fusion of SVM classifiers.
> *step 1:* calculate λ using Eq. (29) after determining g^k or g_i^k in Eq. (34);
> *step 2:* **for** each class c_i **do**
>   **for** each SVM classifier s_k **do**
>   calculate $h_i(s_k)$ using Eqs. (22) and (23);
>   determine $g_i(A_k)$ by Eq. (32);
>   **end_for**
>   compute the fuzzy integral $f(c_i)$ for the class c_i by Eq. (30);
> **end_for**
> *step 3:* Output $argmax\{f(c_1), f(c_2), \cdots, f(c_M)\}$

4. Experiments

4.1 Databases
In this work, a total number of 2608 frontal facial images were selected from three popular face databases. Among them, 481 male and 161 female images were selected from the AR database randomly; 595 male and 445 female frontal facial images are chosen from the CAS-PEAL face database [32]; and 583 male and 406 female facial images are selected from the FERET face database. These facial images are described in Table 1.

Database	Total data	Male	Female	Training	Test
AR	579	481	161	128*2	323
CAS-PEAL	1040	595	445	356*2	328
FERET	989	583	406	324*2	341
Total	2608	1596	1012	808*2	992

Table 1. The data set.

4.2 Fuzzy measure values
In Table 2, the fuzzy measure values of all the classes based on face feature are larger than those of hair feature. However, in our experiments, the fuzzy measure depends on the confusion matrix which can be acquired by 5-fold validation after training. The classifier which takes on good classification performance after cross validation can achieve a high fuzzy measure for each class. The reason is that the confusion matrix is a detailed description form of classification accuracy. Because the classification precision of the gender

classifier based on face feature is higher than hair feature in the course of cross validation, the fuzzy measure favors the SVM (face) classifier. Generally speaking, there are different mapping parameters, A and B in Eq. (23), for different classifiers, which are independent in the process of training and test mutually. At the same time, there is $\lambda > -1$ in most situations. However, there exists $\lambda = -1$ in Table 2 because of some classification accuracy values equal to 100% after cross validation. Therefore, if the classification precision is less than 100%, there exists $\lambda > -1$.

No.	Database	Kernel	g^1 (hair)		g^2 (face)		A		λ	
			male	female	male	female	hair	face	male	female
1		Linear	0.8931	0.9120	1.0000	1.0000	-1.9693	-2.5027	-1.0000	-1.0000
2	AR	Poly	0.9370	0.9302	1.0000	0.9922	-3.1204	-4.9165	-1.0000	-0.9994
3		RBF	0.9297	0.9297	1.0000	1.0000	-2.8025	-5.5546	-1.0000	-1.0000
4		Linear	0.8391	0.8859	0.9581	0.9633	-0.4496	-1.7323	-0.9916	-0.9951
5	CAS-PEAL	Poly	0.8711	0.8732	0.9583	0.9688	-0.4274	-3.7717	-0.9936	-0.9953
6		RBF	0.8837	0.8587	0.9583	0.9688	-0.5357	-4.2729	-0.9943	-0.9947
7		Linear	0.8864	0.8701	0.9179	0.9310	-0.5231	-3.6407	-0.9885	-0.9889
8	FERET	Poly	0.9025	0.8879	0.9067	0.9574	-0.3453	-4.1190	-0.9889	-0.9944
9		RBF	0.9143	0.8919	0.9196	0.9519	-0.8303	-7.5197	-0.9918	-0.9939

Table 2. Values of g_λ-fuzzy measure, parameter A ($B = 0$) and λ.

4.3 Classification results

In our experiments, both the proposed geometric hair model and the LBP approach are used to extract hair and face features, respectively, from all the training samples. Two kinds of SVM classifiers, namely hair classifier and face classifier, are trained on the given data sets shown in Table 1. The results of *five*-fold validation are employed to calculate the confusion matrix, which determines the g_λ-fuzzy measure values of these two SVM classifiers. The classification accuracy of three different classifiers on the test data is described in Table 3.

No.	Database	Kernel	Hair	Face	Face & Hair
1		Linear	84.52	83.90	**93.19**
2	AR	Poly	83.59	89.16	**93.19**
3		RBF	84.52	85.45	**93.50**
4		Linear	89.02	96.34	**96.95**
5	CAS-PEAL	Poly	90.55	96.34	**97.87**
6		RBF	88.72	96.95	**98.48**
7		Linear	81.23	93.26	**94.43**
8	FERET	Poly	84.16	93.26	**94.13**
9		RBF	84.16	92.67	**94.13**

Table 3. Accuracy (%) of three different classifiers: face SVM-classifier along, hair SVM-classifier along, and fuzzy integral fusion of face and hair SVM-classifiers.

From Table 3, we can see that the proposed fuzzy integral fusion method achieves the best classification accuracy in all the cases. It should be noted that when the classification accuracy of both hair classifier and face classifier are relatively lower, the proposed fusion method can dramatically improve the classification accuracy. At the same time, we can see that the face classifier has higher classification accuracy than that of the hair classifier. This indicates that internal features such as face feature are more critical to gender classification than external features such as hair feature. On the other hand, hair features play a good complementary role for gender classification.

5. Conclusions

We have presented a modified geometric hair model for extracting hair features for gender classification. By using this model, hair features are represented as length, area, color, split-location, and texture. In order to integrate the outputs of both hair classifier and face classifier that use hair features and face features, respectively, we have proposed a classifier fusion approach based on fuzzy integral theory. The experimental results on three popular face databases demonstrate the effectiveness of the modified geometric hair model and the proposed classifier fusion method. From the experimental results, we can obtain the following observations. a) Hair features play an important role in gender classification; b) Face features are more critical than hair features to gender classification; c) Implementing the fusion of hair and face classifiers can achieve the best classification accuracy in all of the cases; d) The proposed fusion method can improve the classification accuracy dramatically when the performance of all the single classifier is not good. From this study, we believe that more external features such as hair and clothes should be integrated into face features to develop more reliable and robust gender classification systems.

6. Acknowledgments

7. References

[1] Golomb, B.A., Lawrence, D.T., Sejnowski, T.J.: Sexnet: A neural network identifies sex from human faces. In: Advances in neural information processing systems 3, Morgan Kaufmann Publishers Inc. 572–577 (1990)

[2] S. Gutta, J.R.J. Huang, P.J., Wechsler, H.: Mixture of experts for classification of gender, ethnic origin, and pose of human faces. IEEE Trans. Neural Networks 11(4) 948–960 (2000)

[3] Xu, X.H., S., T.: Soda-boosting and its application to gender recognition. Lecture Notes in Computer Science (4778) 193–204 (2007)

[4] Wu, B., Ai, H., Huang, C.: Lut-based adaboost for gender classification. In: AVBPA03. 104–110 (2003)

[5] Moghaddam, B., Yang, M.: Learning gender with support faces. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(5) 707–711 (May 2002)

[6] Kim, H., Kim, D., Ghahramani, Z., Bang, S.: Appearance-based gender classification with gaussian processes. Pattern Recognition Letters 27(6) 618–626 (April 2006)

[7] Wu, J., Smith, W., Hancock, E.: Gender classification using shape from shading. In: BMVC07. xx–yy (2007)

[8] Saatci, Y., Town, C.: Cascaded classification of gender and facial expression using active appearance models. In: FGR06. 393–400 (2006)

[9] Lian, H.C., Lu, B.L.: Gender recognition using a min-max modular support vector machine. The First International Conference on Natural Computation 433–436 (2005)

[10] Lian, H.C., Lu, B.L.: Multi-view gender classification using multi-resolution local binary patterns and support vector machines. International Journal of Neural System 17(6) 479–487 (2007)

[11] Burton, A., Bruce, V., Dench, N.: What's the difference between men and women? Evidence from facial measurement. Perception 22(2) 153–76 (1993)

[12] R.Brunelli, Poggio, T.: Hyperbf networks for gender classification. In: DARPA Image Understanding Workshop. 311–314 (1992)

[13] Sugeno, M.: Theory of fuzzy integrals and its applications. PhD thesis, Tokyo Institute of Technology (1974)

[14] Yacoob, Y.: Detection and analysis of hair. IEEE Trans. Pattern Anal. Mach. Intell. 28(7) 1164–1169 (2006) Member-Larry S. Davis.

[15] Lapedriza, A., Masip, D., Vitria, J.: Are external face features useful for automatic face classification? In: CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) -Workshops, Washington, DC, USA, IEEE Computer Society 151 (2005)

[16] Yacoob, Y., Davis, L.: Detection, analysis and matching of hair. The tenth IEEE International Conference on Computer Vision 1 741–748 (2005)

[17] Ojala, T., Pietik˙aainen, M., M˙aaenp˙aa˙aaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(7) 971–987 (2002)

[18] Kittler, J., Hatef, M., Duin, R., Matas, J.: On combining classifiers. Pattern Analysis and Machine Intelligence, IEEE Transactions on 20(3) 226–239 (1998)

[19] Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines and other kernel-based learning methods. Publishing House of Electronics Industry (2004)

[20] Sloin, A., Burshtein, D.: Support vector machine training for improved hidden markov modeling. IEEE Transactions on Signal Processing 56(1) 172--188 (2008)

[21] Williams, P., Li, S., Feng, J., Wu, S.: A geometrical method to improve performance of the support vector machine. IEEE Transactions on Neural Networks 18(3) 942--947 (2007)

[22] Platt, J.: Probabilistic outputs for support vector machines and comparison to regularized likelihood method. In Advance in Large Margin Classifiers, A. Smola, P. Bartlett, B. Scholkopf, D. Schuurmans, Eds, Cambridge, MA:MIT Press (2000)

[23] Zhang, Y.Z., Liu, C.M., Zhu, L.K., Hu, Q.L.: Constructing multiple support vector machines ensemble based on fuzzy integral and rough reducts. The 2nd IEEE Conference on Industrial Electronics and Applications 1256–1259 (2007)

[24] Halrnos, P.R.: Measure theory. New York:Van Nostrand (1950)

[25] Pfeffer, W.F.: Integrals und meusures. New York:Marcel Dekker (1977)

[26] Tahani, H., Keller, J.M.: Information fusion in computer vision using the fuzzy integral. IEEE Transactions on Systems, Man and Cybernetics 20(3) 733–741 (1990)

[27] Shafer, G.A.: A mathematical theory of evidence. Princeton, NJ:Princeton Univ. Press (1976)

[28] Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. int. J. Fiizzy Sets Syst. 1(1) 3–28 (1978)

[29] Sugeno, M.: Fuzzy measures and fuzzy integrals: A survey. Fuzzy Automata and Decision Processes, Amsterdam: North Holland 89–102 (1977)

[30] Terano, T., Sugeno, M.: Conditional fuzzy measures and their application. In Fuzzy Automata and Tlzrir Applications to Cognitive and Decision Processes, New York:Academic Press 151–170 (1975)

[31] Banon, G.: Distinction between several subsets of fuzzy measures. Fuzzy Sets Syst. 5 291–305 (1981)

[32] Gao, W., Cao, B., Shan, S.G., et al.: The cas-peal large-scale chinese face database and baseline evaluations. Technical report of JDL, available on http://www.jdl.ac.cn/~peal/peal tr.pdf (2004)

Emotion Modelling and Facial Affect Recognition in Human-Computer and Human-Robot Interaction

Lori Malatesta[1], John Murray[2], Amaryllis Raouzaiou[1], Antoine Hiolle[2],
Lola Cañamero[2] and Kostas Karpouzis[1]

[1]Image, Video and Multimedia Systems Lab, National Technical University of Athens,
[2]Adaptive Systems Research Group, School of Computer Science, University of
Hertfordshire,
[1]Greece
[2]UK

1. Introduction

As research has revealed the deep role that emotion and emotion expression play in human social interaction, researchers in human-computer interaction have proposed that more effective human-computer interfaces can be realized if the interface models the user's emotion as well as expresses emotions. Affective computing was defined by Rosalind Picard (1997) as computing that relates to, arises from, or deliberately influences emotion or other affective phenomena. According to Picard's pioneering book, if we want computers to be genuinely intelligent and to interact naturally with us, we must give computers the ability to recognize, understand, and even to have and express emotions. These positions have become the foundations of research in the area and have been investigated in great depth after their first postulation.

Emotion is fundamental to human experience, influencing cognition, perception, and everyday tasks such as learning, communication, and even rational decision making. Affective computing aspires to bridge the gap that typical human-computer interaction largely ignored thus creating an often frustrating experience for people, in part because affect had been overlooked or hard to measure.

In order to take these ideas a step further, towards the objectives of practical applications, we need to adapt methods of modelling affect to the requirements of particularshowcases. To do so, it is fundamental to review prevalent psychology theories on emotion, to disambiguate their terminology and identify the fitting computational models that can allow for affective interactions in the desired environments.

1.1 Applications of affective computing

Affective computing deals with the design of systems and devices that can recognize, interpret, generate, and process emotions. We are going to fledge out the potentials this research domain can provide in the field of new media applications and identify the

matching theoretical background that will act as a tool for effectively modelling emotional interaction in such environments.

1.1.1 Detecting and recognizing emotional information

Detecting emotional information usually involves passive sensors, which capture data about the user's physical state or behaviour. The data gathered is often analogous to the cues humans use to perceive emotions in others. For example, a video camera might capture facial expressions, body posture and gestures, while a microphone might capture speech. Other sensors detect emotional cues by directly measuring physiological data such as skin temperature and galvanic resistance.

Recognizing emotional information requires the extraction of meaningful patterns from the gathered data. This is done by parsing the data through various processes such as facial expression detection, gesture recognition, speech recognition, or natural language processing.

1.1.2 Emotion in machines

By the term "emotion in machines" we refer to the simulation of emotions using computers or robots. The goal of such simulation is to enrich and facilitate interactivity between human and machine. The most common and probably most complex application of this simulation lies in the field of conversational agents. A related area is that of affective robot companions, although their expressive and communicative capabilities are usually simpler that those of conversational agents, due to a large extent to other constraints and research challenges related to their embodiment. Such a simulation is closely coupled with emotional understanding and modelling as explained below. This being said, it is important to mention that less sophisticated simulation approaches often produce surprisingly engaging experiences in the area of new media. It is often the case that our aim is not to fully simulate human behaviour and emotional responses but to simply depict emotion in a pseudo intelligent way that makes sense in the specific context of interaction.

1.1.3 Emotional understanding

Emotional understanding refers to the ability of a device not only to detect emotional or affective information, but also to store, process, build and maintain an emotional model of the user. This is not to be mistaken for the related term "emotion understanding", which is used to refer to the use of robotic and computer simulations as tools and models to investigate research hypotheses contributing to the understanding of human and animal emotions (Cañamero, 2005; Cañamero & Gaussier, 2005). The goal of emotional understanding is to understand contextual information about the user and her environment, and formulate an appropriate response. This is difficult because human emotions arise from complex external and internal contexts.

Possible features of a system that displays emotional understanding might be adaptive behaviour, for example, avoiding interaction with a user it perceives to be angry. In the case of affect-aware applications, emotional understanding makes sense in tracking the user's emotional state and adapting environment variables according to the state recognised. Questions regarding the level of detail of the tracking performed, the theoretical grounds for the analysis of the data collected, and the types of potential output that would make sense for such an interactive process are paramount.

2. Affect-related concepts

A lot of confusion exists regarding emotion research terminology, and not without a reason. Different definitions of the role and nature of emotions arise from different scientific approaches since emotion research is typically multidisciplinary. Different disciplines (i.e. psychology, cognitive neuroscience, etc) provide theories and corresponding models that are based on diverse underlying assumptions, are based on different levels of abstraction and may even have different research goals altogether.

The actual definition of *emotions* largely remains an open question: some define it as the physiological changes caused in our body, while the others treat it as purely intellectual thought processes. In psychology research, the term 'affect' is very broad (Rusting, 1998), and has been used to cover a wide variety of experiences such as emotions, moods, and preferences. In contrast, the term 'emotion' tends to be used to refer to fairly brief but intense experiences, although it is also used in a broader sense. Finally, moods or states describe low-intensity but more prolonged experiences.

From a cognitive neuroscience point of view, Damasio (2003) makes a distinction between emotions, which are publicly observable body states, and feelings, which are mental events observable only to the person having them. Based on neuroscience research he and others have done, Damasio argues that an episode of emotion begins with an emotionally "competent" stimulus (such as an attractive person or a scary house) that the organism automatically appraises as conducive to survival or well-being (a good thing) or not conducive (bad). This appraisal takes the form of a complex array of physiological reactions (e.g., quickening heartbeat, tensing facial muscles), which is mapped in the brain. From that map, a feeling arises as "an idea of the body when it is perturbed by the emoting process".

It is apparent that there is no right or wrong approach, and an attempt on a full terminology disambiguation would not be possible without biasing our choices towards one theory over the other. This is to make the point that the context of each approach has to be carefully defined. Next we are going to enumerate core elements of emotion and ways to distinguish them from other affective phenomena. This will lead us to a short description of the directions of affective computing. Subsequently we will put forward the most prevalent psychological theories of emotion along with corresponding computational modelling approaches and couple them to the affective computing goals and more specifically to the goals of practical applications.

2.1 Defining 'emotion' and 'feeling'

Emotion, according to Scherer (1987, 2001), can be defined as an episode of interrelated, synchronized changes in the states of all or most of five organismic subsystems in response to the evaluation of an external or internal stimulus event as relevant to major concerns of the organism. The components of an emotion episode are the particular states of the subsystems mentioned. The process consists of the coordinated changes over time.

Most current psychological theories postulate that

- Subjective experience
- Peripheral physiological response patterns, and
- Motor expression

are major components of emotion. These three components have often been called the *emotional response triad*. Some theorists include the cognitive and motivational domains as components of the emotion process. The elicitation of action tendencies and the preparation

of action have also been implicitly associated with emotional arousal. However, only after explicit inclusion of motivational consequences in theories (and Frijda's forceful claim for the emotion-differentiating function of action tendencies, see (Frijda, 1986)), have these important features of emotion acquired the status of a major component. The inclusion of a cognitive information-processing component has met with less consensus. Many theorists still prefer to see emotion and cognition as two independent but interacting systems. However, one can argue that all subsystems underlying emotion components function independently much of the time, and that the special nature of emotion as a hypothetical construct consists of the coordination and synchronization of all these systems during an emotion episode (Scherer, 2004).

How can emotions, as defined above, be distinguished from other affective phenomena such as feelings, moods, or attitudes? Let us take the term 'feeling' first. Scherer aligns feeling with the "subjective emotional experience" component of emotion, thus reflecting the total pattern of cognitive appraisal as well as motivational and somatic response patterning that underlie the subjective experience of an emotion. If we use the term 'feeling', a single component denoting subjective experience process, as a synonym for 'emotion' (the total multi-modal component process), this is likely to produce serious confusion and hamper our understanding of the phenomenon.

If we accept feeling as one of emotions' components, then the next step is to differentiate emotion from other types of affective phenomena. Instances of these phenomena, which can vary in degree of affectivity, are often called "emotions" in the literature. There are five such types of affective phenomena that should be distinguished from emotion: *preferences*, *attitudes*, *moods*, *affective dispositions* and *interpersonal stances*.

2.2 Emotions in applied intelligence

Having distinguished emotions against other types of affective phenomena, it is now of particular interest, in regard to the new media domain, to present a suggested distinction on a different level. Scherer (2004) questioned the need to distinguish between two different types of emotion: (1) *aesthetic* emotions (2) *utilitarian* emotions. The latter correspond to the "garden variety" of emotions usually studied in emotion research, such as anger, fear, joy, disgust, sadness, shame, guilt. These types of emotions can be considered utilitarian in the sense of facilitating our adaptation to events that have important consequences for our well-being. Such adaptive functions are the preparation of action tendencies (fight, flight), recovery and reorientation (grief, work), motivational enhancement (joy, pride), or the creation of social obligations (reparation). Because of their importance for survival and well-being, many utilitarian emotions are high-intensity emergency reactions, involving the synchronization of many subsystems, as described earlier. In the case of aesthetic emotions, adaptation to an event that requires the appraisal of goal relevance and coping potential is absent, or much less pronounced. Kant defined aesthetic experience as "disinterested pleasure" (Kant, 1790), highlighting the complete absence of utilitarian considerations. Thus, *my* aesthetic experience of a work of art or a piece of music is not shaped by the appraisal of the work's ability to satisfy *my* bodily needs, further *my* current goals or plans, or correspond to *my* social values. Rather, aesthetic emotions are produced by the appreciation of the intrinsic qualities of a work of art or an artistic performance, or the beauty of nature. Examples of such aesthetic emotions are: being moved or awed, full of wonder, admiration, bliss, ecstasy, fascination, harmony, rapture, solemnity.

This differentiation of emotions has an impact on the way an appraisal-based modelling approach would be implemented. It would not make sense to try and model all the proposed components of an appraisal process in cases where only aesthetic emotions are expected. On the other hand it would make sense to provide a deeper model in cases where anger or frustration are common emotional states such as in the example of an interactive Television environment.

2.3 Emotion representation

When it comes to machine-based recognition of emotions, one of the key issues is the selection of appropriate ways to represent the user's emotional states. The most familiar and commonly used way of describing emotions is by using categorical labels, many of which are either drawn directly from everyday language, or adapted from it. This trend may be due to the great influence of the works of Ekman and Friesen who proposed that the archetypal emotions correspond to distinct facial expressions which are supposed to be universally recognizable across cultures (Ekman, 1978, 1993).

On the contrary, psychology researchers have extensively investigated a broader variety of emotions. An extensive survey on emotion analysis can be found in (Cowie, 2001). The main problem with this approach is deciding which words qualify as genuinely emotional. There is, however, general agreement as to the large scale of the emotional lexicon, with most lists of descriptive terms numbering into the hundreds; the Semantic Atlas of Emotional Concepts (Averill, 1975) lists 558 words with 'emotional connotations'. Of course, it is difficult to imagine an artificial systems being able to match the level of discrimination that is implied by the length of this list.

Although the labeling approach to emotion representation fits perfectly in some contexts and has thus been studied and used extensively in the literature, there are other cases in which a continuous, rather than discrete, approach to emotion representation is more suitable. At the opposite extreme from the list of categories are dimensional descriptions, which identify emotional states by associating them with points in a multidimensional space. The approach has a long history, dating from Wundt's (1903) original proposal to Schlossberg's reintroduction of the idea in the modern era (Schlosberg, 1954). For example, activation-emotion space as a representation has great appeal as it is both simple, while at the same time makes it possible to capture a wide range of significant issues in emotion (Cowie, 2001). The concept is based on a simplified treatment of two key themes:

- Valence: The clearest common element of emotional states is that the person is materially influenced by feelings that are valenced, i.e., they are centrally concerned with positive or negative evaluations of people or things or events.

- Activation level: Research from Darwin onwards has recognized that emotional states involve dispositions to act in certain ways. A basic way of reflecting that theme turns out to be surprisingly useful. States are simply rated in terms of the associated activation level, i.e., the strength of the person's disposition to take some action rather than none.

There is general agreement on these two main dimensions. Still, in addition to these two, there are a number of other possible dimensions, such as power-control, or approach-avoidance. Dimensional representations are attractive mainly because they provide a way of describing emotional states that is more tractable than using words. This is of particular importance when dealing with naturalistic data, where a wide range of emotional states occur. Similarly, they are much more able to deal with non discrete emotions and variations in emotional state over time.
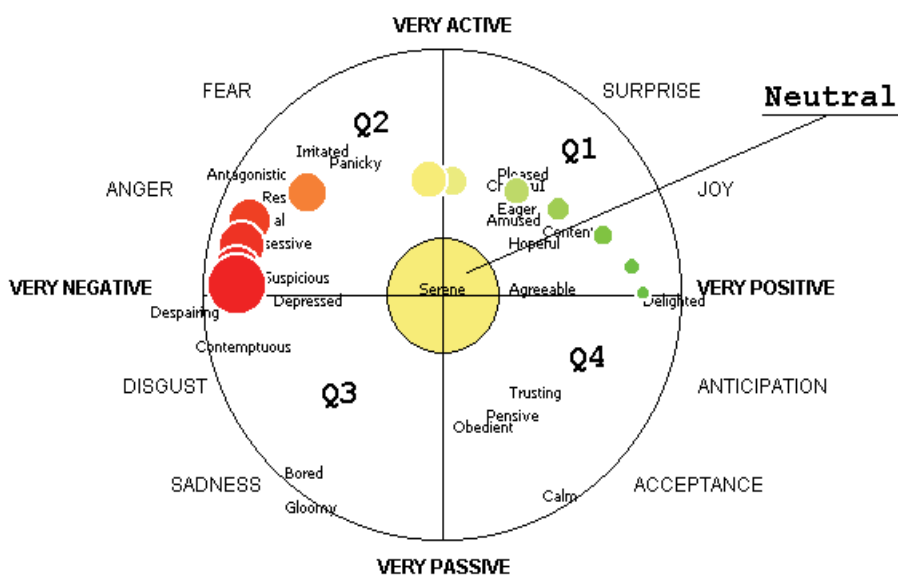
Fig. 1. The activation/valence dimensional representation, after (Whissel, 1989)

As we can see in Figure 1, labels are typically given for emotions falling in areas where at least one of the two axes has a value considerably different than zero. On the other hand, the beginning of the axes (the center of the diagram) is typically considered as the neutral emotion. For the same reasons mentioned above, we find it is not meaningful to define the neutral state so strictly. Therefore, we have added to the more conventional areas corresponding to the four quadrants a fifth one, corresponding to the neutral area of the diagram, as is depicted in Figure 1.

Label	Location in FeelTrace (Cowie, 2000) diagram
Q1	positive activation, positive evaluation (+/+)
Q2	positive activation, negative evaluation (+/-)
Q3	negative activation, negative evaluation (-/-)
Q4	negative activation, positive evaluation (-/+)
Neutral	close to the center

Table 1. Emotion classes

2.4 Other emotion representation models
Having reviewed the areas of affective computing, it is time to start focusing on the available theories, descriptions and models that can support these goals. We start with reviewing the main groups of emotion descriptions as identified by the members of the Humaine Network of Excellence (Humaine, 2008). It is important to stress the difference that exists between emotion models and emotion descriptions. By 'emotion descriptions' we refer to different ways of representing emotions and their underlying psychological theories, whereas with the term 'emotional models' we talk about the computational modelling of these theories in specific context.

2.4.1 Categorical representations

Categorical representations — using a word to describe an emotional state — are the simplest and most widespread. Such category sets have been proposed on different grounds, including evolutionarily basic emotion categories; most frequent everyday emotions; application-specific emotion sets; or categories describing other affective states, such as moods or interpersonal stances (Feeltrace core vocabulary in Cowie, 1999; Ortony, Clore and Collins list of emotion words in Ortony, 1988; Ekman's list of six basic emotions in Ekman, 1993).

2.4.2 Other dimensional descriptions

Dimensional descriptions capture essential properties of emotional states, such as arousal (active/passive) and valence (negative/positive). Emotion dimensions can be used to describe general emotional tendencies, including low-intensity emotions. In addition to these two, there are a number of other possible dimensions, such as power, control, or approach/avoidance, which add some refinement. The most obvious is the ability to distinguish between fear and anger, both of which involve negative valence and high activation. In anger, the subject of the emotion feels that he or she is in control; in fear, control is felt to lie elsewhere.

Dimensional representations are attractive mainly because they provide a way of describing emotional states that is more tractable than using words. This is of particular importance when dealing with naturalistic data, where a wide range of emotional states occur. Similarly, they are much more able to deal with non-discrete emotions and variations in emotional state over time. A further attraction is the fact that dimensional descriptions can be translated into and out of verbal descriptions. This is possible because emotion words can, to an extent, be understood as referring to positions in activation-evaluation space.

2.4.3 Appraisal theories and representations

Appraisal theories focus on the emotion elicitation process in contrast with the previously mentioned approaches that emphasize on the consequences/symptoms of an emotional episode. Appraisal representations characterise emotional states in terms of the detailed evaluations of eliciting conditions, such as their familiarity, intrinsic pleasantness, or relevance to one's goals. Such detail can be used to characterise the cause or object of an emotion as it arises from the context, or to predict emotions in AI systems (Lazarus, 1991, Scherer, 1987, Frijda, 1986).

Appraisal theories are very common in emotion modelling since their structure caters for simulating their postulations in computational models. Moreover, it is often the case that an appraisal theory was formulated explicitly in order to be implemented in a computer. Such an example is the OCC theory (Ortony, 1988). This is sometimes a source of confusion, since the underlying emotion theory is unavoidably very closely linked with the actual modelling approach.

According to cognitive theories of emotion (Lazarus, 1987), emotions are closely related to the situation that is being experienced (or, indeed, imagined) by the agent.

3. Facial expression recognition

3.1 Feature representation

Automatic estimation of facial model parameters is a difficult problem and although a lot of work has been done on selection and tracking of features (Tomasi, 1991), relatively little

work has been reported (Tian, 2001) on the necessary initialization step of tracking algorithms, which is required in the context of facial feature extraction and expression recognition. Most facial expression recognition systems use the Facial Action Coding System (FACS) model introduced by Ekman and Friesen (Ekman, 1978) for describing facial expressions. FACS describes expressions using 66 Action Units (AU) which relate to the contractions of specific facial muscles.

Additionally to FACS, MPEG-4 metrics (Tekalp, 2000) are commonly used to model facial expressions and underlying emotions. They define an alternative way of modelling facial expressions and the underlying emotions, which is strongly influenced by neurophysiologic and psychological studies. MPEG-4, mainly focusing on facial expression synthesis and animation, defines the Facial Animation Parameters (FAPs) that are strongly related to the Action Units (AUs), the core of the FACS.

Most existing approaches in facial feature extraction are either designed to cope with limited diversity of video characteristics or require manual initialization or intervention. Specifically (Tian, 2001) depends on optical flow, (Leung, 2004) depends on high resolution or noise-free input video, (Sebe, 2004) depends on colour information, (Cootes, 2001) requires two head-mounted cameras and (Pantic, 2000) requires manual selection of feature points on the first frame. Additionally, very few approaches can perform in near-real time. In this work we combine a variety of feature detection methodologies in order to produce a robust FAP estimator, as outlined in the following.

3.2 Facial feature extraction

Facial feature extraction is a crucial step to numerous applications such us face recognition, human-computer interaction, facial expression recognition, surveillance and gaze/pose detection (Asteriadis, 2007). In their vast majority, the approaches in the bibliography use face detection as a pre-processing step. This is usually necessary in order to tackle with scale problems, as localizing a face in an image is more scale-independent than starting with the localization of special facial features. When only facial features are detected (starting from the whole image and not from the face region of interest), the size and the position of the face in the image have to be pre-determined and, thus, such algorithms are devoted to special cases, such as driver's attention recognition (Smith, 2003) where the user's position with regards to a camera is almost stable. In such techniques, colour (Smith, 2003) predicates, shape of facial features and their geometrical relations (D' Orazio, 2004) are used as criteria for the extraction of facial characteristics.

On the other side, facial features detection is more scale-independent when the face is detected as a pre-processing step. In this case, the face region of interest can be normalized to certain dimensions, thus making the task of facial feature detection more robust. For example, in (Cristinacce, 2004) a multi-stage approach is used to locate features on a face. First, the face is detected using the boosted cascaded classifier algorithm by Viola and Jones (Viola, 2001). The same classifier is trained using facial feature patches to detect facial features. A novel shape constraint, the Pairwise Reinforcement of Feature Responses (PRFR) is used to improve the localization accuracy of the detected features. In (Jerosky, 2001), a three-stage technique is used for eye centre localization. The Hausdorff distance between edges of the image and an edge model of the face is used to detect the face area. At the second stage, the Hausdorff distance between the image edges and a more refined model of the area around the eyes is used for more accurate localization of the upper area of the head.

Finally, a Multi-Layer Perceptron (MLP) is used for finding the exact pupil locations. In (Viola, 2001), an SVM-based approach is used for face detection. Following, eye-areas are located using a feed-forward neural network and the face is brought to a horizontal position based on the eye positions. Starting from these points, edge information and luminance values, are used for eyebrow and nostrils detection. Further masks are created to refine the eye positions, based on edge, luminance and morphological operations. Similar approaches are followed for the detection of mouth points.

In this work, prior to eye and mouth region detection, face detection is applied on the face images. The face is detected using the Boosted Cascade method, described in (Viola, 2001). The output of this method is usually the face region with some background. Furthermore, the position of the face is often not centred in the detected sub-image. Since the detection of the eyes and mouth will be done on blocks of a predefined size, it is very important to have an accurate face detection algorithm. Consequently, a technique to post-process the results of the face detector is used.

More specifically, a technique that compares the shape of a face with that of an ellipse is used (Asteriadis, 2007). According to this, the distance map of the face area found at the first step is extracted. Here, the distance map is calculated from the binary edge map of the area. An ellipsis scans the distance map and a score that is the average of all distance map values on the ellipse contour el, is evaluated.

$$score = \frac{1}{el} \sum_{(x,y) \in el} D(x, y) \tag{1}$$

where D is the distance map of the region found by the Boosted Cascade algorithm. This score is calculated for various scale and shape transformations of the ellipses. The transformation which gives the best score is considered as the one that corresponds to the ellipses that best describes the exact face contour. The lateral boundaries of the ellipses are the new boundaries of the face region.

A template matching technique follows for the facial feature area detection step: The face region found by the face detection step is brought to certain dimensions and the corresponding Canny edge map is extracted. Subsequently, for each pixel on the edge map, a vector pointing to the closest edge is calculated and its x, y coordinates are stored. The final result is a vector field encoding the geometry of the face. Prototype eye patches were used for the calculation of their corresponding vector fields and the mean vector field was used as prototype for searching similar vector fields on areas of specified dimensions on the face vector field. The similarity between an image region and the templates is based on the following distance measure:

$$E_{L_2} = \sum_{i \in R_k} \| v_i - m_i \| \tag{2}$$

where $\|\|$ denotes the L_2 norm. Essentially for a NxM region R_k the previous formula is the sum of the Euclidean distances between vectors v_i of the candidate region and the corresponding mi of the mean vector field (template) of the eye we are searching for (right or left). The candidate region on the face that minimizes E_{L2} is marked as the region of the left or right eye. To make the algorithm faster we utilize the knowledge of the approximate positions of eyes on a face.

For eye centre detection, the normalized area of the eye is brought back to its initial dimensions on the image and a light reflection removal step is employed. The grayscale image of the eye area is converted to a binary image and small white connected components are removed. The areas that correspond to such components on the original image are substituted by the average of their surrounding area. The final result is an eye area having reflections removed. Subsequently, horizontal and vertical derivative maps are extracted from the resulting image and they are projected on the vertical and horizontal axis respectively. The mean of a set of the largest projections is used for an estimate of the eye centre. Following, a small window around the detected point is used for the darkest patch to be detected, and its centre is considered as the refined position of the eye centre.

For the detection of the eye corners (left, right, upper and lower) a technique similar to that described in (Ioannou, 2007) is used: Having found the eye centre, a small area around it is used for the rest of the points to be detected. This is done by using the Generalized Projection Functions (GPFs), which are a combination of the Integral Projection Functions (IPFs) and the Variance Projection Functions (VPFs). The integral projection function's value on row (column) x (y) is the mean of its luminance intensity, while the Variance Projection Function on row x is its mean variance. The GPF's value on a row (column) x (y) is a linear combination of the corresponding values of the derivatives of the IPF and VPF on row x (column y):

$$GPF_u(x) = (1-a) * IPF_u^{'}(x) + a * VPF_u^{'}$$

$$GPF_v(y) = (1-a) * IPF_v^{'}(y) + a * VPF_v^{'}$$

(3)

Local maxima of the above functions are used to declare the positions of the eye boundaries.

For the mouth area localization, a similar approach to that of the eye area localization is used: The vector field of the face is used and template images are used for the extraction of a prototype vector field of the mouth area. Subsequently, similar vector fields are searched for on the lower part of the normalized face image. However, as the mouth has, many times,
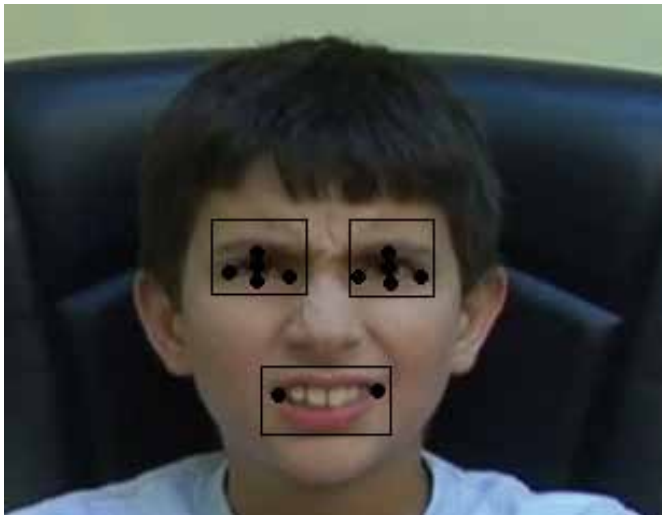

Fig. 2. Detected facial features

similar luminance values with its surrounding skin, an extra factor is also taken into account. That is, at every search area, the mean value of the hue component is calculated and added to the inverse distance from the mean vector fields of the mouth. Minimum values declare mouth existence.

For the extraction of the mouth points of interest (mouth corners), the hue component is also used. Based on the hue values of the mouth, the detected mouth area is binarised and small connected components whose value is close to 0º are discarded similar to the light reflection removal technique employed for the eyes. The remainder is the largest connected component, which is considered as the mouth area. The leftmost and rightmost points of this area are considered as the mouth corners. An example of detected feature points is shown in Figure 2.

4. Expression classification

4.1 Recognizing dynamics

In order to consider the dynamics of displayed expressions, one needs to utilize a classification model that is able to model and learn dynamics, such as a Hidden Markov Model or a recurrent neural network (see Figure 3). This type of network differs from conventional feed-forward networks in that the first layer has a recurrent connection. The delay in this connection stores values from the previous time step, which can be used in the current time step, thus providing the element of memory.
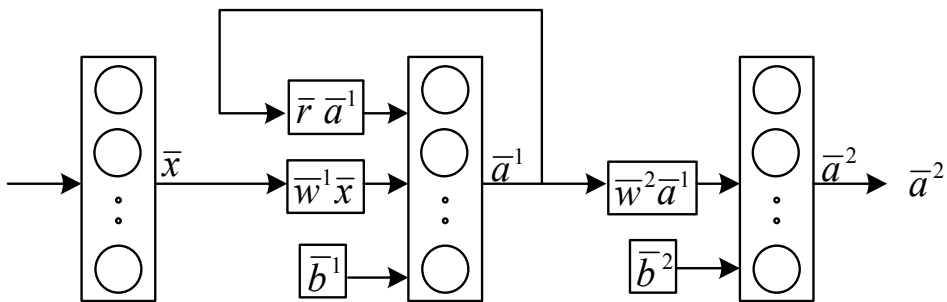

Fig. 3. A recurrent neural network (Elman, 1990, 1991)

Among other implementations of recurrent networks, the Elman net (Elman, 1990, 1991) is the most popular. This is a two-layer network with feedback from the first layer output to the first layer input. This recurrent connection allows the Elman network to both detect and generate time-varying patterns.

The transfer functions of the neurons used in the Elman net are tan-sigmoid for the hidden (recurrent) layer and purely linear for the output layer. More formally

$$a_i^1 = \tan sig(k_i^1) = \frac{2}{1+e^{-2k_i^1}} - 1 , \; a_j^2 = k_j^2 \tag{4}$$

where a_i^1 is the activation of the i-th neuron in the first (hidden) layer, k_i^1 is the induced local field or activation potential of the i-th neuron in the first layer, a_j^2 is the activation of

the j-th neuron in the second (output) layer and k_j^2 is the induced local field or activation potential of the j-th neuron in the second layer.

The induced local field in the first layer is computed as:

$$k_i^1 = \overline{w}_i^1 \cdot \overline{x} + \overline{r}_i \cdot \overline{a}^1 + b_i^1 \tag{5}$$

where $\overline{x}$ is the input vector, $\overline{w}_i^1$ is the input weight vector for the i-th neuron, $\overline{a}^1$ is the first layer's output vector for the previous time step, $\overline{r}_i$ is the recurrent weight vector and b_i^1 is the bias. The local field in the second layer is computed in the conventional way as:

$$k_j^2 = \overline{w}_j^2 \cdot \overline{a}^1 + b_j^2 \tag{6}$$

where $\overline{w}_i^2$ is the input weight and b_j^2 is the bias.

This combination of activation functions is special in that two-layer networks with these transfer functions can approximate any function (with a finite number of discontinuities) with arbitrary accuracy. The only requirement is that the hidden layer must have enough neurons (Schaefer, 1996, Hammer, 2003).

As far as training is concerned, the truncated back-propagation through time (truncated BPTT) algorithm is used (Haykin, 1999).

The input layer of the utilized network has 57 neurons (25 for the FAPs and 32 for the audio features). The hidden layer has 20 neurons and the output layer has 5 neurons, one for each one of five possible classes: *Neutral*, *Q1* (first quadrant of the Feeltrace plane), *Q2*, *Q3* and *Q4*. The network is trained to produce a level of 1 at the output that corresponds to the quadrant of the examined tune (Cowie, 2001) and levels of 0 at the other outputs.

4.2 Classification

The most common applications of recurrent neural networks include complex tasks such as modelling, approximating, generating and predicting dynamic sequences of known or unknown statistical characteristics. In contrast to simpler neural network structures, using them for the seemingly easier task of input classification is not equally simple or straight-forward.

The reason is that where simple neural networks provide one response in the form of a value or vector of values at their output after considering a given input, recurrent neural networks provide such inputs after each different time step. So, one question to answer is at which time step the network's output should be read for the best classification decision to be reached.

As a general rule of thumb, the very first outputs of a recurrent neural network are not very reliable. The reason is that a recurrent neural network is typically trained to pick up the dynamics that exist in sequential data and therefore needs to see an adequate length of the data in order to be able to detect and classify these dynamics. On the other hand, it is not always safe to utilize the output of the very last time step as the classification result of the network because:

1. the duration of the input data may be a few time steps longer than the duration of the dominating dynamic behaviour and thus the operation of the network during the last time steps may be random
2. a temporary error may occur at any time step of the operation of the network

For example, in Figure 4 we present the output levels of the network after each frame when processing the tune of the running example. We can see that during the first frames the output of the network is quite random and changes swiftly. When enough length of the sequence has been seen by the network so that the dynamics can be picked up, the outputs start to converge to their final values. But even then small changes to the output levels can be observed between consecutive frames.
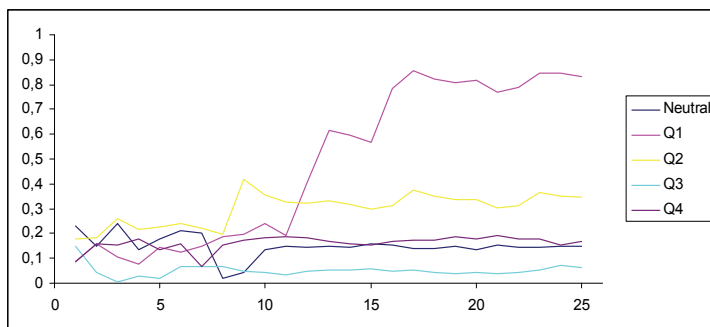

Fig. 4. Individual network outputs after each frame (Caridakis, 2006)
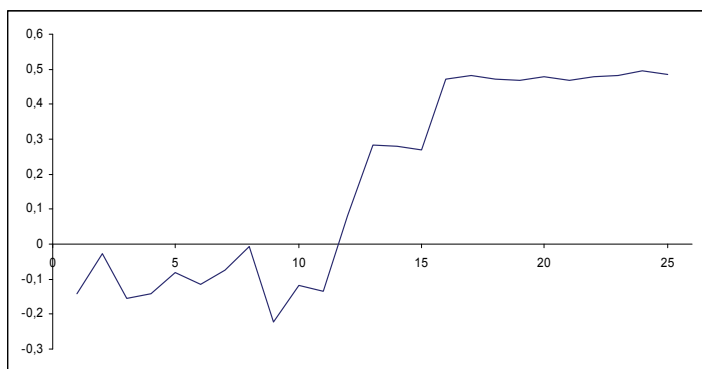

Fig. 5. Margin between correct and next best output

Although these are not enough to change the classification decision (see Figure 5) for this example where the classification to Q1 is clear, there are cases in which the classification margin is smaller and these changes also lead to temporary classification decision changes. In order to arm our classification model with robustness we have added a weighting integrating module to the output of the neural network which increases its stability. Specifically, the final outputs of the model are computed as:

$$o_j(t) = c \cdot a_j^2 + (1-c) \cdot o_j(t-1) \tag{7}$$

where $o_j(t)$ is the value computed for the j-th output after time step t, $o_j(t-1)$ is the output value computed at the previous time step and c is a parameter taken from the (0,1] range that controls the sensitivity/stability of the classification model. When c is closer to zero the model becomes very stable and a large sequence of changed values of k_j^2 is required to affect the classification results while as c approaches one the model becomes more sensitive to changes in the output of the network. When $c = 1$ the integrating module

is disabled and the network output is acquired as overall classification result. In our work, after observing the models performance for different values of c, we have chosen $c = 0.5$.
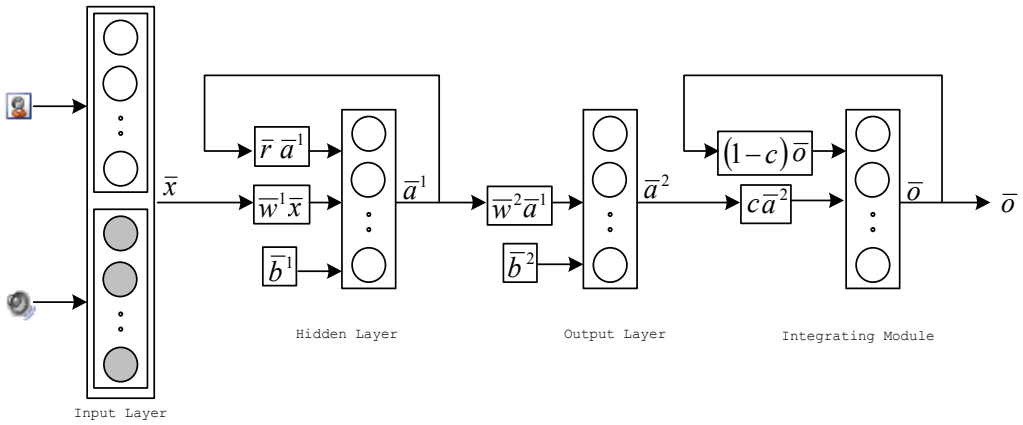

Fig. 6. The Elman net with the output integrator (Cowie, 2008)

In Figure 5, we can see the decision margin when using the weighting integration module at the output of the network. When comparing to Figure 7, we can clearly see that the progress of the margin is smoother, which indicates that we have indeed succeeded in making the classification performance of the network more stable and less dependent on frame that is chosen as the end of a tune.
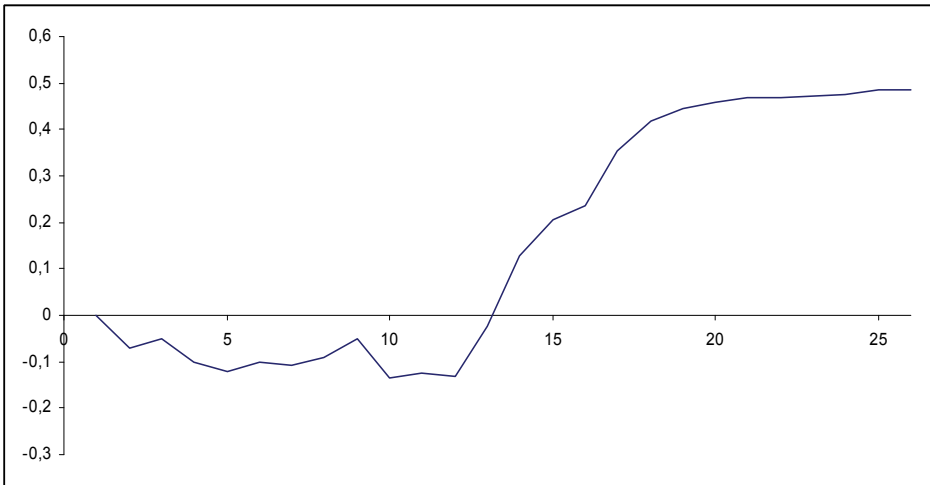

Fig. 7. Decision margin when using the integrator

Of course, in order for this weighted integrator to operate, we need to define output values for the network for time step 0, i.e. before the first frame. It is easy to see that due to the way that the effect of previous outputs wares off as time steps elapse due to c, this initialization is practically indifferent for tunes of adequate length. On the other hand, this value may have an important affect on tunes that are very short. In this work, we have chosen to initialize all initial outputs at

$$\bar{o}(0) = 0 \qquad (8)$$

Another meaningful alternative would be to initialize $\bar{o}(0)$ based on the percentages of the different output classes in the ground truth data used to train the classifier. We have avoided doing this in order not to add a bias towards any of the outputs, as we wanted to be sure that the performance acquired during testing is due solely to the dynamic and multimodal approach proposed in this work.

It is worth noting that, from a modelling point of view, it was feasible to include this integrator in the structure of the network rather than having it as an external module, simply by adding a recurrent loop at the output layer as well. We have decided to avoid doing so, in order not to also affect the training behaviour of the network, as an additional recurrent loop would greatly augment the training time and size and average length of training data required.

5. Application in human-robot interaction

The question of how autonomous robots could be part of our everyday life is of a growing interest. Technology is now at a stage where in the very near future it will be possible for the majority of households to have their own robots, helping with a variety of tasks and even entertaining their owner. However, the problem of deciding on what kind of architectures are going to be embedded in such robots is still not completely answered. Indeed, these architectures require a set of properties that are not yet found in the currently available ones. The robots have to be adaptive in the complex and dynamic environment that we live in. To design such an ideal robot, it is argued that taking an epigenetic approach would be a suited solution (Cañamero et al., 2006).

5.1 Attachment bonds and emotional development

Taking inspiration from psychology, modelling the development of attachment bonds between a robot and its owner or user could offer a promising avenue to improve human-robot interactions. These phenomena would help a robot initiate interactions with the humans in more natural way, without any explicit teaching signal. The interactions and the robot's behaviour would be modulated using the feedback from the current emotional state of the user(s). To that end, an autonomous robot would need to use information from the user's current emotional state, extracting this information from the facial expressions.

A simple application of these idea could be to start to improve already existing models of the imprinting phenomenon. Imprinting was documented by Konrad Lorenz (Lorenz, 1935) and described as the tendency of young birds to follow the first moving object or person seen after hatching. This process is a product of evolution helping the survival of these species which don't have a nest to protect and hide their offspring. A successful architecture has been developed to allow a Koala robot to discover what object or person to follow, and then learn how to follow him/her (Hiolle & Cañamero, 2007) . The robot used the distance between it and the human as a "desired perception" to keep constant, as would be the sight of the mother for a young bird, and then learned how to maintain it using a low-level sensorimotor learning, without any prior knowledge about how to do so. Using facial detection and facial affect recognition would enhance this model since the feedback from the human emotional state could be used as reinforcer for the robot, to learn how and when to follow the human being.

Moreover, applying this kind of architecture to an expressive robot caters for not only developing a following behaviour, but also for allowing it to discover how to respond to the user's facial expression, according to the context (task being handled, number of human present).

From that point, using theories from developmental psychology (Sroufe, 1995), it would be possible for the robot to develop its own set of emotional states, all branching from one precursor of emotion. To clarify, it is believed that infants are born endowed with one proto-emotion, the excitement, which is modulated by fluctuation of endogenous states of the central nervous system and later by external stimulations. During the infant development in the first year of life, these fluctuations of excitement are then correlated with external stimuli, to later build categories of emotional states according to the context. For instance, anger is believed to derive from an increase of the excitement provoked by the inability for the infant to carry out an action because something or someone is preventing it from happening. Giving the possibility for the robot to build these context, and then, the categories of fluctuation of its inner excitement, given the context which has to take into account the other agent (human user or another robot) emotional state, would help the robot developing its own representation from the environment and coping strategies in the various situations it has encountered. This way, each robot would be adapted to the environment they are embedded in, since their development would be a product of their own experiences, and not a set of fixed rules already built-in their architecture.

5.2 Emotional robotic feedback via facial expression

In human-robot interaction, it is of paramount importance that the robot is capable of determining the emotional state of the human user, or on a lesser scale at least able to determine the emotional expression of the user in terms of simple emotional states such as 'stressed', 'relaxed', 'frustrated', or basic emotions such as 'happy', 'sad', 'angry' or 'surprised'. However, there is also a strong case for the reverse of this. That is, allowing the robot to exhibit emotional expressions that relate to some internal state of the robot. The purpose for this would be to allow the human interacting with the robot to assess how their interaction is being handled by the robot in a manner that s/he is familiar with. In order to bridge the gap between humans interacting with robots and in turn robots interacting with humans, the process needs to be as natural as possible without the need for a priori information.

Therefore, we have developed a robot head ERWIN (Emotional Robot with Intelligent Networks) that incorporates several interactive modalities that allows for HRI. ERWIN is designed to be capable of tracking and recognising human faces. The method used for initial detection of a face is based on Viola's (2001) rapid object detection with improvements made by Lienhart and Maydt (2002). In order to be able to detect a face, or any desired object of interest, specially trained classifiers are used. These classifiers are trained as described in (Kuranov *et al.*, 2002) using a sample set of images that contain the particular feature we wish to detect; in our case this was a collection of face images.

Once a set of images has been collated, a separate text file is created which provides the coordinates of a bounding box encapsulating the object within the specified image. These images and the accompanying text file are called the positive training set as they are used to build a classifier that will be used to detect the desired object. However, in order to reduce the chances of the classifier falsely identifying an object during runtime, the classifiers are

also presented with what is termed negative training images. These consist of random images of various objects such as cars, trees, and buildings. An important feature of the negative training images is that they must *not* contain a positive image, i.e. the object we wish to detect – a face.

Sound source localisation is another modality that is built into ERWIN to allow for tracking of sound sources of interest within a cluttered environment. In addition to remaining focused on one user allowing for the two-way communication process to be as natural as possible. Thus, with the addition of emotional expressions being possible with robots it is possible for the human to judge how their interaction with the robot is being taken. For example, it would be possible to allow a robot to greet a user with a look of surprise, shock or happiness when it has not seen that user for a long time. Or if interaction with a human isn't going well, the robot is unable to clearly recognise the speech of the user, then the displayed emotion could reflect anger.

ERWIN also has the ability to display simple, but effective emotions by controlling several actuators attached to various parts of the face. ERWIN can move four separate actuators, each controlling a separate feature; these include the left and right eyebrows, and the upper and lower lips. By controlling these features based on responses and interaction from the external influences,s ERWIN can display a range of emotions that in turn can affect the response from a human thus bringing such HRI closer to human-human interactions. The basic emotions include happy, sad, angry, shocked and surprised. For instance, once ERWIN has been called or recognises a familiar face it can respond with generating a happy face. This gives excellent scope for combining the multiple modalities described, allowing the emotions to change if ERWIN detects sound but does not locate a face or changing emotion if ERWIN cannot understand or interpret what a human is saying. This may later provide an opportunity to develop the emotion response using internal states as modelled in artificial immune systems (Neal, 2002) and endocrine systems (Cañamero, 1997; Neal and Timmis, 2003; Avila-García and Cañamero, 2004; Cañamero and Avila-García, 2007), which allows internal states to influence responses to the external acoustic and visual information gathered.

6. Acknowledgment

7. References

Asteriadis, S., Tzouveli, P., Karpouzis, K., Kollias, S. (2008) Estimation of behavioral user state based on eye gaze and head pose - application in an e-learning environment, *Multimedia Tools and Applications*, accepted for publication.

Averill, J.R. (1975). A semantic atlas of emotional concepts. *JSAS Catalogue of Selected Documents in Psychology*: 5, 330.

Avila-García, O. and Cañamero, L. (2004). Using Hormonal Feedback to Modulate Action Selection in a Competitive Scenario. In *Proc. Eight Intl. Conf. Simulation of Adaptive Behavior (SAB04)*, pp. 243–252. MIT Press, Cambridge, MA.

Cañamero, L.D. (1997). Modeling Motivations and Emotions as a Basis for Intelligent Behavior. In Johnson, W.L. (ed.) Proc. First Intl. Conf. on Autonomous Agents, pp. 148–155. ACM Press, New York.

Cañamero, L. (2005). Emotion Understanding from the Perspective of Autonomous Robots Research. *Neural Networks*, 18: 445-455.

Cañamero, L. and Avila-García, O. (2007). A Bottom-Up Investigation of Emotional Modulation in Competitive Scenarios. In A. Paiva, R. Prada, and R.W. Picard (Eds.), *Proc. Second International Conference on Affective Computing and Intelligent Interaction (ACII 2007),* LNCS 4738, pp. 398–409. Berlin & Heidelberg: Springer-Verlag.

Cañamero, L., Blanchard, A., Nadel, J. (2006), Attachment bonds for human-like robots *International Journal of Humanoid Robotics*, 3(3), 301–320.

Cañamero, L. and Gaussier, P. (2005). Emotion Understanding: Robots as Tools and Models. In J. Nadel and D. Muir (Eds.), *Emotional Development: Recent research advances*, pp. 235-258. Oxford University Press.

Caridakis, G., Malatesta, L., Kessous, L., Amir, N., Raouzaiou, A., Karpouzis, K. (2006) Modeling naturalistic affective states via facial and vocal expressions recognition, International Conference on Multimodal Interfaces (ICMI'06), Banff, Alberta, Canada, November 2-4, 2006.

Cootes, T., Edwards, G., Taylor, C. (2001) Active appearance models, IEEE Trans PAMI 23 (6), pp. 681-685.

Cowie, R., Douglas-Cowie, E., Apolloni, B., Taylor, J., Romano, A., & Fellenz, W. (1999). What a neural net needs to know about emotion words. In N. Mastorakis (Ed.), *Computational intelligence and applications,* pp. 109-114. World Scientific Engineering Society.

Cowie, R., Douglas-Cowie, E., Savvidou, S., McMahon, E., Sawey, M., Schröder, M. (2000) 'Feeltrace': An instrument for recording perceived emotion in real time, in *Proc. ISCA Workshop on Speech and Emotion*, pp. 19–24.

Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., Taylor, J. (2001) Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine*: 18 (1), pp. 32–80.

Cowie, R., Douglas-Cowie, Karpouzis, K., Caridakis, G., Wallace, M., Kollias, S. (2008) Recognition of Emotional States in Natural Human-Computer Interaction, in D. Tzovaras (ed.), *Multimodal User Interfaces - From Signals to Interaction*, pp. 119-153, Springer Berlin Heidelberg.

Cristinacce, D., Cootes, T., Scott, I. (2004) A multi-stage approach to facial feature detection, *15th British Machine Vision Conference*, pp. 231-240.

D' Orazio, T., Leo, M., Cicirelli, G., Distante, A. (2004) An algorithm for real time eye detection in face images, *ICPR*, Vol. 3, 278-281.

Damasio, A. (2003) *Looking for Spinoza: Joy, Sorrow, and the Feeling Brain*, Harcourt, Orlando, FL, USA.

Ekman P., Friesen, W.V. (1978) *The Facial Action Coding System: A Technique for the Measurement of Facial Movement*, Consulting Psychologists Press, San Francisco, USA.

Ekman, P. (1993) Facial expression and Emotion. *Am. Psychologist*, Vol. 48, pp. 384-392.

Elman, J.L. (1990) Finding structure in time, *Cognitive Science*, 14, pp. 179-211.

Elman, J.L. (1991) Distributed representations, simple recurrent networks, and grammatical structure, *Machine Learning*, 7, 195-224, 1991.

Frijda, N.H. (1986). *The Emotions: Studies in Emotion and Social Interaction*, Cambridge University Press, New York, USA.

Goldie, P. (2004) *On Personality*. Rutledge, New York, USA.

Hammer, B., Tino, P. (2003) Recurrent neural networks with small weights implement definite memory machines, *Neural Computation* 15(8), pp. 1897-1929.

Haykin, S. (1999) *Neural Networks: A Comprehensive Foundation*, Prentice Hall International.

Hiolle, A., Cañamero, L., and Blanchard, A. (2007), Learning to interact with the caretaker: A developmental approach. In Paiva, A., Prada, R., and Picard, R., (Eds.), Proc. of the 2nd Intl. Conf. on Affective Computing and Intelligent Interactions, pages 422–433. Berlin and Heidelberg: Springer-Verlag

Humaine FP6 Network, of Excellence, http://emotion-research.net, Retrieved on Jan. 28, 2008.

Ioannou, S., Caridakis, G., Karpouzis, K., Kollias, S. (2007) Robust Feature Detection for Facial Expression Recognition, *EURASIP Journal on Image and Video Processing*, doi:10.1155/2007/29081.

Jesorsky, O., Kirchberg, K.J., Frischholz, R.W. (2001) Robust face detection using the Hausdorff distance, *3rd Conf. AVBPA*, pp. 90-95.

Kant, I. (1790) *Critique of Judgment*, Trans. Werner S. Pluhar, Hackett Publishing, Indianapolis, USA, 1987.

Kuranov, A., Lienhart R., and Pisarevsky V. (2002). An Empirical Analysis of Boosting Algorithms for Rapid Objects With an Extended Set of Haar-like Features. Intel Technical Report MRL-TR, July 2002.

Lazarus, R.S., (1991). Emotion and Adaptation. Oxford University Press, New York, NY.

Lazarus, R.S., Folkman. S. (1987). Transactional theory and research on emotions and coping - European Journal of Personality.

Leung, S.H., Wang S.L., Lau, W.H. (2004) Lip image segmentation using fuzzy clustering incorporating an elliptic shape function, *IEEE Trans. on Image Processing*, vol.13, No.1.

Lienhart R., and Maydt J. (2002). An Extended Set of Haar-like Features for Rapid Object Detection. In *Proc. IEEE Intl. Conf. Image Processing (ICIP)*, Vol. 1, pp. 900-903.

Lorenz, K. (1935), Companions as factors in the bird's environment. In *Studies in Animal and Human Behavior*, volume 1, pages 101–258. London: Methuen & Co., and Cambridge, Mass.: Harvard University Press.

Neal, M.J. (2002). An Artificial Immune System for Continuous Analysis of Time-Varying Data. In J. Timmis and P. J. Bentley, editors, *Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS)*, volume 1, pages 76 -- 85, University of Kent at Canterbury, September 2002. University of Kent at Canterbury Printing Unit.

Neal, M., Timmis, J.: Timidity: A Useful Emotional Mechanism for Robot Control? *Informatica* 27, 197–204 (2003)

Ortony, A., Collins A., Clore. G. L. (1988) *The Cognitive Structure of Emotions*, Cambridge University Press.

Pantic, M., Rothkrantz, L.J.M. (2000) Expert system for automatic analysis of facial expressions, *Image and Vision Computing,* Vol. 18, 2000, pp. 881-905.

Picard, R. W. (1997) *Affective Computing,* MIT Press, 0-262-16170-2, Cambridge, MA, USA.

Rusting, C. (1998) Personality, mood, and cognitive processing of Emotional information: three conceptual frameworks, *Psychological Bulletin*, 124, 165-196.

Schaefer, A. M., Zimmermann, H. G. (2006) Recurrent Neural Networks are Universal Approximators, *ICANN 2006*, pp. 632-640.

Scherer, K. R. (1987) Toward a dynamic theory of emotion: The component process model of affective states, *Geneva Studies in Emotion and Communication*, 1, pp. 1–98, Geneva, Switzerland.

Scherer, K. R. (2001) Appraisal considered as a process of multi-level sequential checking, In K. R. Scherer, A. Schorr, & T. Johnstone (Eds.), *Appraisal processes in emotion: Theory, methods, research*, pp. 2–120, Oxford University Press, New York, USA.

Scherer, K. R. (2004). Feelings integrate the central representation of appraisal-driven response organization in emotion. In A. S. R. Manstead, N. H. Frijda, & A. H. Fischer (Eds.), *Feelings and emotions The Amsterdam symposium*, pp. 136–157, Cambridge University Press.

Schlosberg, H. (1954) A scale for judgment of facial expressions, *Journal of Experimental Psychology*, 29, pp. 497-510.

Sebe, N., Lew, M.S., Cohen, I., Sun, Gevers, Y. T., Huang, T.S. (2004) Authentic Facial Expression Analysis, *International Conference on Automatic Face and Gesture Recognition (FG'04)*, Seoul, Korea, May 2004, pp. 517-522.

Smith, P., Shah, M., da Vitoria Lobo, N. (2003) Determining Driver Visual Attention with One Camera, *IEEE Trans. Intelligent Transportation Systems*, Vol 4, No. 4, pp. 205-218.

Sroufe, L. A. (1995). *Emotional Development: The Organization of Emotional Life in the Early Years*. Cambridge University Press.

Tekalp, M., Ostermann, J. (2000) Face and 2-D mesh animation in MPEG-4, *Signal Processing: Image Communication* 15, pp. 387-421, Elsevier.

Tian, Y.L., Kanade, T., Cohn, J.F. (2001) Recognizing Action Units for Facial Expression Analysis, *IEEE Transactions on PAMI*, Vol.23, No.2.

Tomasi, C., Kanade, T. (1991) Detection and Tracking of Point Features, *Carnegie Mellon University Technical Report* CMU-CS-91-132.

Viola P., Jones, M. (2001) Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Vol. 1, pp. 511-518.

Whissel, C.M. (1989) The dictionary of affect in language, in Plutchnik, R. and Kellerman, H. (Eds.): *Emotion: Theory, Research and Experience: The Measurement of Emotions*, , Vol. 4, pp.113–131, Academic Press, New York.

Wundt, W. (1903) *Grundzuge der Physiologischen Psychologie*, vol. 2. Engelmann, Leipzig.

Edited by Mario I. Chacon M.

Notwithstanding the tremendous effort to solve the face recognition problem, it is not possible yet to design a face recognition system with a potential close to human performance. New computer vision and pattern recognition approaches need to be investigated. Even new knowledge and perspectives from different fields like, psychology and neuroscience must be incorporated into the current field of face recognition to design a robust face recognition system. Indeed, many more efforts are required to end up with a human like face recognition system. This book tries to make an effort to reduce the gap between the previous face recognition research state and the future state.

IntechOpen