FEATURES AND STATISTICAL CLASSIFIERS FOR FACE IMAGE ANALYSIS

QING SONG

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

FEATURES AND STATISTICAL CLASSIFIERS FOR FACE IMAGE ANALYSIS

by

Qing Song, B.Eng., M.Eng.

A thesis submitted to the

School of Graduate Studies

in partial fulfilment of the

requirements for the degree of

Doctor of Philosophy

Faculty of Engineering and Applied Science

Memorial University of Newfoundland

Jan 2001

St. John's Newfoundland Canada

Abstract

This thesis presents the systematic analysis of feature spaces and classification schemes for face image processing. Linear discriminants, probabilistic classifiers, and nearest neighbour classifiers are applied to face/nonface classification in various feature spaces including original greyscale space, face-image-whitened space, anything-image-whitened space, and double-whitened space. According to the classification error rates, the probabilistic classifiers performed the best, followed by nearest neighbour classifiers, and then the linear discriminant classifier. However, the former two kinds of classifiers are more computationally demanding. No matter what kind of classifier is used, the whitened space with reduced dimensionality improves classification performance.

A new feature extraction technique, named *dominant feature extraction*, is invented and applied to face/nonface classification with encouraging results. This technique extracts the features corresponding to the mean-difference and variance-difference of two classes. Other classification schemes, including the repeated Fisher's Linear Discriminant (FLD) and a moving-centre scheme, are newly proposed and tested. The Maximum Likelihood (ML) classifier based on hyperellipsoid distribution is applied for the first time to face/nonface classification.

Face images are conventionally represented by greyscales. This work presents a new representation that includes motion vectors, obtained through optical flow analysis between an input image and a neutral template, and a deformation residue that is the difference between the deformed input image and the template. The face images compose a convex cluster in this representation space. The viability of this space is tested and demonstrated through classification experiments on face detection, expression analysis, pose estimation, and face recognition. When the FLD is applied to face/nonface

classification and smiling/nonsmiling face classification, the new representation of face images outperforms the traditional greyscale representation. Face recognition experiments using the nearest neighbour classifier on the Olivetti and Oracle Research Laboratory (ORL) face database shows that the deformation residue representation is superior to all other representations. These promising results demonstrate that as a general-purpose space, the derived representation space is suitable for almost all aspects of face image processing.

Contents

List of Figures

List of Tables

List of Acronyms

FLD	Fisher's Linear Discriminant
ICA	Independent Components Analysis
LFA	Local Feature Analysis
MAP	Maximum a posterior
MDF	Most Discriminating Features
MEF	Most Expressive Features
MICD	Minimum Intra-Class Distance
ML	Maximum Likelihood
ORL	Olivetti & Oracle Research Laboratory
PCA	Principal Components Analysis

Acknowledgements

My biggest debt of gratitude goes to my supervisor, Dr. John Robinson, for his academic guidance and financial support. I have benefited enormously from his remarkable breadth of knowledge and capacity for insight, and from the diverse and energetic laboratory environment that he created at the Multimedia Communications Lab at Memorial University of Newfoundland.

I have also benefited greatly from the discussions of technical problems with my colleagues and friends: Li-Te Cheng, Charles Robertson, Xiaomeng Ping, Yan Shu, Xin Cheng, and Wenbo Pan.

I am very grateful for the help provided by the staff members of the Faculty of Engineering, particularly Dr. Sharp, Dr. Haddara, and Mrs. Moya Crocker who have helped me in a variety of ways in my graduate studies.

Financial support by School of Graduate Studies is gratefully acknowledged.

An important thanks goes to my dear husband, Zhihong Jia, for his love and encouragement throughout this endeavour.

Finally, I would like to thank my parents for all their sacrifices to give me the best education and life that they can provide, for their confidence in me, and for their unconditional love.

Chapter 1

Introduction

1.1 Background

Automatic face-image interpretation, a type of pattern recognition problem, has been studied for more than thirty years, and has become a particularly active research field in the last ten years. Face-image-processing tasks include face detection and tracking, face recognition, and interpretation (of pose, expression, etc.).

The importance of techniques that understand face images automatically comes from their extensive applications. Face recognition systems, such as FaceIt [Visionic] and True Face [Miros], have been used in access control, alertness monitoring, and surveillance. In addition, these techniques can be used in forensic applications, tele- and video-conferencing, and human-computer interaction. From another perspective, face-image research represents a cornerstone problem in natural scene analysis, and therefore has great value apart from any direct applications. Until now, the recognition of complex objects in normal conditions has met with only limited success. Since face images are one of the first types of natural images to be examined, and one of the most widely researched, a solution of the face-image interpretation problem would represent a breakthrough in image analysis.

Over the years, numerous strategies have been proposed for the face-image-processing tasks. The strategies include template matching, statistical pattern recognition

techniques and neural networks. A statistical pattern recognition system is composed of a preprocessing and representation module, a feature extraction module, and a classification module. The current methods of representing an image include grey-level, edge, shape, optical flow, Gabor wavelet output, etc., or their combination. The commonly used methods of extracting features are principal component analysis (PCA), local feature analysis (LFA), and independent components analysis (ICA). The classification module usually contains probabilistic classifiers (Bayes, MAP, ML), linear classifiers (FLD), neural networks, or *k*-nearest neighbour classifiers.

A face image is subject to variations in individual appearance, 3D pose, facial expression, illumination condition, and occlusion (facial hair, spectacles, etc.). Because of this, deformable templates that deal with and compensate for the main source of variation have become attractive. Lanitis et al. [Lanitis 1997] proposed a parameterised model of facial appearance which includes a shape model, and a shape-free greyscale model that is obtained by deforming a face image to have the same shape of the mean face. Jones and Poggio [Jones 1998] presented a morphable model for representing a face class. Each prototypical greyscale image in one class is converted into a shape vector and a texture vector. A shape vector describes how the 2D shape may change and the texture vector describes how the grey-levels may change. Nastar et al. [Nastar 1996] let a greyscale image have a deformable intensity surface. The intensity surface of one image can be deformed to the intensity surface of another image by using a 3D model. Elastic graph matching ("dynamic link architecture") [Tefas 1998] involves overlaying a grid on an image and deforming the grid. To sum up, deformable template matching has attracted a great deal of attention and is a promising direction of pattern recognition.

Face image processing systems differ in whether the various tasks are tackled separately or dependently. Because of the large amount of variation in face images, some

researchers have concentrated on particular applications and treated the various tasks independently. However, others attempt to develop "unified" approaches to the problems of face-image interpretation [Lanitis 1997]. The reason for proposing unified approaches is that these researchers believe all the sources of variability are so mixed that it is difficult to extract a description for one characteristic (e.g., expression) which is not susceptible to others (e.g., individual appearance).

All the tasks, face detection, recognition, expression analysis and pose estimation, can be performed on image sequences, or on static images. Among these tasks, facial-expression-analysis research has been focussed on image sequences, which facilitate the reduction of the influence of personal appearance and pose, and provide the temporal information. Face pose estimation is usually performed on image sequences too, where the expression change is required to be small. The techniques that solve these tasks on static images would be more general.

The last step in a face-image-processing system is to classify an unknown pattern into a group of classes. Conventional classifiers, probabilistic, FLD, *k*-nearest neighbour, etc., have been widely used. However, are there any other classifiers that can improve the classification performance?

The data obtained after the feature extraction step are commonly modelled by a Gaussian distribution. If other models are used, what will the results be?

All in all, conducting research on greyscale, static images and developing a representation applicable to all the face-image-processing tasks are of great importance.

1.2 Scope of This Study

This research is motivated by three goals.

The first goal is to develop an effective, general-purpose representation for face images. It is desirable to find a face representation that is general enough to satisfy all the face-image-processing tasks. We can identify criteria that such an ideal representation would fulfil. Assuming it consists of continuous features, then all images would be mappable into the feature space, but faces would form a compact, convex set within it. Different people's faces, different expressions, different poses, etc., would be well separated within the feature space. It has been found that combining shape information with intensity information improves face recognition accuracy. The new representation therefore utilises a deformable template that allows for variations in face pose and expression to some extent, and incorporates the shape and intensity information. After an image is represented using this method, the Principal Components Analysis (PCA) is applied because PCA provides a powerful selection and combination mechanism between input patterns. For classifying faces and nonfaces, we wish to characterise the shape of face space and define a face/nonface decision boundary. In the PCA derived feature space, the face set can be modelled as a hyperellipsoid. Note that if the nonface set extends to boundaries of the space, we may not need an enclosing solid - in the limit a hyperplane may be sufficient to divide space between face and nonface, so that the FLD may be an ideal classifier. However, we could detect a subclass of the nonface set that contains images that are "face-like". The class of face-like nonfaces can then be modelled as a hyperellipsoid. Then probabilistic classifiers can be used to define the boundary between the face class and the nonface class.

The second goal of this work is to propose new classifiers. Although classifiers for pattern recognition have been studied for over 50 years, there is still room for improvement. We investigate the FLD and propose a repeated FLD scheme that generates a group of hyperplanes for classification. Another moving-centre scheme

models the face class as a hypersphere and moves the centre and radius of this hypersphere iteratively using a steepest-descent algorithm. Testing the idea of modelling the data using a hyperellipsoid distribution instead of a Gaussian distribution is another contribution of this work. Moreover, a feature extraction technique is applied for the first time to the task of face detection.

The third goal of this work is a comprehensive, systematic investigation of classifiers including the ones developed here in high-dimensional space. Based on the same training and test sets, FLD, ML classifier, nearest neighbour classifier, k-nearest neighbour classifier, and k / l nearest neighbour classifier are applied to face/nonface classification. The experiments are carried out in the original greyscale space, anything-image-whitened space, face-image-whitened space, and/or double whitened space. The test results are plotted and/or tabulated for objective comparisons.

Chapter 2 reviews the contemporary pattern recognition techniques used in face-image-processing area, the definition and typical systems for each task, and the application of face-image-processing systems.

Chapter 3 describes the face images, nonface images, and anything-images used throughout this work, and three feature spaces for feature extraction.

Chapter 4 proposes several classifiers and compares them with commonly used classifiers. The test is done on face/nonface classification.

Chapter 5 proposes a new representation of face images. This representation utilises the motion vectors and deformation residue.

Chapter 6 applies this new representation to all the tasks of face-image-processing: face detection, face recognition, expression analysis and pose estimation. The experimental results are presented.

Chapter 7 summarises this work and outlines the future plan.

Chapter 2

Contemporary Face Image Processing Techniques: A Review

Over 30 years, especially recently, face images have received increasing attention in the academic communities in pattern recognition, computer vision, image processing and computer graphics. The main tasks in processing face images include the following:

- face detection and tracking,
- face recognition,
- facial expression analysis,
- 3D pose estimation, and
- gender and race analysis

These tasks are basically pattern recognition tasks with various face images or nonface images as input data. Researchers have proposed various approaches to fulfil these tasks. This chapter describes and compares these approaches, lists the results that have been achieved on each task, and outlines the broad applications of face image processing.

2.1 Approaches

The approaches in face-image-processing area can be grouped into three categories: template matching, statistical pattern recognition techniques, and neural networks [Jain 2000]. These approaches work independently or cooperatively.

2.1.1 Template matching

Template matching is one of the simplest and earliest approaches. An input pattern is compared with a stored template of the same type as the input pattern. Comparisons are performed by calculating their correlation, the Euclidean distance or Mahalanobis distance. The template is usually learned from the training samples. In greyscale template matching, this template is the greyscale values of the whole face or visually prominent facial regions, such as the eye area. Template matching strategies generally perform robustly against complex backgrounds but cannot deal with partially occluded faces.

One drawback of template matching systems is the huge computation time they require. Moreover, they are vulnerable to changes in scale and illumination. However, these problems are being tackled. Recently some hardware systems which can calculate correlation between a template and input image in real-time have become available. Multiscale templates are used to match patterns at various scales. Intensity preprocessing, including histogram equalisation and shade removal, has been used to reduce lighting effects.

2.1.2 Statistical approaches

Statistical approaches are the most popular approaches among the researchers in the face-image-processing area. In the statistical approach, every pattern is represented by M features or measurements. Therefore, every pattern can be viewed as a point in an M-

dimensional space. These M features should be chosen such that points from different classes occupy compact and non-overlapping regions in this feature space. Based on a set of training images, the decision boundary that separates different classes is established. The parametric form of the decision boundary can be linear or quadratic.

2.1.3 Neural network

"Neural networks can be viewed as massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections." [Jain 2000]. These "simple processors" refer to artificial neurons that are capable of learning nonlinear input-output relationships. Therefore, neural networks are adaptive to the data and able to approximate nonlinear functions.

The following are the commonly used neural networks for face-image-processing tasks.

- Multilayer perceptron (MLP) networks

  MLP networks are organized into layers. Each layer is composed of a linear network. These layers are glued together by feed-forward connections between them. Such a network was used for face detection in [Sung 1998].

- Convolutional networks (CN)

  Convolutional networks incorporate knowledge about the invariance of 2D shapes by using local connection patterns, and by imposing constraints on the weights. Convolutional networks combine three ideas to ensure some degree of shift, scale, and distortion invariance: local receptive fields, shared weights, and spatial sub-sampling. Each unit in a layer receives inputs from a set of units located in a small neighbourhood in the previous layer. The convolutional network is trained with the standard back-propagation algorithm.

Convolutional networks have been successfully applied to face detection [Rowley 1998] and face recognition [Lawrence 1997].

- Self-Organizing Map (SOM)

  The SOM, introduced by Kohonen [Kohonen 1995], is based on unsupervised, competitive learning. The SOM is primarily used for data clustering and feature mapping. The SOM is unlike most classification or clustering techniques in that it provides a topology-preserving mapping from the high dimensional space to map units. The mapping preserves the relative distance between the points, i.e., points that are near each other in the input space are mapped to nearby map units in the SOM. The SOM can thus serve as a cluster analysis tool for high-dimensional data. The topological preservation of the SOM process also makes it useful in the classification of data that includes a large number of classes.

  The SOM was used by Lawrence et al for face recognition in [Lawrence 1997]

In pattern recognition, neural networks allow simultaneous training of a set of discriminating hyperplanes. This ability to automatically learn from examples, along with the robustness, makes neural network approaches increasingly popular in the image processing literature.

Although neural network models and classical statistical pattern recognition techniques are apparently different, in essence they are equivalent or similar. For example, Fisher's Linear Discriminant can be approximated by a perceptron network.

2.2 Statistical Pattern Recognition

Statistical pattern recognition techniques have been successfully applied to many areas, including speech recognition, automatic target recognition and image classification. A statistical pattern recognition system usually consists of three modules:

- Preprocessing and representation

  The preprocessing module extracts a pattern from the background, removes the noise, and normalises the pattern. Sometimes the preprocessing module transforms the obtained pattern into another form of representation. For example, an image is not necessarily represented by greyscales. It can be represented by edges, optical flow, etc. This representation may increase or reduce the dimensionality of the input pattern.

- Feature extraction

  Feature extraction module transforms the input patterns so that they can be represented by low-dimensional vectors that can be easily matched or compared, and are relatively invariant with respect to transformations and distortions of the input patterns that do not change their nature. The most popular approaches in the face-image-processing literature are mainly identified by their differences in the input representation.

- Classifier selection

  In the training stage, a classifier is selected to separate the training samples in the feature space. In the case of parametric classifiers, the parameters are estimated based on the training samples. In the testing stage, the classifier assigns an input pattern to one of the identified classes based on the features of this pattern.

  In the training or learning stage, there are two kinds of learning methods: supervised learning and unsupervised learning. Unsupervised learning means that no human intervention during the learning or little knowledge about the training data is required. However, supervised learning means that every training sample is labelled with the class to which this sample belongs.

To sum up, the performance of a pattern recognition system depends on the preprocessing steps, feature extraction, and classifier selection. The feature extraction module is the part where most face-image-processing techniques differ.

2.2.1 Preprocessing and representation module

Previous investigators have shown that greyscale information is very important for interpreting face images. After an image is extracted from the background (picture) and normalised for size, usually other operations are performed on this greyscale image. These operations include compensating for lighting conditions and expanding the range of intensity.

Although images are often characterised directly in terms of pixel intensity, they can also be represented by other characteristics. The following are the attributes of face images that have been used by researchers.

- Edges

  Just like grey-levels, the direction and amount of edges are different from face to face. The edges in the mouth and eyes areas are usually the attributes of a face that researchers look for. The edge strength can be calculated by a variety of filters, such as the Sobel filter. Approaches based on edges are less dependent on colour and illumination but they require a plain background and are sensitive to partial occlusion (even the hair on the forehead). Edge information has been used in face detection and tracking in [Tsukamoto 1994a and b, Desilva 1995, Yang 1994]

- Shape and shape-free greyscale

  Inherited from the work of Craw and Cameron [Craw 1992], a parameterised model of facial appearance was proposed by Lanitis [Lanitis 1997]. A shape model models the shapes of facial features and their spatial relationships. Then a grey-level model

of "shape-free" appearance is generated by deforming each face in the training set to have the same shape as the mean face. Shape and grey-level models are used together to describe the overall appearance of each face image. In the training stage, a large number (152) of landmarks are manually marked on the training face images to define face shapes. This method is one of the deformable template approaches.

- Optical flow

  Using two successive frames of an image sequence, a two-dimensional vector field, called the optical flow, is computed which specifies the most likely displacement of image points between two frames. The scene in those two frames must obey the constraints: constant illumination, small difference between two adjacent frames, and no overlap between moving objects. Optical flow algorithms are generally used for facial expression analysis, although they have been applied to face recognition too [Kruizinga 1994]. Yacoob and Davis [Yacoob 1996] describe a method for interpreting facial expressions in image sequences based on optical flow. They analyse interframe motion of edges extracted in the area of the mouth, nose, eyes, and eyebrows. This method is not applicable to static images.

  Optical flow algorithms require both high computational cost and a well-ordered environment to obtain stable optical flow.

- Gabor Wavelet Representation

  Use of the 2D Gabor wavelet representation in image processing was proposed by Daugman [Daugman 1980, 1988]. The Gabor wavelet representation allows description of spatial frequency structure in the image while preserving information about spatial relations. The Gabor wavelets are of similar structure as the receptive fields of simple cells in the primary visual cortex (V1). They are located in both

space and frequency domains and have the shape of plane waves restricted by a Gaussian envelope function.

Gabor filters remove most of the variability in images due to variation in lighting and contrast. Representations of faces based on Gabor wavelets have proven successful for face detection [Krüger 1997] and facial expression analysis [Donato 1999].

However, Gabor functions are not orthogonal, so there is a trade-off between redundancy and completeness in the design of the Gabor filters. Second, the selection of filters is image dependent. Inappropriate selection would cause a generally impractical number of filters. Moreover, the application of the filters to images is not simple because of the large computation demand.

2.2.2 Feature extraction module

After an input pattern is appropriately represented, the feature extraction module transforms the input pattern into a lower-dimensional vector, whose components are called features. The reason for reducing the dimensionality is to cut down measurement cost and increase classification correctness. A typical face image used in face recognition is more than 100×100 pixels large. If this image is represented by grey-levels, it will be a point in a 10000 dimensional space. This high dimensionality makes it difficult or impossible to estimate the class-conditional-density function used by classifiers. This phenomenon is called "the curse of dimensionality"[Trunk 1979]. It is well known that the number of training samples per class should be at least ten times as many as the number of features. A small number of features can alleviate the curse of dimensionality when the number of training samples is limited. Added features may not be worth the

cost, but may actually degrade the performance of a classifier. On the other hand, the features cannot be too few or else discriminatory power will be lost.

Feature extraction serves not only to reduce the dimensionality, but also to capture the characteristics of the input pattern. We select features that are most effective for preserving the class separability. The commonly used feature extraction techniques in face image processing are the following.

- Principal Components Analysis (PCA)

  The best-known linear method for feature extraction and multivariate data projection is the Karhunen-Loève transform (KLT) or eigenvector expansion via principal components analysis (PCA). PCA generates a set of orthogonal axes of projections known as the principal components, or the eigenvectors, of the covariance matrix of input data in the order of decreasing variance.

  Given a training set of patterns $\{\mathbf{x}_i\}_{i=1}^{N_T}$. $\mathbf{x} \in R^{N\times 1}$, where N is the dimensionality of the input pattern and N_T is the number of patterns. The covariance matrix of this set is defined as

$$\mathbf{\Sigma} = \frac{1}{N_T - 1}\sum_{i=1}^{N_T}(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$$

  where $\mathbf{m}$ is the mean. $\mathbf{m} = \frac{1}{N_T}\sum_{i=1}^{N_T}\mathbf{x}_i$

  The KLT decomposes the covariance matrix $\mathbf{\Sigma}$ into the following parts

$$\mathbf{\Sigma} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^T$$

  where $\mathbf{\Phi} = [\phi_1 \ \ \phi_2 \ \ ...\phi_N] \in R^{N\times N}$ is an orthonormal eigenvector matrix and $\mathbf{\Lambda} = diag\{\lambda_1, \lambda_2, ..., \lambda_N\} \in R^{N\times N}$, a diagonal eigenvalue matrix with diagonal elements in descending order, i.e. $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$. The eigenvalue λ_i represents

the variance of the data set along the axis described by eigenvector ϕ_i. The first principal component points in the direction of maximum variability, the second principal component points in the direction of maximum variability orthogonal to the first, and so on.

PCA extracts a low-dimensional feature vector for an input pattern $\mathbf{x}$. This principal component feature vector is $\mathbf{y} = \mathbf{P}^T \tilde{\mathbf{x}}$, where $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{m}$, and $\mathbf{P}$ is composed of the first M columns of $\mathbf{\Phi}$, i.e., the largest M eigenvectors. $M << N$ and $\mathbf{P} \in R^{N \times M}$. The eigenvectors in $\mathbf{P}$ are now used as basis vectors of a lower-dimensional subspace.

An important property of PCA is decorrelation which means that in the PCA derived subspace the coefficients for one of the axes cannot be linearly predicted from the coefficients of the other axes.

Moreover, the original input pattern can be reconstructed from its feature vector. Multiplying by $\mathbf{P}$ on both sides of $\mathbf{y} = \mathbf{P}^T \tilde{\mathbf{x}}$ generates $\mathbf{P}\mathbf{y} = \tilde{\mathbf{x}}$ because $\mathbf{P}\mathbf{P}^T$ is an identity matrix. Therefore, the input pattern $\mathbf{x}$ is approximated by $\mathbf{x} = \mathbf{P}\mathbf{y} + \mathbf{m}$.

If PCA is performed on the greyscale representation of a set of face images, the extracted eigenvectors are face-like and often termed "eigenfaces", and the subspace is called "face space".

Eigenface-based face recognition [Sirovich 1987, Kirby 1990, Turk 1991] has been demonstrated to possess recognition abilities under certain varying input conditions, such as changes in lighting, expression and even partial occlusion of the face. This technique requires only one distance to the eigenvectors, instead of correlation with all the face patterns. Thus it is superior to the multiple template matching technique by saving memory and computation time, and by the ability to

deal with occlusion. PCA yields projection axes based on the variations from all the training samples; hence these axes are fairly robust for representing both training and testing images.

On the other hand, PCA is criticised for the following reasons.

1) It is not suitable for non-Gaussian data since PCA relies on the second-order property of the data.
2) PCA output does not preserve the local spatial relationships between pixels in images in the PCA output.
3) PCA performs worse in complex backgrounds than in uncluttered background.

- Local Feature Analysis (LFA)

Penev and Atick proposed local feature analysis (LFA) which defines a set of topographic, local kernels that are matched to the second-order statistics of the training samples [Penev 1996]. The kernels are derived from the principal component axes **P** and defined as

$$\mathbf{K} = \mathbf{P}\mathbf{V}\mathbf{P}^T \qquad \mathbf{K} \in R^{N \times N}$$

where $\mathbf{V} = \Lambda^{-\frac{1}{2}} = diag\left(\frac{1}{\sqrt{\lambda_i}}\right) \qquad i = 1,\ldots,M.$

M is the number of principal components to be retained in the PCA subspace. The whitening factor $\frac{1}{\sqrt{\lambda_i}}$ normalises the variance of training samples in the PCA subspace to unity.

The rows of **K** contain the kernels with spatially local properties. These kernels are "topographic" because they are indexed by spatial location.

The kernel matrix **K** transforms an input pattern **x** to

$$\mathbf{y} = \mathbf{K}(\mathbf{x} - \mathbf{m}) = \mathbf{K}\tilde{\mathbf{x}} \qquad \mathbf{y} \in R^N$$

where N is the dimensionality of $\mathbf{x}$.

Because $\mathbf{y} = \mathbf{K}\tilde{\mathbf{x}} = \mathbf{P}\mathbf{V}\mathbf{P}^T\tilde{\mathbf{x}} = \mathbf{P}(\mathbf{V}\mathbf{P}^T\tilde{\mathbf{x}})$, the LFA output $\mathbf{y}$ can be explained as the image reconstruction using normalised PCA feature vectors.

The dimensionality of $\mathbf{y}$ is the same as that of the input pattern $\mathbf{x}$. Unlike PCA, LFA does not simply take a number of top eigenvectors to reduce the dimensionality. A sparsification algorithm based on multiple linear regression was proposed to do so [Penev 1996].

Compared with PCA in face recognition, this technique was claimed [Visionic] to reduce the number of training images required and overcome some problems: sensitivity to poses, lighting condition and expression. In the 1996 FERET face recognition competition, an LFA based system outperformed PCA.

- Independent Components Analysis

  Like PCA, independent components analysis (ICA) [Bartlett 1998, Donato 1999] is a linear feature extractor. An input pattern is also represented as a linear superposition of basis functions. However, unlike PCA, ICA does not model the data as a multivariate Gaussian, does not require the axes to be orthogonal, and attempts to place them in the directions of statistical dependencies in the data. PCA and LFA are based on the second-order dependencies of training data, the covariance. ICA encodes the high-order dependencies in the training data in addition to the second-order dependencies.

  The ICA components are obtained by "blind-source separation" on the training data. The linear projection matrix is found using an algorithm that separates the statistically independent components of the training data through unsupervised

learning. The algorithm is based on the principle of maximum information transfer between neurons. The algorithm maximises the mutual information between the input and the output of a transfer function by maximizing the joint entropy of the output. This produces statistically independent outputs under certain conditions.

ICA has been successfully used in facial expression analysis [Donato 1999].

Besides PCA, LFA, and ICA, neural networks can also be used for feature extraction. For example, the SOM has been used for dimensionality deduction in [Lawrence 1997] and results similar to PCA are obtained.

2.2.3 Classifier selection module

Once the feature extraction module finds a proper representation, a classifier can be designed. In the face-image-processing field, currently there are three types of commonly used classifiers.

1) Probabilistic classifiers

Let $\omega_1, \omega_2, ..., \omega_c$ denote the object classes, and $\mathbf{x} = (x_1, x_2, ..., x_M)$ a vector of M feature values. The features are assumed to have a probability density on each class. This probability is called class-conditional probability, and denoted as $p\{\mathbf{x} | \omega_i\}$. The class to which $\mathbf{x}$ belongs can be decided by the following probability based classifiers.

- Bayes decision rule

  The Bayes decision rule is that the cost of misclassification should be minimised. Define c_{ij} as the cost of assigning $\mathbf{x}$ to class ω_i when actually $\mathbf{x}$ belongs to class ω_j. If

$$c_{ij} = \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases},$$

the Bayes decision rule is simplified so that the input pattern $\mathbf{x}$ is assigned to the class ω_i if

$P(\omega_i \mid \mathbf{x}) > P(\omega_j \mid \mathbf{x})$ for all $j \neq i$

$P(\omega_i \mid \mathbf{x})$ is the *a posteriori* probability.

This simplified rule is called maximum *a posteriori* (MAP) rule

- Maximum likelihood (ML) classifier

  According to Bayes theorem

$$P(\omega_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \omega_i) P(\omega_i)}{p(\mathbf{x})}$$

  where $P(\omega_i)$ is *a priori* probability, $p(\mathbf{x} \mid \omega_i)$ is the class-conditional probability density, and $p(\mathbf{x})$ is the mixture density. Since in this expression $p(\mathbf{x})$ does not depend on the class ω_i, it can be discarded. The ML classifier further assumes that the c classes are equiprobable. The Bayes decision rule becomes

  $\mathbf{x}$ is assigned to the class ω_i if

  $p(\mathbf{x} \mid \omega_i) > p(\mathbf{x} \mid \omega_j)$ for all $j \neq i$

  A classifier based on this rule is called a maximum likelihood (ML) classifier because its cost function maximises $p(\mathbf{x} \mid \omega)$.

The probabilistic approaches require estimating density functions first, and then constructing the discriminant functions that specify the decision boundaries. The Bayes classifier is optimal according to statistical pattern recognition theory since the Bayes error is the best criterion to evaluate feature sets. But for applications

involving high-dimensional signals, the demand for a large number of training samples to construct a good Bayes classifier is difficult to satisfy due to the lack of training samples.

Probabilistic approaches have been used in almost every aspects of face-image processing, such as face detection [Moghaddam 1994], and face recognition [Liu 1998].

2) Linear classifiers

Linear classifier design is a special case of feature extraction, involving the selection of a linear mapping to reduce the dimensionality. In two-class classification case, the number of features is one. Classification is then performed by specifying a single threshold.

The most commonly used linear classifier is Fisher's Linear Discriminant (FLD). The FLD is a supervised learning procedure that projects the images into a subspace that maximises the between-class scatter while minimizing the within-class scatter of the projected data. This approach assumes linear separability of the classes. The FLD minimises the mean squared error (MSE) between the classifier output and the desired labels.

The FLD is a general approach for pattern recognition; for example, it performed well for recognising faces under changes in lighting [Belhumeur 1997]. The FLD is simple and fast. However it is criticised for lack of generalisation ability [Donato 1999, Liu 1998]. The generalisation ability of a classifier refers to its performance in classifying test patterns that were not used during the training stage. The FLD is said to overfit to the training data. Second, when the classes have very different underlying covariance matrices, a quadratic classifier is a better choice than the FLD.

Furthermore, the projection of the data onto a very few dimensions can make linear separability of the classes impossible.

3) Nearest neighbour classifiers

The nearest neighbour classifier assigns a test pattern to the class of the nearest training pattern. It requires the computation of the distances between a test pattern and all the patterns in the training set. Euclidean distance is usually the distance metric.

The *k*-nearest neighbour classifier assigns patterns to the majority class among *k* nearest neighbours in the training set. Unlike the nearest neighbour classifier, the *k*-nearest neighbour classifier needs training. The parameter *k* is chosen such that the classification error in the training set is lowest.

Nearest neighbour classifiers are slow and metric dependent. Nevertheless, no probability density function of training data needs to be estimated, and they perform quite stably at high-dimensions.

2.2.4 Combination of three modules

The success of a pattern recognition system relies on the appropriate design of three modules: preprocessing and representation, feature extraction, and classifier selection.

Liu and Wechsler [Liu 1998] proposed a model that combines the PCA technique and the Bayes classifier, and applied it to face recognition.

A technique which attracts much attention is PCA plus FLD. Belhumeur et al [Belhumeur 1997] developed an approach called "Fisherfaces" for face recognition by applying first the PCA for dimensionality reduction and then the FLD for discriminant analysis. Swets and Weng [Swets 1996] used a similar approach.

2.3 Deformable Template Based Approaches

As mentioned in Section 2.1.1, rigid template matching is susceptible to the pattern distortion due to the imaging process, viewpoint change, or large intraclass variations among the patterns. Deformable template models can be used to overcome this susceptibility and this has become one of the frontiers in pattern recognition.

In recent years, various deformable models have been proposed.

[Lanitis 1997]

Lanitis et al. proposed a parameterised model of facial appearance (described in Section 2.2.1). This model includes a shape model, and a shape-free greyscale model that is obtained by deforming a face image to have the same shape of the mean face. Then PCA is performed to reduce the parameters of this model to less than 100. This representation of a face image somewhat compensates for the pose and facial expression difference among the faces in one class. The authors called this approach a unified approach because it has been used in all the aspects of face-image processing.

[Jones 1998]

Jones and Poggio presented a morphable model for representing a face class. A set of prototypical images in a class are used to train this model. Each prototypical greyscale image is converted into a shape vector and a texture vector. The shape vector for a prototype is a flow field that contains the pixelwise correspondences to that prototype from a reference prototype. The texture vector for a prototype image is the image in which the grey-levels of the image are moved to the corresponding positions in the reference prototype. The reference prototype is just a sample image. The match between

a morphable model and a novel image is conducted by minimizing an error function that compares grey-level values between the model and novel images. This model was claimed to be good for face recognition and robust to partial occlusion.

[Nastar 1996]

Nastar et al. introduced an approach termed the *deformable intensity surface*. The intensity surface of a test image is warped into the intensity surface of a prototype image by using a 3D model. The low order part of strain energy that describes the amount of energy needed to deform a surface into another is used as a similarity measure. This approach deals with object appearance variations, and has been used in face detection and recognition.

2.4 Areas of Face Image Processing

There is tremendous amount of variability among faces. First, the skin colour, face shape, and facial organs differ from face to face. Second, for a particular person, the face image is subject to the following changes: facial expression, pose or orientation, hairstyle, facial hair (beard and/or moustache), and eyeglasses. And finally, in the face image that face can appear against any complex background under various illuminations.

In face recognition, only the face structural difference is emphasised, while other variability, such as pose, expression and lighting, is suppressed. Likewise, pose estimation only pays attention to the orientation of a face and ignores expression and face identity.

The aforementioned pattern recognition techniques are carefully selected and combined together for use in the face-image processing area.

2.4.1 Face detection

Automatic *human face location*, or *detection*, detects and finds the position and size of each human face (if any) in a two-dimensional natural, complex scene [defined in Sung, 1998].

The key issue and difficulty in face detection is to account for a wide range of variations in face images. There have been numerous innovative strategies proposed for solving this problem. Most of them used image invariants [Rowley 1998], snake (spline) [Vieren 1995, Welsh 1991], template(s) [Sung 1998, Lanitis 1997, Tsukamoto 1994a and b], or eigenfaces [Pentland 1994, Turk 1991].

Face detection algorithms on a single image are usually performed in the following way. The system scans through an input image and receives a small window as the test pattern for examination. Psychologists find that when a face occupies less than 18×18 pixels, it is hard for human beings to identify it as a face [Bruce 1991]. This might account for the fact that researchers use a window size of at least 19×19 pixels. Every test pattern is preprocessed (corner masking, shade removal and histogram equalisation) and image pyramids are adopted to cope with scale variations. The difference between various face detection systems lies in the feature extractor and the classifier. Some face detection systems have achieved encouraging results.

[Rowley 1998]

Rowley, Baluja and Kanade demonstrated a neural network-based face detection system. They applied two networks each with one hidden layer. The system arbitrates between the two networks to improve performance over a single network by ANDing, ORing. They recently extended this method to detect faces at any degree of in-plane

rotation [Rowley 1997]. A separate neural network, called a "router", is added to the above system. The input image is first sent to the router, which examines every test pattern and returns the angle of that test pattern. According to this angle, the test pattern is rotated back to upright, and finally sent to the above system.

[Sung 1998]

Sung and Poggio proposed an example-based learning approach for face detection They synthesised six "face" pattern clusters and six "nonface" pattern clusters. Two distance-metrics are used for measuring the distance from a test pattern to a cluster prototype. The first measurement is a normalised Mahalanobis distance between the test pattern and the cluster prototype, in a low dimensional subspace created by the cluster's 75 most significant eigenvectors, and the second measurement is the normalised Euclidean distance between the test pattern and its projection in the 75-dimensional subspace. Finally, these 12 distance-measurement pairs are fed into a multilayer perceptron (MLP) net classifier that gives the ultimate decision.

[Schneiderman 1998]

Schneiderman and Kanade derived *a posterior* probability function $P(\text{class} \mid \text{image})$ for detecting human faces from frontal and profile views. A face region of 64×64 pixels is divided into subregions of 16×16 pixels. The intensity pattern in a subregion is called the local appearance. The space of local appearance is partitioned into a finite number of patterns. The frequency of these patterns over various sets of training images is counted. The functional form of the posterior probability function combines joint statistics of local

appearance, the position of a subregion in the whole face region, and the frequency of occurrence of this finite set of patterns.

Table 2.1 Performance of three face detection systems; modified from [Schneiderman 1998]

Paper	Window size	Detection rate	Number of false alarms	Test set
[Rowley 1997]	20 × 20	92.5%	862	Combined sets of [Sung 1998] and [Rowley 1998] (130 images)
[Sung 1998]	19 × 19	84.6%	13	23 images in [Sung 1998] test set
[Schneiderman 1998]	64 × 64	93.0%	88	Combined sets of [Sung 1998] and [Rowley 1998] excluding 5 images of line drawn faces (125 images)

Table 2.1 lists the performance of these systems for comparison. The system of [Schneiderman 1998] performed the best, partially because the size of test pattern is bigger.

The achievement of these face detection systems teaches us that this task is fulfilled by extracting the structural similarity and reducing detailed variations among all the face patterns. This principle could also be applied to other similar pattern recognition problems, for instance, object detection.

Note that the face images do not compose a compact cluster in the original space. To solve this problem, Rowley et al. used multiple hyperplanes as the decision boundary, while Sung and Poggio used multiple nonface clusters to insert between the face clusters.

2.4.2 Face recognition

Face recognition refers to the automatic identification of a human face. In the face recognition task, a face image is compared with each of the images of the human faces stored in a database, whose identities are known. Each comparison produces a similarity score, which indicates the degree of similarity between the pair of human faces compared. As a result, a matching candidate list can be produced in descending order of the similarity scores.

A face recognition system should be able to exploit the differences between separate faces for the purpose of identification while removing the differences that may be present in multiple images of the same face. As mentioned before, multiple images of a single face may have the variations in scale, positioning, orientation, illumination, facial expression, and age, as well as the addition or removal of eyeglasses and facial hair. Face recognition systems have been designed to be as robust as possible to filter out some or all of these variations.

A face recognition system is just a special kind of pattern recognition system. It includes the input/output part, any processing performed on the input image to extract features, and the classifier. The most popular approaches in the face recognition literature are mainly differentiated by the feature extraction part. Until now, PCA, LFA, Gabor filters, optical flow, and Hidden Markov Model (HMM) have been used for representation, while the FLD, Bayes classifier, and neural networks have been used for classification.

PCA, or eigenfaces, for face recognition was proposed thirteen years ago [Sirovich 1987]. Since then, many interesting theories and techniques have been proposed to enhance eigenface-based face recognition.

1) Complementary eigenspaces

Moghaddam and Pentland [Moghaddam 1998] proposed an improved probability model that exploits not only the principal features, but also extra information inherent in multiple training images in a face class. Two complementary sets of eigenfaces are found and employed. This system utilised both intra- and extra-facial variations while minimizing its sensitivity to intra-facial variations. This technique combined with Maximum A-Posteriori (MAP) rule showed a performance increase of 5% over standard PCA when evaluated on the Ferret Database.

2) View based eigenspaces

The use of multiple subspaces, each corresponding to a different head orientation has been reported in [Pentland 1994, Frey 1998]. This technique, called view-based eigenspaces, allows the recognition of faces from multiple orientations simultaneously, thus allows a face recognition system to operate in considerably more complex environments. Eigenfaces are generated for several separate sets of facial images, each corresponding to a different characteristic view (e.g. 45 degrees, frontal, profile etc.). Gaussian probability models are used to convert weightings that link the distance from face space to an absolute probability. The probabilities corresponding to several viewpoints are compared and the closest match is determined.

3) Eigenfeatures

The use of eigenfeatures has been proposed in [Pentland 1994] to decrease sensitivity to changes in expression, disguise and occlusion. The eigenfeatures technique consists of estimating face identity using a combination of both global facial and local feature information. The eyes, nose, mouth and other significant features are located and extracted from the face being recognized. The set of features

are then projected into a feature space in a similar way to the projection into face space used by the eigenfaces technique.

4) Fisherfaces

Belhumeur et al [Belhumeur 1997] developed a face recognition approach that is insensitive to large changes in lighting direction and facial expression. This approach, called "Fisherfaces", combines PCA and FLD techniques. Similar to the Fisherfaces approach, Swets and Weng [Swets 1996] mentioned that the eigenfaces derived using PCA are only the most expressive features (MEF). The MEF are suitable for face representation but unrelated to actual face recognition. In order to derive the most discriminating features (MDF), a subsequent FLD projection is needed. Their experimental results show that MDF provides an effective feature space for face recognition. However, the MDF space is superior to the MEF space only when the training images include the main range of variations in a face class. Moreover, the combined technique also has the drawback of the FLD: poor generalisation to new subjects.

Besides the above PCA based approaches, other approaches have been proposed. Kruizinga and Petkov [Kruizinga 1994] compute the optical flow between two face images and use it to get a measure for the dissimilarity of the images. Based on this distance, they search for the nearest neighbour of an input face image in a database of pre-stored face images and use the search result for person identification. They used an image pyramid, divided the face image into 8×8 pixel blocks, and thus calculated the optical flow. They achieved 92% recognition rate on a 38-person database.

Current face recognition techniques have been investigated and compared by several researchers. Table 2.2 lists some benchmark results obtained on the same face databases.

Note that other techniques that have not been applied to these two databases are not included.

The two databases used are the University of Bern (UB) [Bern] and the Olivetti & Oracle Research Laboratory (ORL) [Olivetti] face databases. The UB database contains 10 face images of each of 30 persons, while the ORL database contains 10 face images of each of 40 persons. The head sizes in UB and ORL are 170×230 and 92 × 112 pixels respectively.

Table 2.2 Comparative recognition rates for ORL and Bern face databases; modified from [de Vel 1999].

Paper	Technique	ORL	Bern
[Samaria 1994]	HMM	95.0	-
[Zhang 1997]	Eigenface	80.0	87.0
	Elastic matching	80.0	93.0
[Lin 1997]	Neural network	96.0	-
[Lawrence 1997]	Neural network	96.2	-
	Eigenface	89.5	
[Achermann 1996]	HMM	-	90.0
	Eigenface	-	94.7
	Combination	-	99.7
[de Vel 1999]	Line segments	99.7~100	99.7~100

Samaria and Young [Samaria 1994] used a Hidden Markov Model (HMM) to model the statistical and structural information of face images simultaneously. Face images are segmented into regions. Assuming that each face is in an upright, frontal position, features will occur in a predictable order, i.e. forehead, eyes, nose etc. This natural order is modelled by the HMM.

Zhang et al. [Zhang 1997] compared the eigenface approach and an elastic template matching approach on face recognition. Elastic matching approach is based on the Gabor filter representation of face images, and uses an energy function for similarity measurement. Four individual databases and their combination of 113 persons were used for experimental evaluation of these approaches. They concluded that eigenface classifier performed well on the individual databases where lighting conditions were consistent, but performed badly on the combined database due to the lighting condition difference among the databases. The elastic template approach performed comparatively well. It was found to be insensitive to the variation in lighting condition, face position and expression.

Lin et al [Lin 1997] first extracted specified face regions from images, normalised the face region, and then used intensity and edge information with a neural network to recognise faces.

Lawrence et al. [Lawrence 1997] tested local image sampling with SOM and convolutional neural networks in addition to the eigenface classifier.

Achermann and Bunke [Achermann 1996] implemented a face recognition system based on the combination of an eigenface classifier, a classifier based on hidden Markov models (HMM), which is similar to [Samaria 1994], and a profile classifier. When tested on a 30-person database (University of Bern database) with moderate pose variation among the 10 views per person, the eigenface classifier performed best (94.7 percent), followed by the HMM classifier (90.0 percent) and the profile based classifier (85.0 percent). The combination of eigenface classifier and HMM classifier gave the best result (99.7 percent).

De Vel and Aeberhard [de Vel 1999] proposed a new representation of face images: a set of random one-dimensional rectilinear line segments. They used multiple views of

the same person in the viewing sphere The combination of 1D line segments exploits the inherent coherence in one or more 2D face image views in the viewing sphere. This representation was claimed to be robust to in-plane rotation, scale invariant, changes in illumination intensity, but not to changes in illumination direction. Making use of a nearest-neighbour classifier, their system achieved 99.7% to 100% recognition rates on ORL and Bern databases in quasi-real time.

2.4.3 Facial expression analysis

Facial expression determination is the classification of the change in a person's facial features. Ekman and Friesen [Ekman 1978] categorised spontaneous facial expressions into happiness, sadness, surprise, fear, anger, disgust, and neutral. They have produced a widely used system for describing "all visually distinguishable facial movements," called the *Facial Action Coding System* or *FACS*. It is based on the enumeration of a face's all "action units" that cause facial movements. Many of the expression recognition systems reported in the literature are trained to classify expressions into these seven categories. Understanding facial expression is an important task in human-computer interaction.

The work on facial expression analysis has focused on facial motion analysis through optical flow estimation [Essa 1997, Yacoob 1996, Cohn 1999, Rosenblum 1996] or surface textures based on PCA [Padgett 1997, Lanitis 1997, Colmenarez 1999].

1) Optical flow based methods

Essa and Pentland [Essa 1997] used optical flow to estimate activity in a detailed geometric and physical model of both the skin and muscle of the face. They used both the temporal and spatial information. The optical flow was estimated in a feedback-controlled framework. The estimated motion was then used to classify the

facial expressions. A recognition rate of 98% was achieved on a database of 52 sequences.

Yacoob and Davis [Yacoob 1996] constructed a mid-level representation of facial motion directly from the optic flow based on individual pixels. The interframe motion of edges in the mouth, eyes, and eyebrows were analysed. In the training stage, they established a set of rules about the motion of these edges involved in an expression. These heuristic rules are applied to the mid-level representation to create a complete temporal map describing the evolving facial expression. These mid-level representations were classified into one of six facial expressions. The system achieved a recognition rate of about 85% in 46 image sequences when the rigid head motion was kept as small as possible. Rosenblum, Yacoob, and Davis [Rosenblum 1996] expanded this system. Instead of using heuristic rules, a radial basis function network was used to learn the correlation of facial feature motion pattern and human emotions.

Cohn et al. [Cohn 1999] developed a system for automatic facial action classification based on feature-point tracking, dense-flow tracking with PCA, or high gradient component (furrows of the face) detection. The displacements of 36 manually located feature points are estimated using optical flow. Pixel-wise dense flow was calculated and compressed using PCA. HMM was used for classifying the facial expressions. The recognition results of the upper face expressions using each method were 85%, 93%, and 85% respectively.

2) Surface textures based on PCA

Padgett and Cottrell [Padgett 1998] used a feed forward neural network for recognising expressions. The features from the eye and mouth regions, represented in

grey-levels, were extracted and PCA was performed. When trained and tested on the FACS image sequences, this neural network categorised facial expressions.

Based on the FACS database, Donato et al [Donato 1999] explored and compared techniques for facial expression analysis, and then obtained the comparisons in Table 2.3.

Table 2.3 shows that the local Gabor filter representation and ICA representation performed the best. They concluded that using local filters and high spatial frequencies was the key to the success of a face expression analysis system.

Table 2.3 Correct rate for various face expression analysis techniques

Technique	Correct rate
Optical flow and correlation	85.6% ± 3.3
PCA	79.3% ± 3.9
LFA	81.1% ± 3.7
FLD	75.7% ± 4.1
ICA	95.5% ± 2.0
Gabor filter	95.5% ± 2.0

Note that all these approaches used temporal information and are not applicable to static images.

2.4.4 Pose estimation

Face pose is the 3D orientation of the face relative to the camera. A face can lean left or right, tilt left or right, or nod. A human head has 3 degrees of freedom. Suppose a vertically oriented frontal face is located at the zero position of a 3D pose "ball", the range of all possible poses are -90° ~ 90° in azimuth, -90° ~ 90° in elevation, and -90° ~ 90° in leaning angle. Pose estimation ought to recover all the possible poses.

The knowledge of the pose is useful to face recognition, where an appropriate template can be chosen to speed up the recognition process or the face can be rotated back to facilitate recognition.

In 3D wireframe model-based coding systems for facial image communication, the pose information assists the mapping from the original 2D image to that 3D model [Forchheimer 1989].

Current proposed pose estimation systems are commonly associated with face locating algorithms. The following groups of people have conducted extensive research in this area.

[Gee 1994]

Gee and Cipolla described a simple method of estimating the face pose and gaze in a single, monocular view. They assume that the five feature points, the far corners of the eyes and mouth, and the nose tip, are already known and will not be affected by facial expressions. Then based on the measurements provided by these feature points, they calculated the pose and gaze direction. This method does not deal with leaning head cases. The accuracy of results depends on the accuracy of location of these feature points. The position of those points is also affected by facial expressions and is different from face to face. The advantages of this strategy are the small amount of calculation and insensitivity to changes in face size and illumination. In [Gee 1996], they estimated face pose in an image sequence. A dark pixel detector automatically detects the locations of a group of facial features.

[Brunelli 1997]

Brunelli recovered face pose by using the asymmetry between the two eyes due to in-depth (left-right) rotation. After the left and right eyes are located and the illumination is compensated, the asymmetry is measured as the relative amount of gradient intensity in two eye regions. The author observes that the gradient asymmetry approximately linearly depends on the amount of rotation around the vertical image axis. The pose of the face is restricted to in-depth rotations. The method is reportedly fast and does not require calibration.

[Tsukamoto 1994a, 1994b]

Tsukamoto et al. presented a method for face tracking and pose estimation from an input image sequence. They first divide the face area into 7×5 blocks, which are parameterised by intensity and edge busyness. After detecting frontal faces, they map the 2D image onto a 3D face model, then rotate the model in three directions: left-right, up-down, and in-plane (such as the leaning of head), and finally re-map the model onto the image plane. In this way, they synthesise some model images in different poses. For a new incoming image, they calculate the correlation (intensity difference) between this image and other model images. The face pose is computed as the linear combination of these correlations. This strategy requires a large amount of memory and computation time, and is sensitive to abrupt changes in illumination, and partial occlusion. However, it does not demand the location of any facial features

[Krüger 1997]

Krüger et al. presented a system that employs labelled graphs to estimate position, size and pose of a human head in a still image. Every face is represented by a graph,

which is composed of dozens of nodes and edges between adjacent nodes. The nodes are labelled with the convolutions of the local intensity with a set of Gabor wavelets. The edges are labelled with the distance between its connected nodes. For a particular pose, they built a labelled graph (pose graph). The pose of a new image is determined by comparing its similarity to all the pose graphs. The graph with the highest similarity gives the pose for that image. While only five pose graphs (the pose graphs for frontal views, two profile views, two views between profile and frontal) are shown in that paper, how many poses have been represented by graphs are unknown. Plus it is impossible to build graphs for all possible poses. The head rotation is limited to in-depth rotation and the processing of one picture requires 112s on a SPARC 20. Furthermore, this system is very dependent on manual construction of the initial pose graphs.

2.4.5 Description of gender, race and age

Another aspect of face image processing involves determining the gender, race, age, etc., of a face. Gender recognition has received little attention because it is strongly subject to hair and makeup. Similarly, the recognition of race and age is unexplored.

2.5 Applications

The research in face-image processing area has been stimulated by a broad range of applications for systems able to code and interpret face images. For example:

- *Forensic applications* [Strother-Vien, 1998], such as in creating a composite sketch of a suspect and then finding a match in a mug shot database.
- *Personal identification* [Nelson, 1998] (credit cards, driver's licence, passports, employee ID).
- *Access control*

Access control has been implemented by two face-recognition commercial products [Visionics, Miros], which can restrain the access to check-cashing ATMs, building or security sensitive rooms).

- *Security monitoring*

  A face tracking and recognition system can be used as a security system in shopping malls, other public areas, or private houses.

- *Video coding, video databases, and teleconferencing system*

  It is well known that people are most sensitive to coding errors in facial features. The coder would

  - Encode very precisely facial features (such as eyes, mouth, nose, etc.)
  - Encode less precisely the rest of the picture.

  This requires that the coder first detects face location, and then exploits this information to achieve high quality coding.

  Facial expression determination is an active research interest for image coding of facial video sequences

- Face detection would have immediate applications for image enhancement during film processing and automatic storage and retrieval of pictures, such as in a newspaper archive

- *Human-computer interaction* (HCI)

  Currently some computer games can be played with head movement, instead of mouse or keyboard (but the player must wear headgear). The automatic estimation of the head pose enables people to communicate with a computer using head gestures, which would be very helpful to handicapped people.

Furthermore, [Robinson 1998] introduces a virtual figure who recognises and talks to the user and knows what the user is doing, such as drinking a coffee. This kind of application makes the computer more like a human being than a machine.

Chapter 3

Data Preparation and Feature Spaces

One of the goals of this research is to compare the performance of different classifiers. In order to fulfil this goal, the training and test data sets need to be properly prepared. This chapter describes the method of obtaining face images, nonface images, as well as the baseline feature spaces in which the classification experiments will be conducted.

3.1 Data Preparation

3.1.1 Extracting face regions

In [Rowley 1997] Rowley et al. manually marked the eyes, tip of the nose, and the corners and centre of the mouth of 1048 faces, aligned the marked faces to each other using an iterative procedure, and generated the facial feature distribution as shown in Figure 3.1.

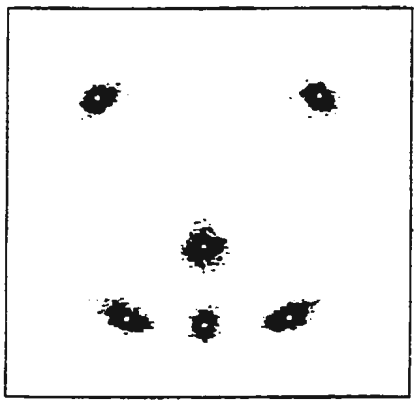


Figure 3.1 Positions of average facial feature locations (white circles), and the distribution of the actual feature locations from all the examples (black dots). Reproduced from [Rowley 1997].

These feature locations are used here as the basis of extracting the face regions from the images.

Suppose the 6 average feature locations, i.e., the white circles, in Figure 3.1 are denoted as (d_{xi}, d_{yi}), $i = 1, 2, ..., n$, $n = 6$. However, the manually marked 6 feature locations on a face image are (m_{xi}, m_{yi}), $i = 1, 2, ..., n$. We developed a method of computing the rotation, translation, and scaling that minimise the distances between the corresponding features.

The rotation, translation, and scaling are calculated as follows.

1) Rotate the picture according to the locations of two eyes, and make the line connecting two eyes of the rotated picture parallel to the horizon. The rotated feature locations are (m'_{xi}, m'_{yi}).

2) If the rectangle region with the top left corner position (s_x, s_y) and with equal width and height is the desired face region, then the squared distance between the

average feature locations and the transformed feature locations should be minimum.

The squared distance is

$$E = \sum_{i=1}^{n} ((\gamma(m'_{xi} - s_x) - d_{xi})^2 + (\gamma(m'_{yi} - s_y) - d_{yi})^2) \tag{3.1}$$

where γ is the scaling ratio between the desired face region size and the actual face region size.

To minimize E, we set

$$\frac{\partial E}{\partial s_x} = 0, \frac{\partial E}{\partial s_y} = 0, \text{ and } \frac{\partial E}{\partial \gamma} = 0$$

By solving the above three equations, we get

$$\gamma = \frac{\frac{1}{n}\sum_{i=1}^{n} d_{xi} m'_{xi} + \frac{1}{n}\sum_{i=1}^{n} d_{yi} m'_{yi} - \bar{m}_x \bar{d}_x - \bar{m}_y \bar{d}_y}{\frac{1}{n}\sum_{i=1}^{n} m'^2_{xi} + \frac{1}{n}\sum_{i=1}^{n} m'^2_{yi} - \bar{m}_x^2 - \bar{m}_y^2} \tag{3.2}$$

$$s_x = \bar{m}_x - \frac{1}{\gamma}\bar{d}_x \tag{3.3}$$

$$s_y = \bar{m}_y - \frac{1}{\gamma}\bar{d}_y \tag{3.4}$$

where $\bar{m}_x = \frac{1}{n}\sum_{i=1}^{n} m'_{xi}$, $\bar{m}_y = \frac{1}{n}\sum_{i=1}^{n} m'_{yi}$, $\bar{d}_x = \frac{1}{n}\sum_{i=1}^{n} d_{xi}$, and $\bar{d}_y = \frac{1}{n}\sum_{i=1}^{n} d_{yi}$.

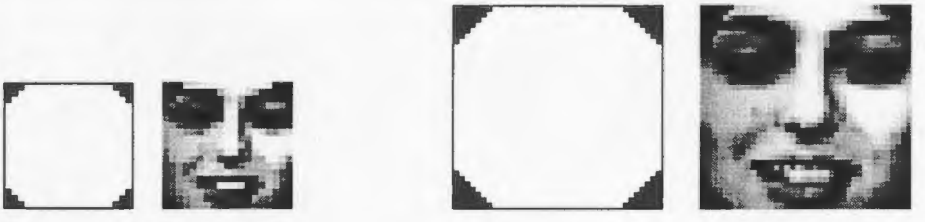
Then the determined face region is extracted according to the parameters s_x, s_y and γ. When the original face picture is resized, a mean filter with a size of $\frac{1}{s_x} \times \frac{1}{s_x}$ is applied to generate a smooth image.

3.1.2 Preprocessing steps

Currently the extracted images are of two sizes: 19 × 19 pixels and 38 × 38 pixels. The preprocessing steps are performed on the every image no matter whether it is a face image or not. These steps are as follows:

1) Masking

Two masks corresponding to two face sizes are used. Pixels outside the oval mask will be excluded in the computation because they may represent the background. The masks and masked images are shown in Figure 3.2. 6 pixels and 21 pixels in each corner are removed in these 19 × 19 and 38 × 38 pixel images respectively. After masking, the number of remaining pixels is 337 and 1360 for these 19 × 19 and 38 × 38 pixel images respectively.



(a) 19 × 19 pixel (b) 38 × 38 pixel

Figure 3.2 Masks and masked images

2) Shade removal

The average intensity of each row and column of a face pattern is first calculated separately (excluding the corner pixels). Then the best-fit horizontal and vertical lines are obtained. These two lines compose a best-fit linear plane that approximates the intensity of every part of the face. After this plane is subtracted from the face pattern, the shade caused by different lighting conditions is reduced or removed. The beneficial effect of shade removal is shown in Figure 3.3b.

3) Histogram equalisation

This operation replaces the original grey-levels by the scaled frequency of values up to and including the current value, which is obtained from the cumulative histogram. The histogram is computed for pixels inside the oval mask. Histogram equalisation increases contrast as shown in Figure 3.3c.

(a)original image (b)shade removed (c)histogram equalised

Figure 3.3 Preprocessing steps on a face image

The shade removal and histogram equalisation steps are adapted from [Rowley 1998] and [Sung 1998]

3.1.3 Face image preparation

The face images were obtained via the Internet from the University of Stirling face database, Yale University face database, MIT face database, and others. These face images are the frontal or near-frontal views of faces. A large amount of variation in lighting condition, pose, and expression exists among these face images. From each original image, we first generate five face examples by extracting the face region, rotating the extracted face region (about its centre points) to the left 5° or to the right 10°, scaling the extracted face region to 90% or 110%. Then we get another five face examples by horizontally mirroring all the five image obtained before. An original image and 10 images generated from it are shown in Figure 3.4.

Figure 3.4 10 images extracted from one face

The mean and variance of 4650 face images of size 38 × 38 pixels are shown in Figure 3.5.

(a) mean face (b) histogram-equalised mean face (c) variance

Figure 3.5 The mean and variance of 4650 face images

Figure 3.5a shows the mean of all the face images. Figure 3.5b is the histogram equalised mean face. Figure 3.5c shows the variance of each pixel. The bright part has high variance, while the dark part has low variance. From Figure 3.5c we can see that the centre of the mouth, eyebrows, and the sides of the nose have large variation, while the cheeks and nose ridge have small variation.

The race and age composition of those 4650 face images are shown in Figure 3.6.

Figure 3.6 The race and age composition of 4650 face images. (a) race composition, (b) age composition

Among those images, 20% are images of females.

3.1.4 Nonface image and anything-image preparation

Nonface images were collected from natural scenery images at different scales by template matching. Specifically, in these images any window with Euclidean distance from the histogram-equalised mean face (shown in Figure 3.5b) within a threshold was extracted and used as a non-face image. Examples of nonface images are shown in Figure 3.7.

Figure 3.7 Examples of 38 × 38 pixel nonface images

Another kind of image is called *anything-image*, which was also collected from natural scenery images but template matching was not used. They were obtained by arbitrarily dividing an image into $n \times n$ pixel blocks. Currently we have only collected 19 × 19 pixel anything-images, i.e., $n = 19$. Examples of anything-images are shown in

Figure 3.8. For us to see clearly, the observed size of all the images in Figure 3.8 is the double of their actual size.

Figure 3.8 Examples of 19 × 19 pixel anything-images

The preprocessing steps have been applied to all the nonface images and anything-images.

3.2 Feature Spaces

3.2.1 Original greyscale space

Images can be directly characterised in terms of pixel intensities. All the images that we deal with in this thesis are greyscale images, or black and white images. An image of size $m \times n$ is simply a matrix of 8-bit values with each element representing the intensity at that particular pixel. This image may also be viewed as a vector of length $m \times n$ or a single point in an $m \times n$ dimensional space. The construction of this vector from an image is performed by a simple concatenation - the rows of the image are placed each beside one another, as shown in Figure 3.9.

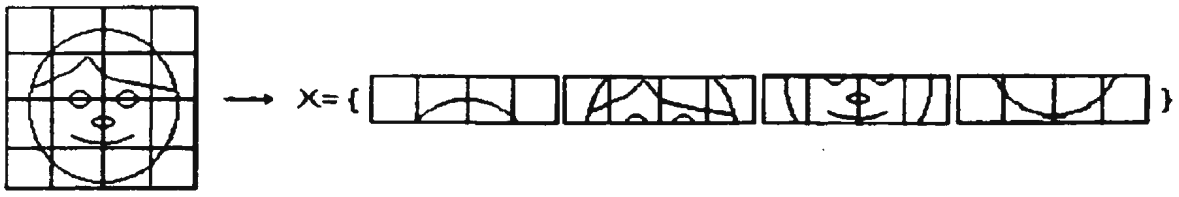


Figure 3.9 Formation of the face's vector from the face's image

For the images used in this study, after the preprocessing steps the resulting vector for a 19 × 19 pixel image is 337 dimensional, and for a 38 × 38 pixel image is 1360

dimensional. Normalisation is then performed to make the vector zero mean and unit variance. Note that this does not shift the relative locations of pictures in feature space very much because they are all previously histogram equalised. The image space to which an image vector described here belongs is termed *the original greyscale space.*

This original greyscale space may not be an optimal space for face description. The task presented here aims to build a feature space where all the face vectors are located in a very compact, convex cluster, as shown in Figure 3.10. Different people's faces, different expressions, different poses, etc., would be well separated within this space. Ideally, facial attributes like structure, pose and expression would be orthogonal.

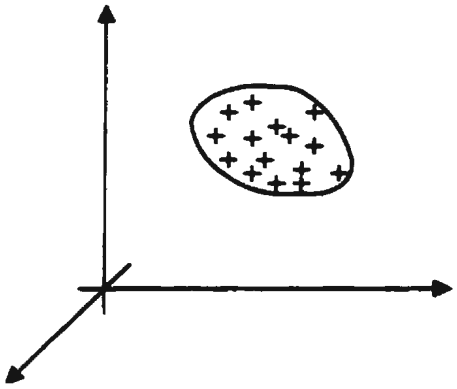


Figure 3.10 Ideal space and face cluster

In the following sections we describe or propose various feature spaces for fulfilling this task.

3.2.2 Orthogonal whitening process

The Karhunen-Loeve Transform (KLT) [Loève 1955], or Principal Component Analysis (PCA), is eigenvector-based technique that is commonly used for dimensionality reduction and feature extraction in pattern recognition. A low-dimensional

subspace that is composed of statistically uncorrelated variables is extracted. Classification is then performed in this eigenspace.

As described in Section 2.2.2, given a training set of images $\{\mathbf{x}_i\}_{i=1}^{N_T}$, $\mathbf{x} \in R^N$, where N is the dimensionality of the images in the original greyscale space and N_T is the number of images, the covariance matrix of training samples can be decomposed into $\Sigma = \Phi\Lambda\Phi^T$, where Φ contains the eigenvectors and Λ contains the eigenvalues λ_i, i = 1,..., N. In PCA, a partial KLT is performed to identify the largest-eigenvalue eigenvectors of the covariance matrix and obtain a principal component feature vector $\mathbf{y} = \Phi_M^T \tilde{\mathbf{x}}$, where $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{m}$ ($\mathbf{m}$ is the sample mean), and Φ_M is composed of the M largest eigenvalue eigenvectors.

The Mahalanobis distance in the original greyscale space is

$$d(\mathbf{x}) = \tilde{\mathbf{x}}^T \Sigma^{-1} \tilde{\mathbf{x}} \tag{3.5}$$

The calculation of Σ^{-1} can be simplified by using eigenvectors and eigenvalues

$$\begin{aligned} d(\mathbf{x}) &= \tilde{\mathbf{x}}^T \Sigma^{-1} \tilde{\mathbf{x}} \\ &= \tilde{\mathbf{x}}^T (\Phi\Lambda\Phi^T)^{-1} \tilde{\mathbf{x}} \\ &= \tilde{\mathbf{x}}^T (\Phi\Lambda^{-1}\Phi^T) \tilde{\mathbf{x}} \\ &= \mathbf{y}^T \Lambda^{-1} \mathbf{y} \end{aligned} \tag{3.6}$$

Because of the diagonalised form, the Mahalanobis distance can also be expressed as

$$d(\mathbf{x}) = \sum_{i=1}^{N} \frac{y_i^2}{\lambda_i} \tag{3.7}$$

where y_i is the i-th element of $\mathbf{y}$, the new vector obtained by the change of coordinates in a KLT.

In this new coordinate system, if $\mathbf{y}$ is divided by the eigenvalues $\Lambda^{\frac{1}{2}}$, the resulting vector $\mathbf{z} = \mathbf{y}\Lambda^{-\frac{1}{2}}$ makes the covariance matrix of the $\mathbf{z}$ vectors equal to unity. Therefore, this step is called "whitening" and the space is called whitened space. The Mahalanobis distance becomes

$$d(\mathbf{x}) = \sum_{i=1}^{N} z_i^2 \tag{3.8}$$

The Mahalanobis distance in the original space is equivalent to the Euclidean distance in the whitened space.

This whitening process can be applied to the anything-images, face images, or their combination to derive different feature spaces.

3.2.3 Anything-image-whitened space

Anything-image whitening is a frequency-based method that preserves the dimensions along which images differ most. Low frequency components of images are kept but the high frequency components are discarded. Anything-image whitening does not assume anything about the underlying probability density of face data, so it can be used by general-purpose schemes.

An anything-image-whitened space is obtained by applying the whitening algorithm to anything-images.

1) Collect many (we use 5999) anything-images. Normalisation is performed such that every image has zero mean and unit variance.
2) Do KLT on the anything-images to obtain eigenvalues $\mathbf{\Lambda}_a$ and eigenvectors $\mathbf{\Phi}_a$. If the number of dimensions of an image is N, we will get $N - 1$ valid eigenvectors. Take the largest K, $K \leq N-1$, eigenvalue eigenvectors and obtain the transformation

matrix $\mathbf{T}_a = \Lambda_a^{-\frac{1}{2}} \Phi_a^T$ which orthonormally whitens anything-images. Now $\Lambda_a \in R^{K \times K}, \Phi_a \in R^{N \times K}$ and thus $\mathbf{T}_a \in R^{K \times N}$. As shown in Figure 3.11 the eigenvectors with high variance (i.e. high eigenvalues) are low frequency, and the ones with low variance are high frequency. These form a set of basis images that resemble the filtering performed by some types of cells in the primary visual cortex.

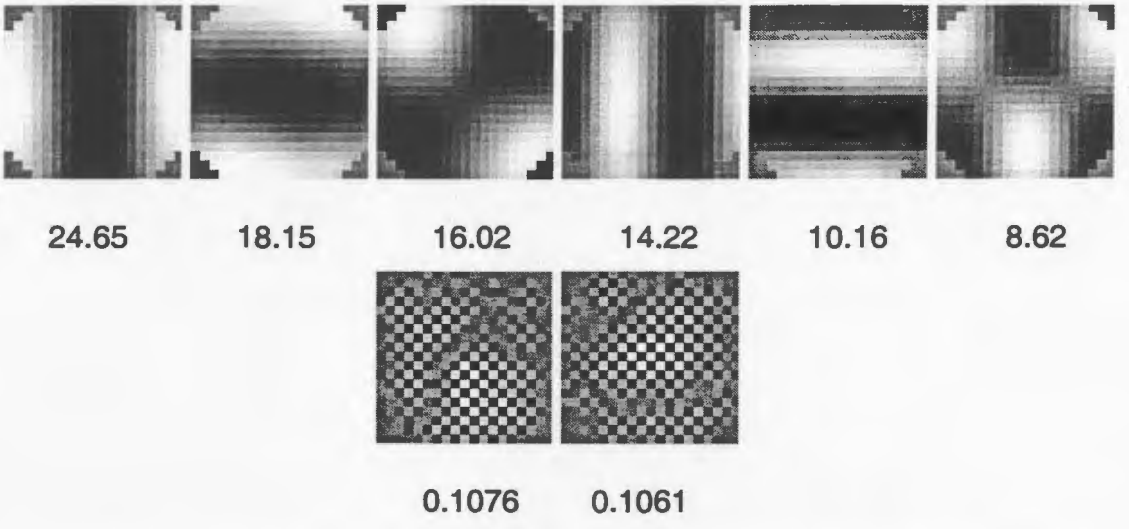


Figure 3.11 The largest 6 eigenvalue eigenvectors and the smallest 2 eigenvalue eigenvectors of anything-images. The eigenvalues are presented.

In Figure 3.12, we use a 2D space to illustrate the whitening process in a very high-dimensional space. The bigger ellipse or circle represents the distribution of the anything-image class ω_a, while the smaller one represents the distribution of the face class ω_f.

a) Figure 3.12a shows that in the original greyscale space, the eigenvectors of anything-images are $\{\phi_{ai}\}, i = 1,..., N-1$.

b) Then this space is rotated along Φ_a as shown in Figure 3.12b.

c) If the representations in this space are divided by $\Lambda_a^{\frac{1}{2}}$, we get an anything-image-equal-variance space as shown in Figure 3.12c. The variance of anything-images in any direction is one. Thus the anything-images have been whitened.

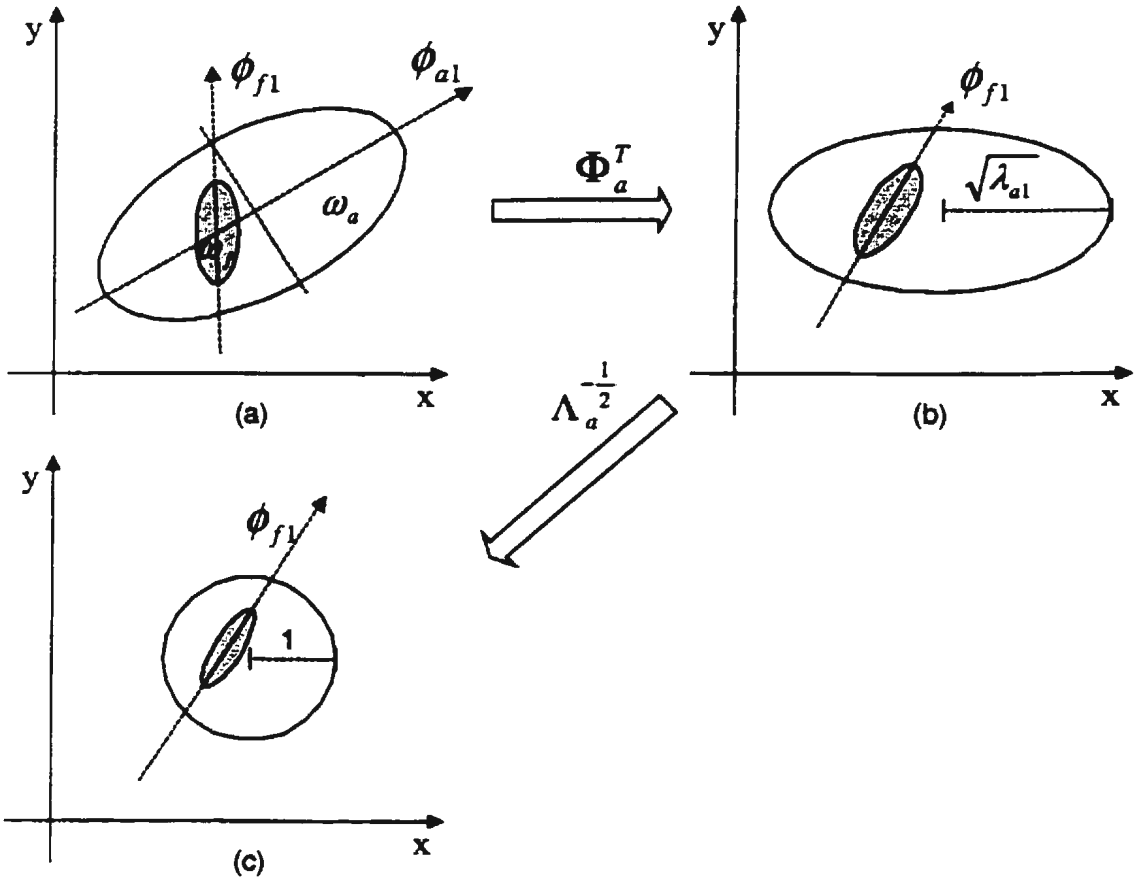
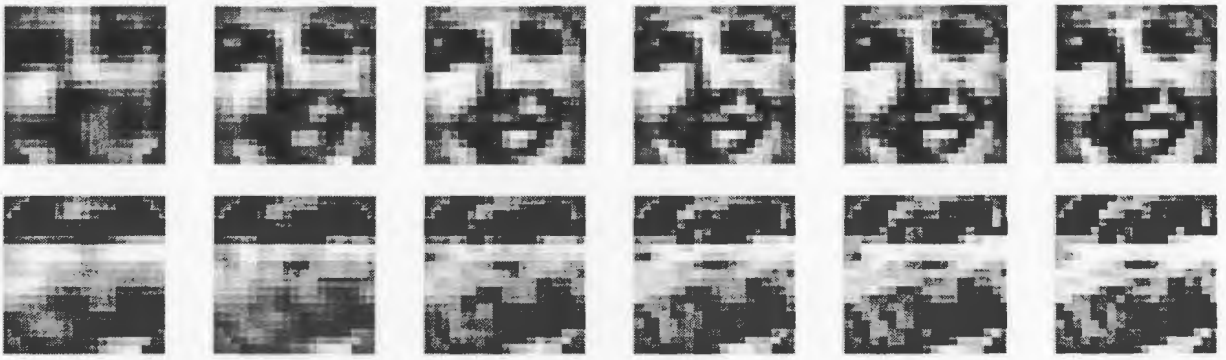


Figure 3.12 Anything-image-whitening scheme

Now we look at the reconstruction error. Because only the top K eigenvalue eigenvectors are used to construct the anything-image-whitened space, the high frequency components of a vector are discarded. Figure 3.13 shows the reconstructed images for a face image and a nonface image of size 19 × 19 pixels when K varies.



(a)50 (b)100 (c)150 (d)200 (e)250 (f)original

Figure 3.13 Reconstructed face images (first row) and nonface images (second row) using various number of largest eigenvalue eigenvectors of anything-images.

Whether the reconstructed image is a face can be reliably judged when more than K = 150 eigenvectors are used.

3.2.4 Face-image-whitened space

Face-image-whitening scheme is a specific purpose scheme used for face image processing.

A face-image-whitened space is obtained by applying the whitening algorithm to face images. The face-image-whitening process is as follows.

1) Collect many (we use 4556) face images. Normalisation is performed such that every image has zero mean and unit variance.

2) Do KLT on the face images to obtain eigenvalues Λ_f and eigenvectors Φ_f, take the largest M eigenvectors, and obtain the transformation matrix $\mathbf{T}_f = \Lambda_f^{-\frac{1}{2}} \Phi_f^T$, where $\Lambda_f \in R^{M \times M}, \Phi_f \in R^{N \times M}$ and thus $\mathbf{T}_f \in R^{M \times N}$. These eigenvectors compose a face-image-whitened space.

3) The projection of an image $\mathbf{x}$ into this space is $\mathbf{y} = \mathbf{T}_f \mathbf{x}$.

Figure 3.14 illustrates the process of face-image whitening in 2D. In the face-image-whitened space, the variance of face images along every dimension is one.

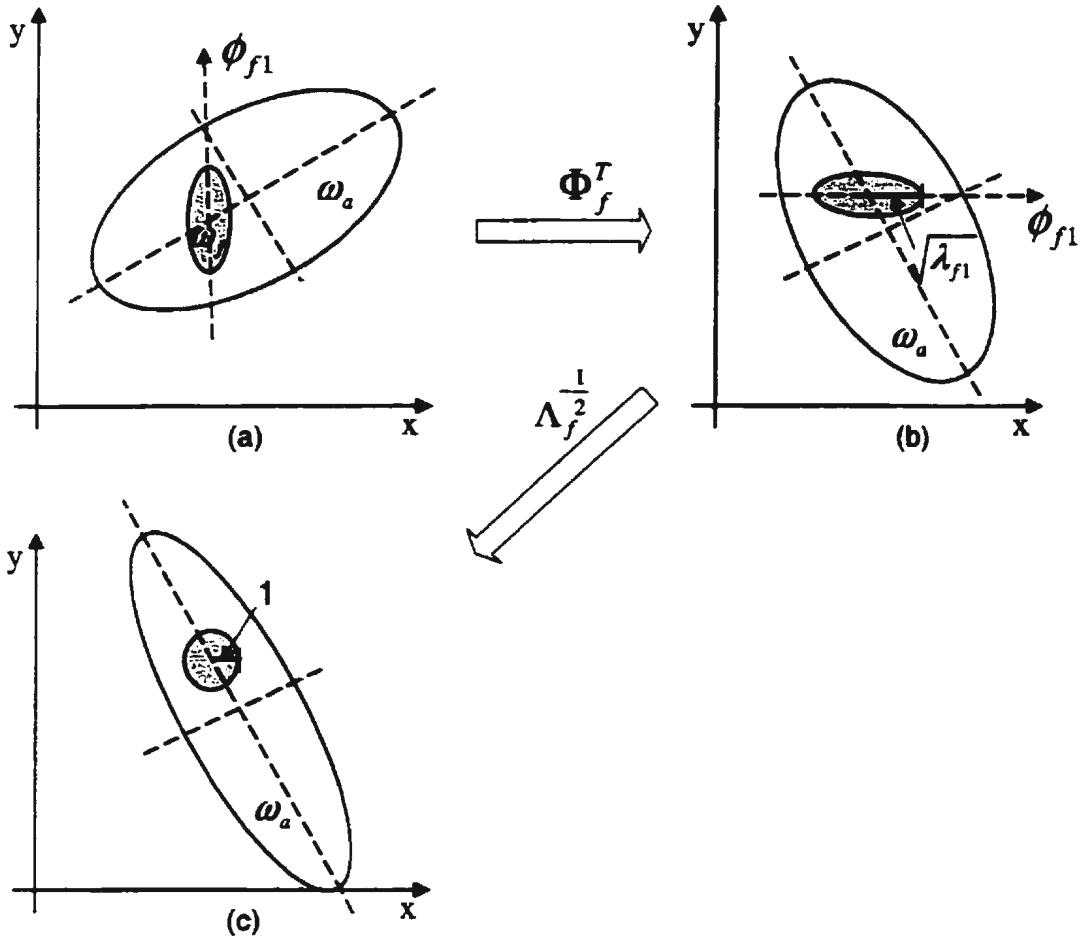
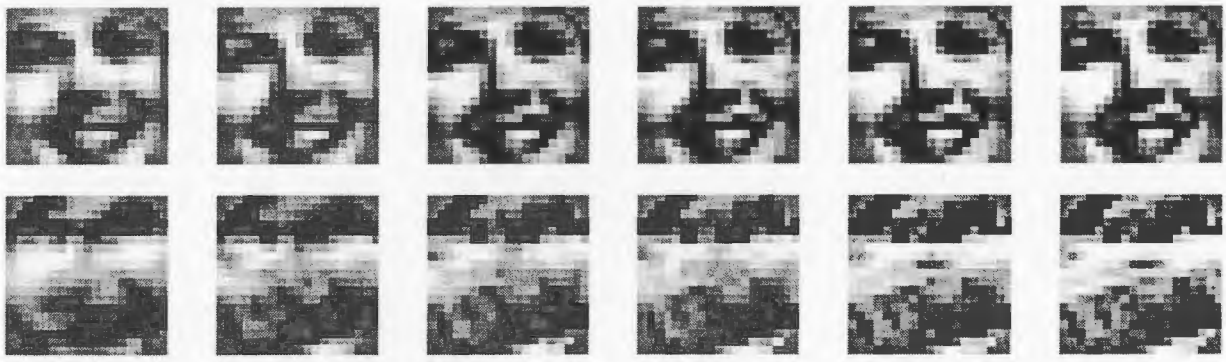


Figure 3.14 Face-image-whitening scheme



(a)50 (b)100 (c)150 (d)200 (e)250 (f)original

Figure 3.15 Reconstructed face images (first row) and nonface images (second row) using various number of eigenvectors of training faces

Figure 3.15 shows the reconstructed faces and nonfaces using different values of M, the number of eigenvectors to compose the principal component space. When M is greater than 100, it is clear that the reconstructed image is a face. Compared with Figure 3.13, Figure 3.15 shows that fewer eigenvectors are required to construct a good quality face.

3.2.5 Double-whitened space

A double-whitened space is obtained by performing face-image whitening after anything-image whitening. More specifically, anything-image whitening is first performed and a K-dimensional anything-image-whitened space is obtained. Then face-image whitening is performed based on the images in this anything-image-whitened space and an M-dimensional double-whitened space is obtained. $M \leq K$.

Figure 3.16 illustrates the face-image whitening after the anything-image whitening depicted in Figure 3.12.

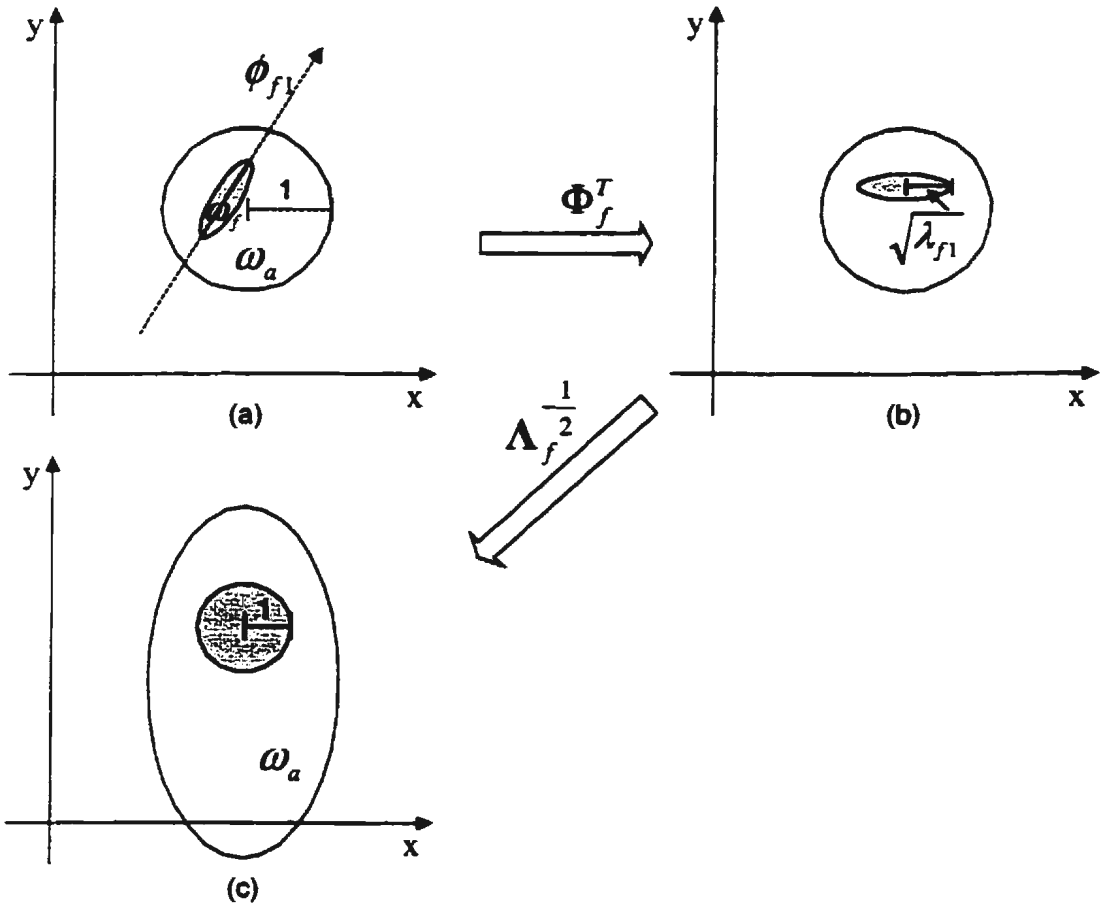


Figure 3.16 Face-image whitening after the anything-image whitening

Figure 3.16a is identical to Figure 3.12c, which represents the anything-image-whitened space. Figure 3.16c shows that the double-whitening process derives a face-equal-variance space.

The difference between this and the face-image-whitened space is the elimination of features at the anything-image-whitening stage.

3.2.6 Analysis of the eigenvectors of faces

If the image size in the original space is N, a maximum of $N - 1$ eigenvectors of anything-images are obtained. We select the largest K eigenvalue eigenvectors of them to generate an anything-image-whitened space.

In the anything-image-whitened space, the eigenvectors of faces are found. Figure 3.17 shows the histogram of the projections of faces and anything-images onto the smallest eigenvalue eigenvector of faces. $N = 337$ and $K = N - 1$.

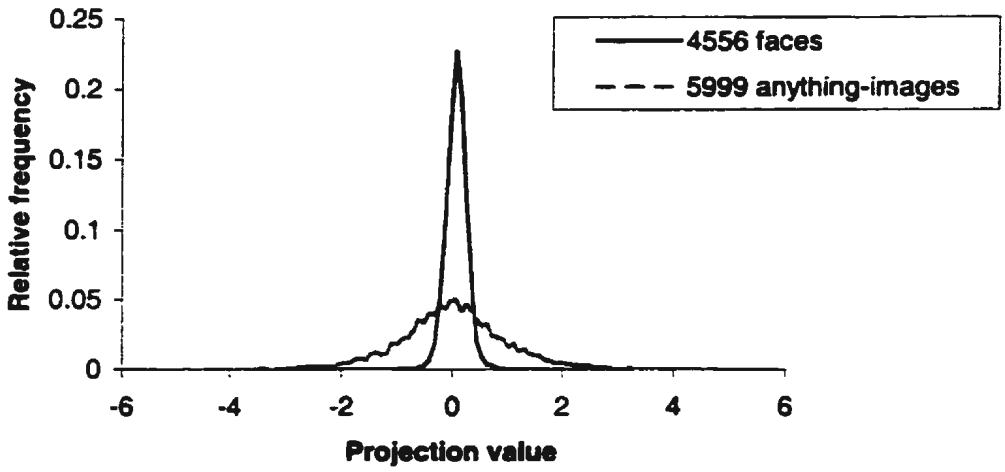


Figure 3.17 In the anything-image-whitened space, the projection onto the smallest eigenvalue eigenvector of faces

Every curve of Figure 3.17 is plotted as follows:

- In the horizontal axis, we divide the range from the minimum projection value to the maximum projection value into 100 intervals.
- In each interval, the number of instances is counted.

- In the vertical axis corresponding to the midpoint of each interval, we mark "Relative frequency", defined as $\frac{\text{Number of instances}}{\text{Total number of samples}}$. Using "Relative frequency" can help us more objectively see the histogram of two distributions that have quite different numbers of samples.

This plotting method of showing distribution is used throughout the thesis.

Figure 3.17 shows that the distribution of face images is tight. The variance of anything-images is preserved to be one.

As an example, Figure 3.18 shows the projection onto the 11-th smallest eigenvalue eigenvector of faces.

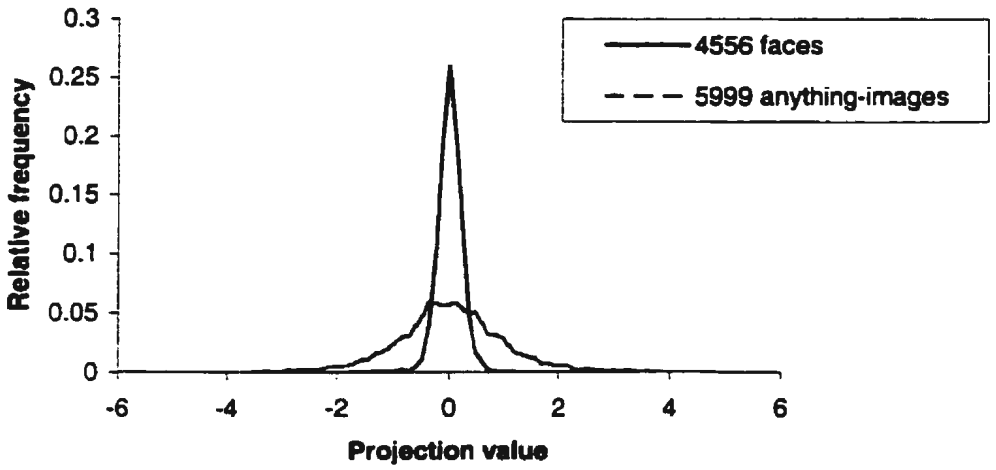


Figure 3.18 In the anything-image-whitened space, the projection onto the 11th smallest eigenvalue eigenvector of faces

Not much difference between Figure 3.17 and Figure 3.18 is observed.

Figure 3.19 shows the projection onto the largest eigenvalue eigenvector of faces.

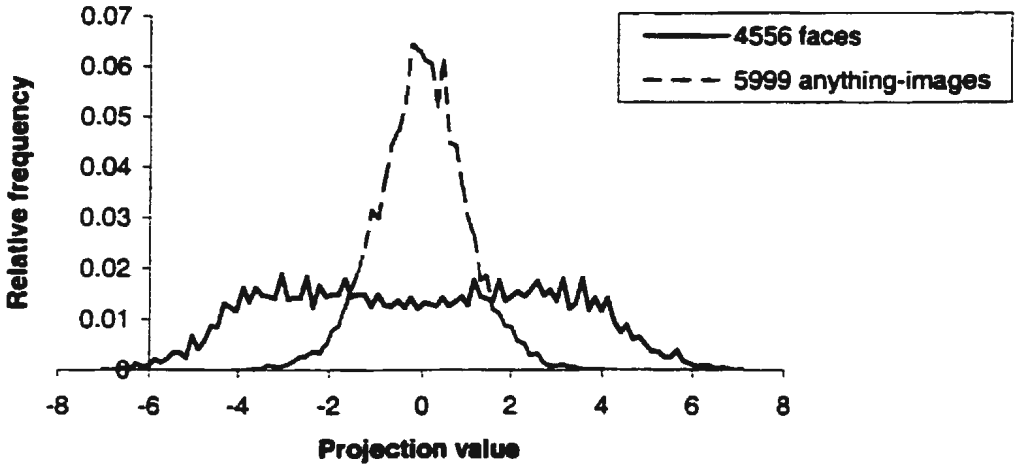


Figure 3.19 In the anything-image-whitened space, the projection onto the largest eigenvalue eigenvector of faces

In Figure 3.19 the distribution of faces spreads out, but that of anything-images keeps the same shape as those in Figure 3.17 and Figure 3.18. The distribution of face images is bimodal, but the distributions in the other two figures are unimodal.

Because the smallest eigenvectors of faces represent directions in which the variance of face space divided by the variance of anything-image space is at a minimum, we can derive a distance measure based on an image's projection along these directions. This distance measure will be described and implemented in Section 4.11.1.

3.3 Faces versus Nonfaces

In the previous section, a set of 5999 anything-images and a set of 4556 face images have been used to generate the whitened spaces. In this section we analyse these two sets, along with a set of 3286 nonface images that will be used as the training set for the classification experiments in next chapter. We do the analysis in both the original greyscale space and the anything-image-whitened space.

3.3.1 In the original greyscale space

We perform a separate PCA on these three sets. Figure 3.20 shows the top 100 eigenvalues of each set. When the serial number is greater than 10, the trail of the anything-image eigenvalue curve is above those of the nonfaces and faces. This verifies that there is more diversity in the anything-image set.

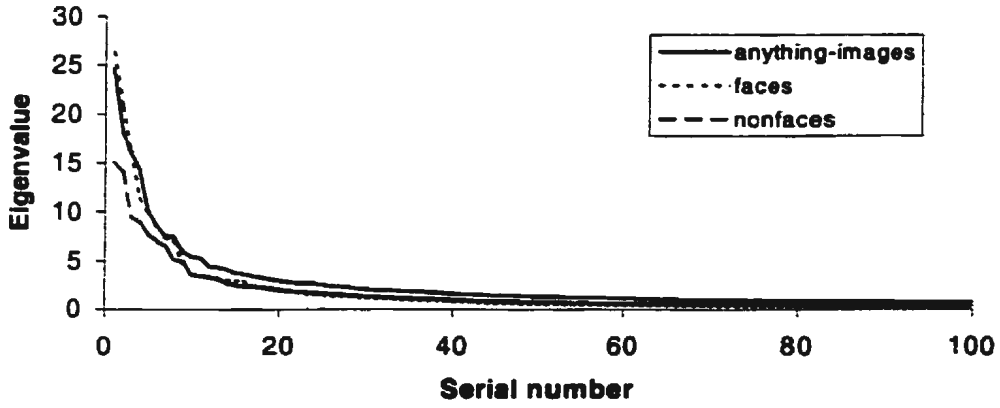


Figure 3.20 Top 100 eigenvalues of each set in the original space

An interesting point is how we choose K, the number of eigenvectors, to compose the principal space. According to literature [Sung 1998, Swets 1996], there are two methods to get a suitable K.

Method A:

K is chosen such that the sum of these unused eigenvalues is less than some fixed percentage of the sum of the entire set. So we let K satisfy

$$\left(\sum_{i=K+1}^{N}\lambda_i\right)\Big/\left(\sum_{i=1}^{K}\lambda_i\right) < 5\% \qquad (3.9)$$

Method B:

K is chosen such that $\sum_{i=K+1}^{N}\lambda_i$ is as close to the largest eigenvalue λ_1 as possible.

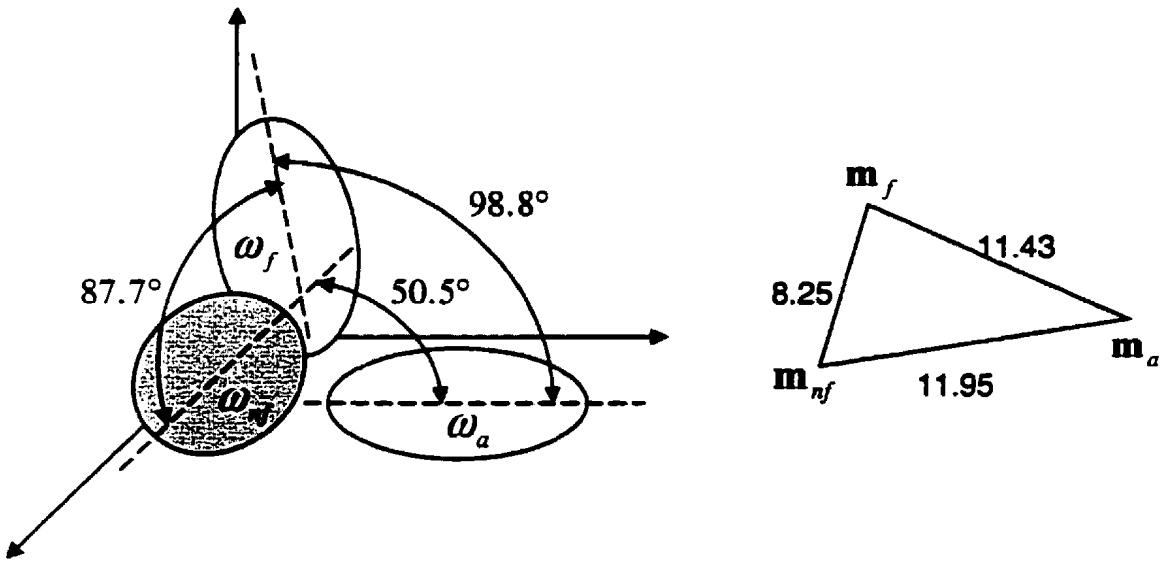
These two methods were applied to the face set, nonface set, and anything-image set. The results presented in Table 3.1 tells that the within-set variation increases in the order of face set, nonface set, and anything-image set. More than 200 eigenvectors of anything-images are necessary.

Table 3.1 Desired number of eigenvectors in face, nonface, and anything-image set

	Face set	Nonface set	Anything - image set
Method A	135	193	247
Method B	63	156	214

A hyperellipsoid distribution is assumed for all the three classes. To consider how they are distributed, we calculate their relative orientation that is determined by computing the product of their respective first eigenvectors. This analysis yields the cosine of the angle between the major axes of the each pair of hyperellipsoids.

In the original greyscale space, the hyperellipsoid distribution of the face class ω_f, nonface class ω_{nf}, and the anything-images ω_a is illustrated in Figure 3.21. The scale of the hyperellipsoids approximates the corresponding eigenvalues. The largest eigenvalue of the nonface class is smaller than the largest eigenvalue of the face class.



(a)The angles between the main axes of hyperellipsoids

(b)The distances between the means of each class

Figure 3.21 Illustration of the hyperellipsoid distributions of the face class, nonface class, and anything-image class in the original greyscale space

The Euclidean distances between the means of each class are shown in Figure 3.21b. $\mathbf{m}_a$, $\mathbf{m}_f$, and $\mathbf{m}_{nf}$ are the means of the anything-image class, face class, and nonface class respectively. The distance from $\mathbf{m}_a$ to $\mathbf{m}_f$ is almost equal to that from $\mathbf{m}_a$ to $\mathbf{m}_{nf}$.

3.3.2 In the anything-image-whitened space

In the 100-dimensional anything-image-whitened space (K = 100), we get the projection of the faces and nonfaces. Then PCA is performed on these two sets separately in order to estimate their distribution. Figure 3.22 illustrates the hyperellipsoid distribution of the face class ω_f and the nonface class ω_{nf} in this space.

The angle between the major axes of the face distribution and the nonface distribution is found to be 92.2°.

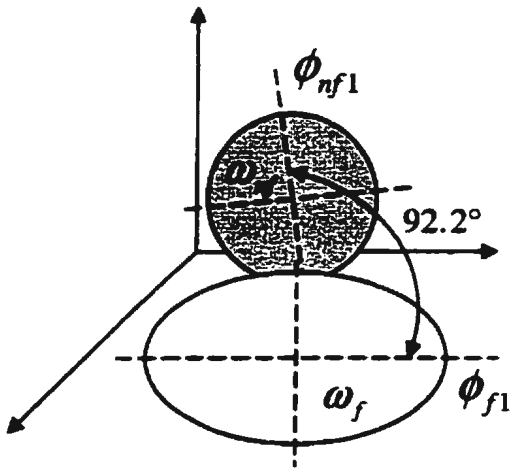


Figure 3.22 Illustration of the hyperellipsoid distributions of face class and nonface class in the anything-image-whitened space

3.4 Summary

This chapter presents two cornerstones of this thesis: data preparation and feature spaces.

The face region in a picture containing a face is decided by first manually marking six feature points: two eyes, nose tip, two corners and centre of mouth. Then this picture is rotated, translated, and scaled in order to generate face images with desired size and orientation. Ten face images with slightly different orientation angles and scale are obtained from one picture.

Nonface images were collected from natural scenery images at different scales by template matching. Anything-images were obtained by randomly extracting a part of scenery images without template matching.

The preprocessing steps: masking, shade removal, and histogram equalisation, are performed on all the images.

The image space where a normalised greyscale image resides is called the original greyscale space. Three feature spaces are proposed for face image processing.

- Anything-image-whitened space

  PCA is performed on a set of anything-images. The top K eigenvectors of anything-images are used to compose this space. The variance along each eigenvector is normalised to unity.

- Face-image-whitened space

  It is obtained in the same way as the anything-image-whitened space except that the face images, instead of the anything-images, are used.

- Double-whitened space

  It is obtained by performing face-image whitening after anything-image whitening.

The hyperellipsoid distribution of the face class, nonface, and the anything-image class is analysed in the original greyscale space and the anything-image-whitened space.

Chapter 4

Face/nonface Classification

In this chapter we systematically compare the performance of the FLD, the probabilistic classifiers, and the nearest neighbour classifiers. Two new classifiers, one feature extraction technique, and one data-modelling technique are proposed. The classifiers are described immediately prior to the corresponding experimental results to avoid repetition.

These classifiers are tested on face/nonface classification in different feature spaces. All the available samples are split into training and test sets. The numbers of images in training and test sets are listed in Table 4.1.

Table 4.1 Number of images in the data sets

Data set	Number of images
Training face set	4556
Training nonface set	3286
Test face set	1130
Test nonface set	2553

The method of generating these images is described in Section 3.1. The images in the training sets are distinct from those in the test sets. Every image is of size 19×19 pixels. After removing the corner pixels, every image becomes a 337-dimensional vector

recording pixel greyscales. These data sets are used throughout all the experiments to give an objective comparison between different schemes. The classifier is first designed using training samples, and then it is evaluated based on its classification performance on the test samples. An error rate is estimated as the number of misclassified images divided by the total number of images in a set. Four error rates are used.

e_1: error rate in the training face set;

e_2: error rate in the training nonface set;

e_3: error rate in the test face set;

e_4: error rate in the test nonface set.

4.1 Fisher's Linear Discriminant (FLD)

Fisher's Linear Discriminant (FLD) is a supervised learning procedure that projects the images into a subspace that maximises the between-class scatter while minimizing the within-class scatter of the projected data. This approach assumes linear separability of the classes. The dimensionality reduction is performed as follows.

We consider a set of sample images $\{\mathbf{x}_i\}_{i=1}^{N_T}$ taking values in an N-dimensional image space, and assume that each image belongs to one of c classes $\{\omega_1, \omega_2, ..., \omega_c\}$.

Let the between-class scatter matrix be defined as

$$\mathbf{S}_B = \sum_{i=1}^{c} (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T \qquad (4.1)$$

and the within-class scatter matrix be defined as

$$\mathbf{S}_W = \sum_{i=1}^{c} \frac{1}{n_i} \sum_{\mathbf{x}_k \in \omega_i} (\mathbf{x}_k - \mathbf{m}_i)(\mathbf{x}_k - \mathbf{m}_i)^T \tag{4.2}$$

where $\mathbf{m}_i$ is the mean of class ω_i, $\mathbf{m}$ is the total mean of all the class means, and n_i is the number of samples in class ω_i. When $\mathbf{S}_w$ is nonsingular, the optimal projection is the eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$, denoted as $\mathbf{W}$. Note also that there are at most $c - 1$ nonzero eigenvalues and thus only $c - 1$ valid eigenvectors [Jain 1988]. These eigenvectors maximise the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples. Finally a given image $\mathbf{x}$ is classified by projecting $\mathbf{x}$ into the subspace. The discriminant function is

$$d(\mathbf{x}) = \mathbf{W}^T(\mathbf{x} - \mathbf{m}) = \mathbf{W}^T\mathbf{x} - w_o \tag{4.3}$$

where $w_o = \mathbf{W}^T\mathbf{m}$ is the separation vector between classes.

In a two-class classification case, there is only one valid eigenvector of $\mathbf{S}_W^{-1}\mathbf{S}_B$. Therefore, $\mathbf{W}$ becomes a vector, called the "Fisher vector".

Although $\mathbf{S}_W$ and $\mathbf{S}_B$ are symmetric matrices, $\mathbf{S}_W^{-1}\mathbf{S}_B$ may not be symmetric, the eigensystem calculation could be unstable. A method in [Swets 1996] was adopted and slightly modified to solve this problem.

Compute the eigenvectors $\mathbf{H}$ and eigenvalues $\mathbf{\Lambda}$ of $\mathbf{S}_W$. If a zero eigenvalue presents in $\mathbf{\Lambda}$, the eigenvector associated with the zero eigenvalue is removed from $\mathbf{H}$. Compute the eigenvectors $\mathbf{U}$ and eigenvalues $\mathbf{\Sigma}$ of $(\mathbf{H}\mathbf{\Lambda}^{-\frac{1}{2}})^T\mathbf{S}_B\mathbf{H}\mathbf{\Lambda}^{-\frac{1}{2}}$, which is a symmetric matrix. Then the eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$ are in $\mathbf{\Delta} = \mathbf{H}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}$ and the eigenvalues are in $\mathbf{\Sigma}$.

The reason is that $\mathbf{S}_W = \mathbf{H}\Lambda\Lambda^T$ gives $\mathbf{S}_W^{-1} = \mathbf{H}\Lambda^{-1}\mathbf{H}^T$, and $(\mathbf{H}\Lambda^{-\frac{1}{2}})^T \mathbf{S}_B \mathbf{H}\Lambda^{-\frac{1}{2}} = \mathbf{U}\Sigma\Sigma^T$ gives $\mathbf{S}_B = \mathbf{H}\Lambda^{\frac{1}{2}}\mathbf{U}\Sigma\Sigma^T\Lambda^{\frac{1}{2}}\mathbf{H}$, so that

$$\mathbf{S}_W^{-1}\mathbf{S}_B = \mathbf{H}\Lambda^{-1}\mathbf{H}^T\mathbf{H}\Lambda^{\frac{1}{2}}\mathbf{U}\Sigma\Sigma^T\Lambda^{\frac{1}{2}}\mathbf{H}^T$$

$$= \mathbf{H}\Lambda^{-\frac{1}{2}}\mathbf{U}\Sigma\Sigma^T\Lambda^{\frac{1}{2}}\mathbf{H}^T$$

$$= \Delta\Sigma\Delta^{-1}$$

The last equation indicates that Δ and Σ contain the eigenvectors and the eigenvalues of $\mathbf{S}_W^{-1}\mathbf{S}_B$.

We first find the Fisher vector W and the separation point w_0 according to the training sets, then project the test sets onto this Fisher vector, and thus get the error rates in the test sets.

This process is performed in the original greyscale space, the face-image-whitened space, the anything-image-whitened space, and the double-whitened space.

- In the original greyscale space

  The error rate of misclassified faces, e_3, is 18/1130 = 1.59%. That of misclassified nonfaces, e_4, is 90/2553 = 3.53%. The reason why the error rate of misclassifying nonfaces is higher is that with low resolution the test nonfaces are close to the face template in the Euclidean distance, i.e., they look like faces. From the total 1064469 test windows of a 733 × 495 pixel natural scenery image, only about 200 nonfaces, less than 0.019% were extracted for training or testing.

- In the anything-image-whitened space

  The error rates versus the dimensionality of the anything-image-whitened space are shown in Figure 4.1.

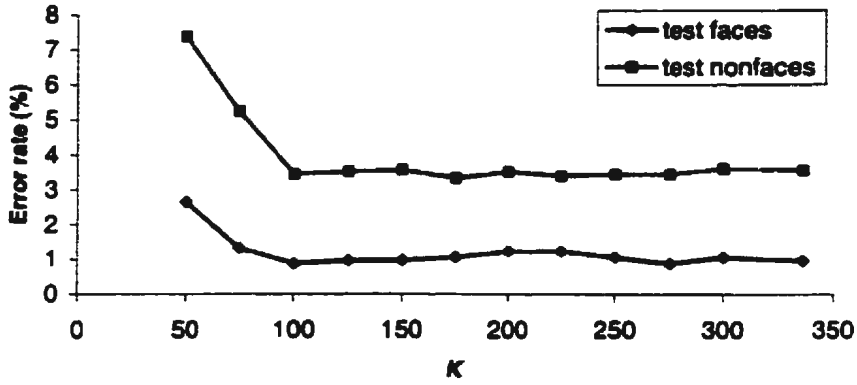


Figure 4.1 Using the FLD, the error rates versus the dimensionality of the anything-image-whitened space

In Figure 4.1, K is the number of dimensions of the anything-image-whitened space. The two curves are nearly flat after K = 100. This tells us that K does not matter much. The lowest error rates are 10/1130 = 0.88%, and 88/2553 = 3.45% at K = 100.

- In the face-image-whitened space

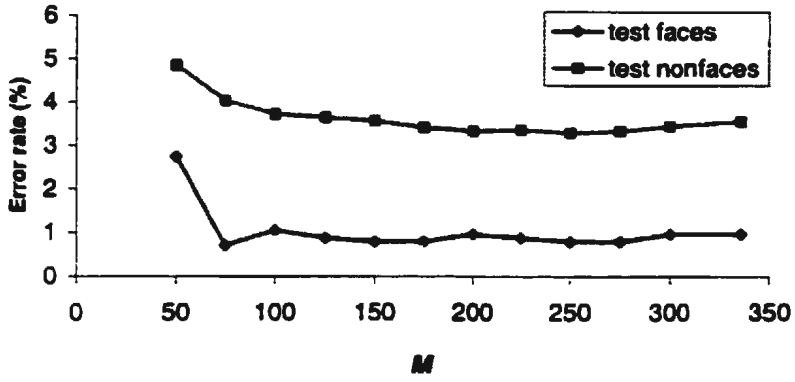


Figure 4.2 Using the FLD, the error rates versus the dimensionality of the face-image-whitened space

In Figure 4.2, M is the largest eigenvalue eigenvectors selected for composing this face-image-whitened space. When $M = 250$, we get the lowest error rates of 9/1130 = 0.80%, and 84/2553 = 3.29% for faces and nonfaces respectively. Figure 4.2 shows that the error rates do not respond sensitively to the change in M.

- In the double-whitened space

  If $K = 250$, but M varies from 50 to 250, the results are shown in Figure 4.3.

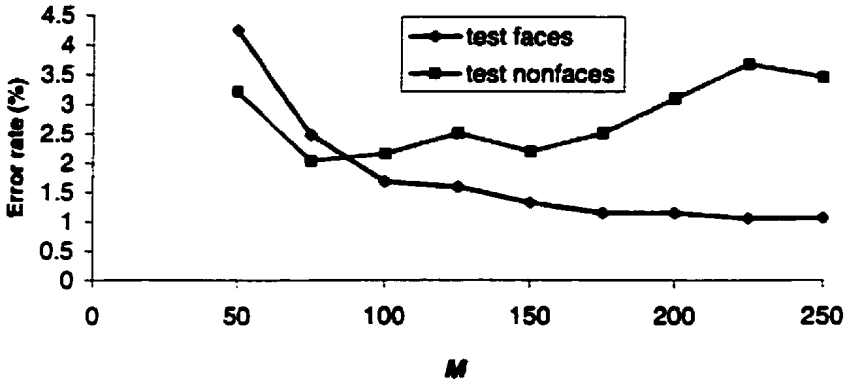


Figure 4.3 Using the FLD, the error rates versus the dimensionality of the double-whitened space ($K = 250$)

At the point $M = 150$, we get the smallest sum of face misclassification rate, 1.33%, and the nonface misclassification rate, 2.19%. Unlike the curves in Figure 4.1 and Figure 4.2, the tails of curves in Figure 4.3 change in response to the increase in dimensionality.

It is observed from the experimental results of the FLD that the dimensionality reduction does provide lower error rates. The FLD performs nearly equally well in the face-image-whitened space and the anything-image-whitened space. However, the lowest error rates are obtained in the double-whitened space.

4.2 Repeated FLD

We propose a repeated FLD scheme that obtains a group of Fisher vectors, W_1, W_2, ..., W_n. This group of Fisher vectors are applied to classification one by one. In the face/nonface classification case, a test pattern is classified as a face only when all the Fisher vectors label it as a true face. The process of using a group of Fisher vectors in face detection is shown in Figure 4.4.

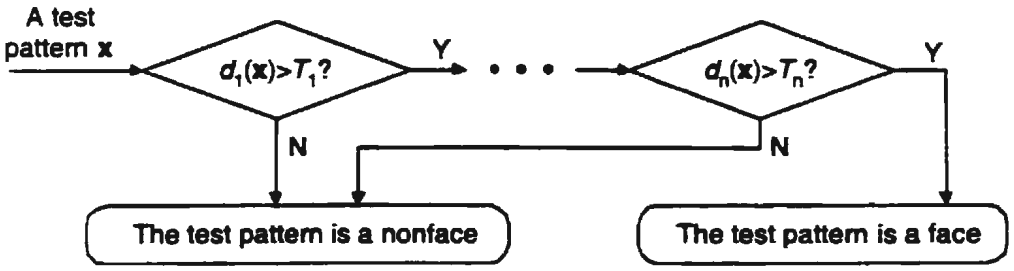


Figure 4.4 The process of applying a group of Fisher vectors to face detection

We project a test pattern $\mathbf{x}$ onto the first Fisher vector W_1, and compare the projection value $d_1(\mathbf{x}) = W_1^T\mathbf{x} - w_0$ with a threshold T_1. If it is less than T_1, we declare the test pattern as a nonface and stop. Otherwise, we project it onto the second Fisher vector W_2, and compare the projection value with another threshold T_2. This process is repeated until the test pattern has been projected to all the Fisher vectors. If it is always regarded as a face, the final decision is that this test pattern is a face.

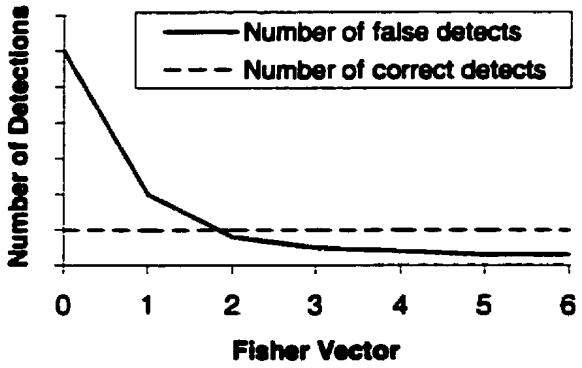


Figure 4.5 Ideal situation when a group of Fisher vectors are sequentially applied to face detection

Figure 4.5 shows the ideal situation when a group of Fisher vectors is used sequentially in face detection. The horizontal axis is the number of Fisher vectors used. It shows that the number of false detects should decrease when more Fisher vectors are used, but the number of valid faces found should remain the same. Each nonface that could not be eliminated by the first Fisher vector should be eliminated later. Therefore this scheme can progressively reject some kind of nonfaces while maintaining the real faces.

Two methods are developed to obtain a group of Fisher vectors.

4.2.1 Reducing the training samples

Changing the training samples usually would change the direction of the Fisher vector. Gradually reducing the training samples is a way of achieving this. The whole process is as follows.

1) Assume class ω_A contains samples $\{\mathbf{x}_i\}_{i=1}^{n_A}$, and class ω_B contains samples $\{\mathbf{y}_j\}_{j=1}^{n_B}$. Perform the FLD. The Fisher vector between them is W and the separation point is w_0.

2) Project all the samples onto W; The projection of a sample $\mathbf{x}$ onto W is

$$d(\mathbf{x}) = W^T\mathbf{x} + w_0$$

3) The mean of the projections in each class is

$$m_A = \frac{1}{n_A}\sum_{i=1}^{n_A} d(\mathbf{x}_i)$$

$$m_B = \frac{1}{n_B}\sum_{j=1}^{n_B} d(\mathbf{y}_j)$$

The direction of W is chosen such that $m_A > m_B$

4) Set a positive threshold θ.

5) Select samples $\mathbf{x}_i$ in class ω_A satisfying $d(\mathbf{x}_i) \le w_0 + \theta$. Likewise, select samples $\mathbf{y}_j$ in class ω_B satisfying $d(\mathbf{y}_j) \ge w_0 - \theta$. The selected samples compose two new data sets.

6) Repeat steps 1) to 5) until the desired number of Fisher vectors are obtained or the within class scatter matrix S_W becomes singular.

The Fisher vector obtained in the first round is denoted as W_1, and so on.

This implementation puts more emphasis on the face-like nonfaces and nonface-like faces. Because the number of training samples dwindles gradually, the within class scatter matrix will become singular after several loops.

If we set the threshold $\theta = (m_A - m_B)/2$, after each round of iteration, about half of the samples in the training set are used in the next round.

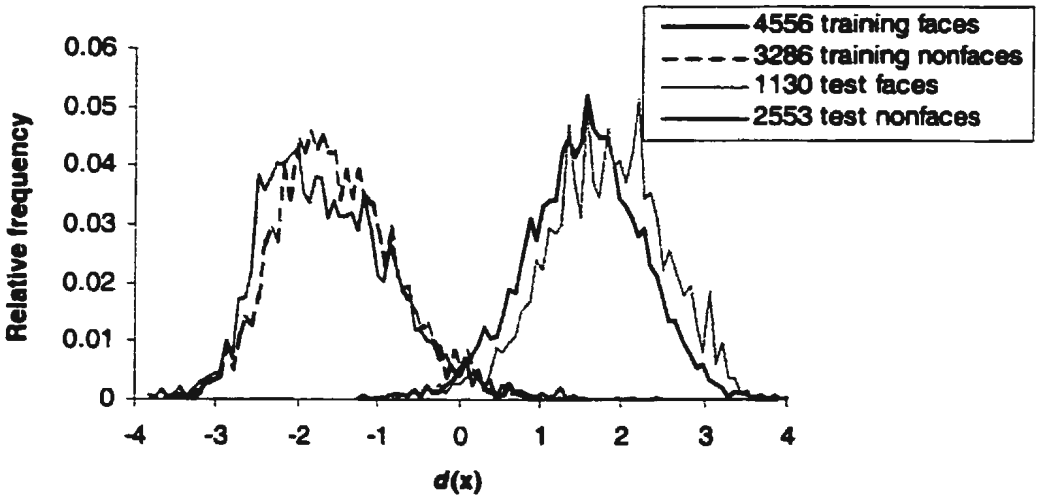
In a 150-dimensional anything-image-whitened space we applied this scheme to face/nonface classification.

Table 4.2 shows the number of training samples used in each round.

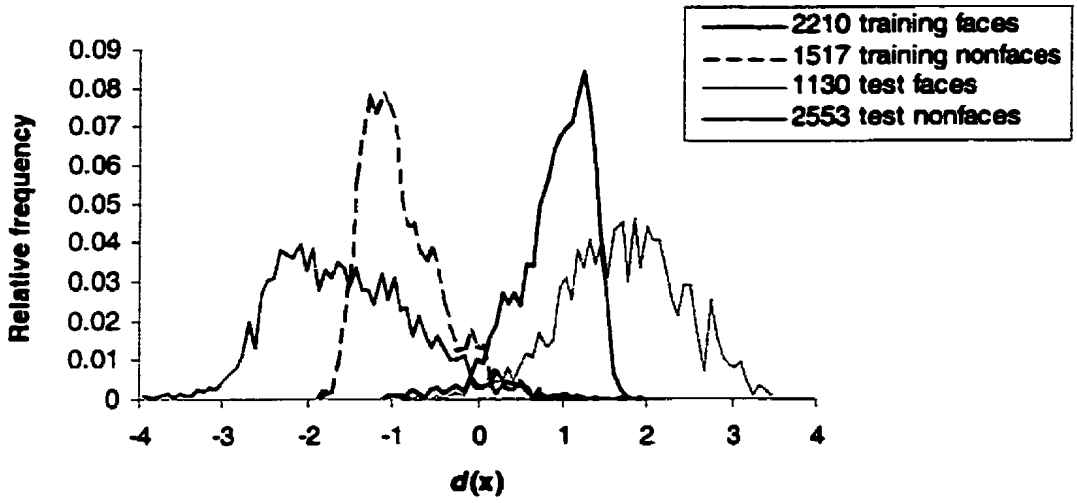
Table 4.2 Number of training samples used in repeated FLD scheme

	Number of faces	Number of nonfaces
Round 1	4556	3286
Round 2	2210	1517
Round 3	937	619
Round 4	398	250

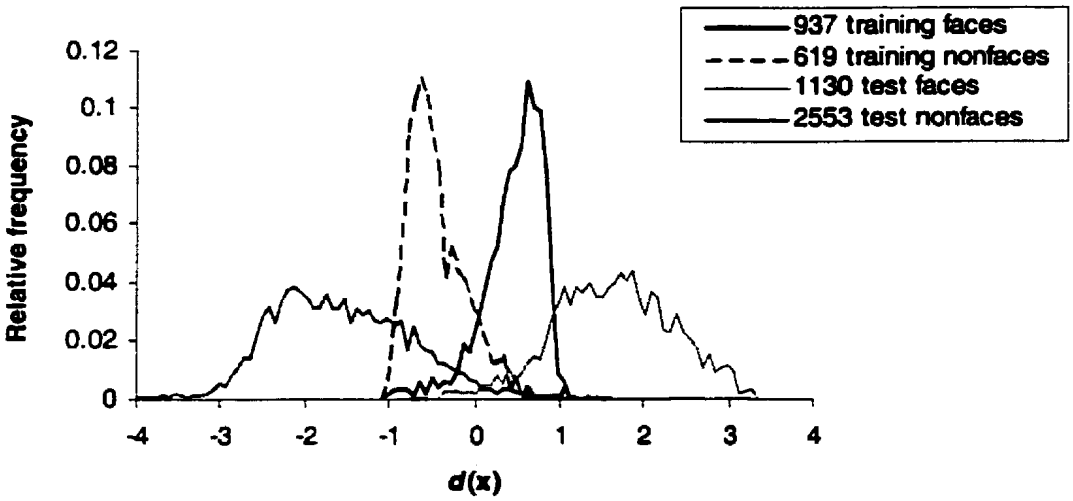
Figure 4.6 shows the projection of training and test sets onto four Fisher vectors obtained by using the repeated FLD scheme.



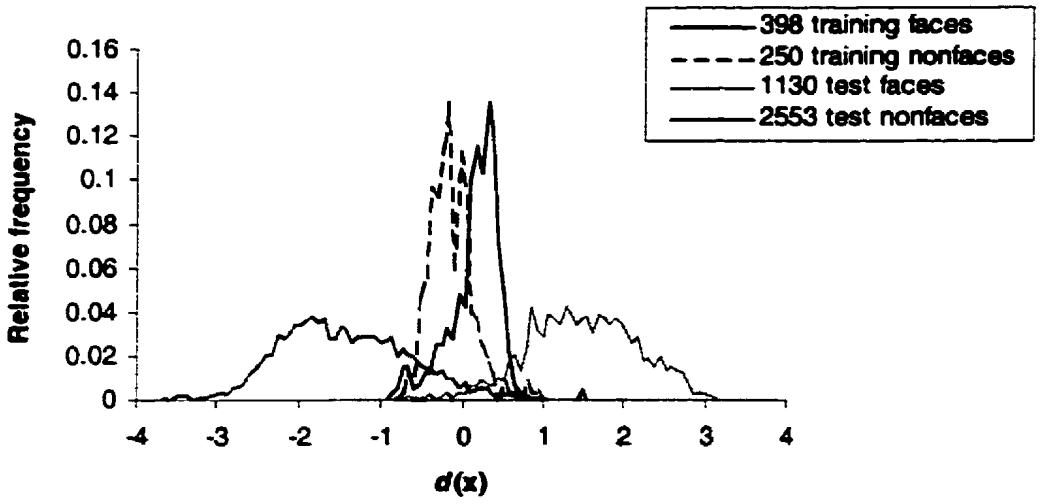
(a) Projection onto W_1



(b) Projection onto W_2



(c) Projection onto W_3



(d) Projection onto W_4

Figure 4.6 The projection of images onto the Fisher vectors obtained from the repeated FLD scheme

We can see that the training face and training nonface sets become tighter and closer together after every round. The reason that the two training classes get closer and closer is that the FLD tries to maximise the difference between two class-means in the projected space. However, after each round we are left with the "nonface-look" faces and "face-look" nonfaces. There is not much difference between these two class means.

It is worthwhile to look at the angles between every pair of Fisher vectors as presented in Table 4.3.

Table 4.3 Angles between a pair of Fisher vectors using repeated FLD scheme

	W_1	W_2	W_3	W_4
W_1		8.7°	15.2°	24.7°
W_2			8.3°	19.6°
W_3				14.5°

Table 4.3 shows that the discriminant hyperplanes are almost parallel to each other.

If the separation point is zero in the horizontal axis, the number of misclassified test images due to each Fisher vector is presented in Table 4.4.

Table 4.4 Number of misclassified images when the Fisher vectors are applied separately

	Misclassified nonfaces	Misclassified faces
W_1	91	11
W_2	81	13
W_3	85	18
W_4	112	21

In Table 4.4 these four Fisher vectors are applied to classification individually.

If the four Fisher vectors were applied to classification sequentially, the results would be different. It is important to select the thresholds so that they make all the faces detected and make as many nonfaces eliminated as possible. Using the four Fisher vectors whose discriminating ability are shown in Figure 4.6, we set the thresholds T_1, T_2, T_3 and T_4 to −1.203, −1.083, −1.042, and −0.88 respectively to make all the test faces correctly classified. The process shown in Figure 4.4 gives the number of misclassifications in the test sets as shown in Table 4.5.

Table 4.5 Number of misclassified images when the Fisher vectors are applied sequentially ($T_1 = -1.203$, $T_2 = -1.083$, $T_3 = -1.042$, and $T_4 = -0.88$)

	Misclassified nonfaces	Misclassified faces
Using W_1	739	0
Using W_1 and W_2	669	0
Using W_1 ,W_2 and W_3	634	0
Using W_1 ,W_2, W_3 and W_4	551	0

In Table 4.5 although we keep number of misclassified faces to zero, the number of misclassified nonfaces are too large to be acceptable. Therefore, we adjust the thresholds and get another group of results as shown in Table 4.6.

Table 4.6 Number of misclassified images when the Fisher vectors are applied sequentially ($T_1 = T_2 = T_3 = T_4 = 0$)

	Misclassified nonfaces	Misclassified faces
Using W_1	91	11
Using W_1 and W_2	76	13
Using W_1 ,W_2 and W_3	71	18
Using W_1 ,W_2, W_3 and W_4	68	22

Comparing Table 4.4 with Table 4.6, we can see that the results in Table 4.6 are slightly better. However, difficult nonfaces remain difficult for all the four Fisher vectors.

Besides gradually reducing the training samples, we can apply different sets of training nonface samples in order to change the direction of Fisher vector. For example,

after getting the first Fisher vector, we apply it to face detection and generate a set of false detects. These false detects are used as the training nonfaces and thus another Fisher vector is obtained. By repeating the process, we get a group of Fisher vectors. In face finding, this method does eliminate some false detects. However, the training faces and nonfaces become inseparable after several rounds using the FLD.

4.2.2 Rotate-coordinate-system-and-remove-dimension scheme

The classification results in Section 4.2.1 are not satisfying because the discriminant hyperplanes are nearly parallel. Thus we derive a scheme that generates perpendicular hyperplanes and expect better results from it.

This process is as follows:

1) Get the Fisher vector W from samples in class ω_A and ω_B.
2) Apply the Gram-Schmidt algorithm to rotate the coordinate system. The new coordinate system uses W as a basis vector.
3) Get the representation of all the training images in this coordinate system and remove the subspace spanned by the vector W. Thus the number of dimensions of training images decreases by one.
4) From the dimension-reduced training images, get a new Fisher vector.
5) Repeat the steps 2) to 4) until we run out of dimensions

The Gram-Schmidt algorithm can obtain an orthonormal basis. In an n-dimensional space, the standard basis vectors are

$\mathbf{e}_1 = \{1,0,0,..., 0\}$

$\mathbf{e}_2 = \{0,1,0,..., 0\}$

$\vdots$

$\mathbf{e}_n = \{0,0,0,..., 1\}$

Assume a vector $\mathbf{s}_1 = \{s_{11}, s_{12},..., s_{1n}\}$ is a unit-length vector and $s_{11} \neq 0$. If $\mathbf{s}_1$ stands for a basis vector in a space, then the other basis vectors $\mathbf{s}_2$,...., $\mathbf{s}_n$ of this space are obtained from the formulas:

$$\mathbf{v}_k = \mathbf{e}_k - \sum_{i=1}^{k-1} (\mathbf{e}_k \cdot \mathbf{s}_i)\mathbf{s}_i \tag{4.4}$$

$$\mathbf{s}_k = \frac{\mathbf{v}_k}{\|\mathbf{v}_k\|} \tag{4.5}$$

where $k = 2,..., n$.

A vector $\mathbf{b} = \{b_1, b_2,..., b_n\}$ in the standard space becomes $\mathbf{b}' = \{b'_1, b'_2,..., b'_n\}$ in the transformed space. The relationship between them is

$$\mathbf{b}' = \mathbf{S}\mathbf{b} \tag{4.6}$$

where $\mathbf{S} = \begin{bmatrix} \mathbf{s}_1^T \\ \mathbf{s}_2^T \\ \vdots \\ \mathbf{s}_n^T \end{bmatrix}$.

Then the first element b'_1 is removed and therefore the number of dimensions is reduced by one.

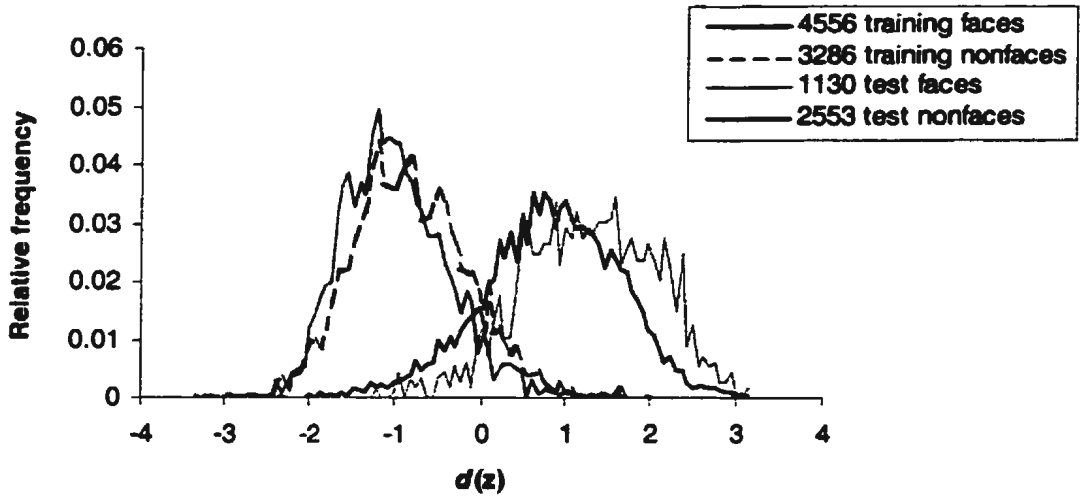
This technique was applied to the same sets of training and test faces/nonfaces as in Section 4.2.1. In the anything-image-whitened space, the number of dimensions at the beginning is 150. After each round, a Fisher vector is calculated and the number of dimensions decreases by one.

Table 4.7 lists the angle between every pair of Fisher vectors. The four Fisher vectors are orthogonal.

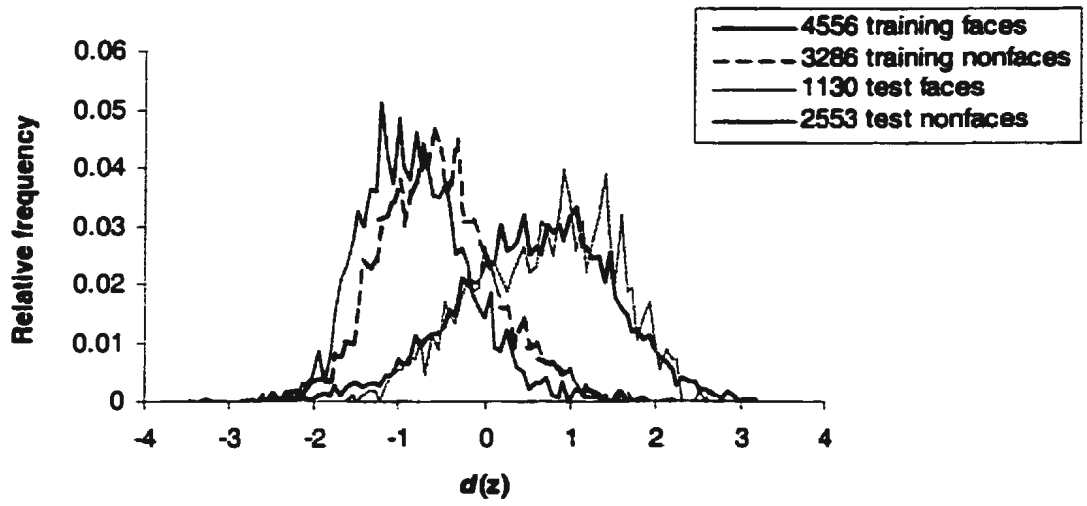
Table 4.7 Angles between a pair of Fisher vectors

	W_1	W_2	W_3	W_4
W_1		90°	90°	90°
W_2			90°	90°
W_3				90°

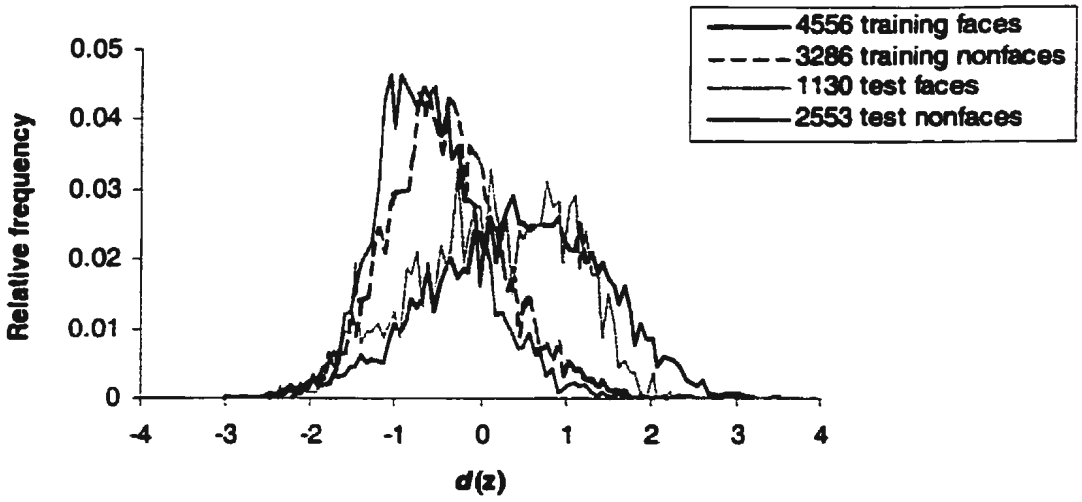
The projections onto W_2, W_3, and W_4 are shown in Figure 4.7. Note that because the same training data are used, the first Fisher vector W_1 is the same as that in Section 4.2.1. The projection onto W_1 has been shown in Figure 4.6a.



(a) Projection onto W_2 in a 149-dimensional space



(b) Projection onto W_3 in a 148-dimensional space



(c) Projection onto W_4 in a 147-dimensional space

Figure 4.7 The projection of images onto the Fisher vectors obtained from the rotate-coordinate-system-and-remove-dimension scheme

As shown in Figure 4.7, in the FLD subspace when the number of dimensions decreases the training face set and nonface set projection distributions become closer, and the overlap between the test face set and nonface set increases.

Table 4.8 Number of misclassified images when the Fisher vectors are applied sequentially ($T_1 = 0$, $T_2 = -1$, $T_3 = -1.2$ $T_4 = -2$)

	Misclassified nonfaces	Misclassified faces
Using W_1	91	11
Using W_1 and W_2	52	13
Using W_1 ,W_2 and W_3	40	18
Using W_1 ,W_2, W_3 and W_4	40	25

Table 4.8 shows the number of misclassified test faces and nonfaces when the thresholds are chosen to be (0, −1, −1.2,−2). These results are better than those in Table 4.6, the best we have achieved in Section 4.2.1. The last Fisher vector W_4 does not eliminate any nonfaces but contributes to misclassifying more faces.

If the last two thresholds are modified to (−1.3,−2.5), we get new results as shown in Table 4.9,

Table 4.9 Number of misclassified images when the Fisher vectors are applied sequentially ($T_1 = 0$, $T_2 = -1$, $T_3 = -1.3$ $T_4 = -2.5$)

	Misclassified nonfaces	Misclassified faces
Using W_1	91	11
Using W_1 and W_2	52	13
Using W_1 ,W_2 and W_3	42	16
Using W_1 ,W_2, W_3 and W_4	42	17

After experimenting with various groups of $T_i, i = 1,..,4$, we regard the results of 42 misclassified test nonfaces and 16 misclassified test faces as the best that repeated FLD can achieve.

4.2.3 Closest-nonface scheme

In the face-image-whitened space or the double-whitened space, the shape of face samples is a hypersphere. The nonface samples scatter all around as shown in the 2D illustration Figure 4.8. We look for a discriminant hyperplane that can separate these two classes. If this hyperplane is found, another hyperplane that is orthogonal to the previous one can separate another group of nonfaces from faces. In an $N-1$ dimensional space, this process will generate $N-1$ discriminant hyperplanes that are orthogonal to each other. The closest-nonface scheme is proposed to give good directions for measurement and generate these hyperplanes.

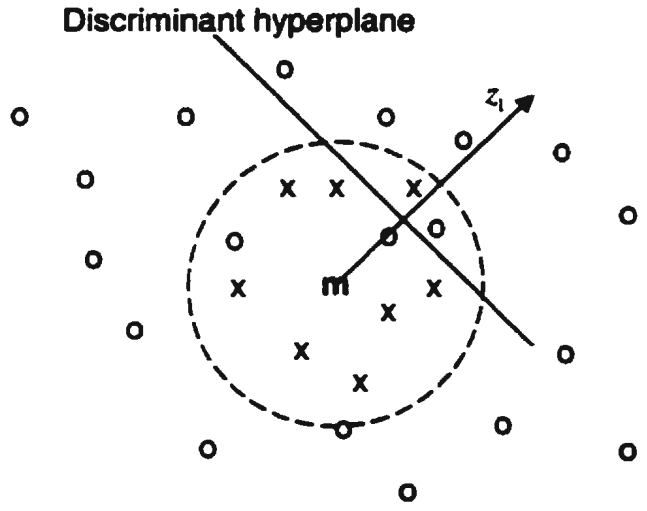
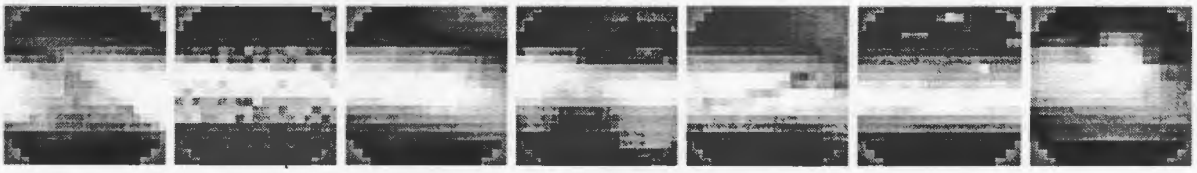


Figure 4.8 Illustration of furthest nonface scheme. "x" stands for a face sample. "o" stands for a nonface sample.

As depicted in Figure 4.8, in the face-image-whitened space, we implement the closest-nonface scheme in the following steps.

1) Find the closest nonface to mean face $\mathbf{m}$ in terms of Euclidean distance. Denote this nonface as $\mathbf{y}_1$.
2) Get the difference vector $\mathbf{z}_1 = \mathbf{y}_1 - \mathbf{m}$. Normalise $\mathbf{z}_1$.
3) Project all the faces and nonfaces onto $\mathbf{z}_1$. Set a threshold to do classification according to the projections.
4) Use Gram-Schmidt algorithm to get a space using $\mathbf{z}_1$ as one of its basis vectors. Remove the $\mathbf{z}_1$ direction. Thus the number of dimensions decreases by one. Eliminate the nonface $\mathbf{y}_1$.
5) Repeat the steps 1 to 4 until the process runs out of dimensions.

We applied this scheme to a set of face and nonface images in a 170-dimensional face-image-whitened space. The closest nonfaces in each iteration are shown in Figure 4.9. The number of dimensions left is marked under each nonface image.

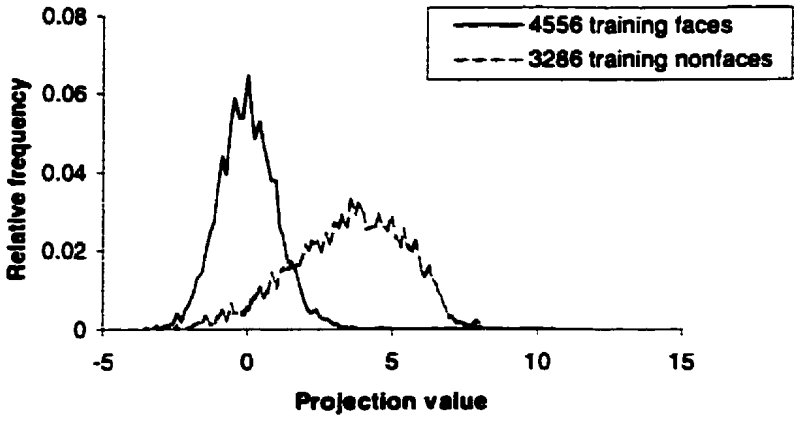


(a) 170 (b 169 (c) 168 (d) 167 (e) 166 (f) 165 (g) 164

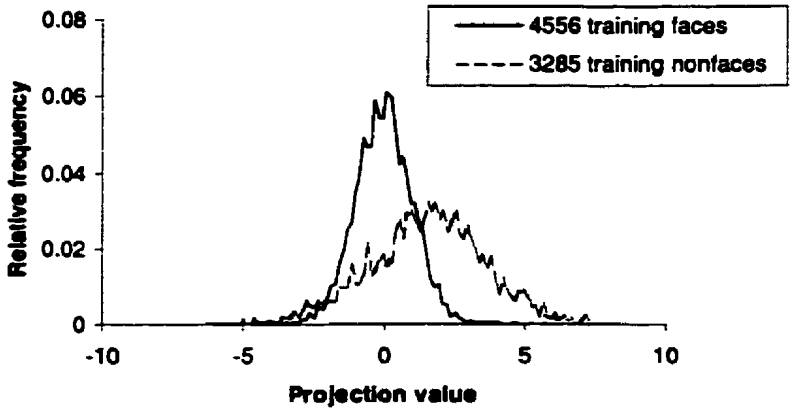
Figure 4.9 Closest nonfaces to mean face in each iteration

All the 7 closest nonfaces are of low frequency and look similar to each other.

In a 170-dimensional face-image-whitened space, the projection of images onto the vector $\mathbf{z}_1$ corresponding to the closest nonfaces in Figure 4.9a and b is shown in Figure 4.10a and b.



(a) first round



(b) second round

Figure 4.10 In the face-image-whitened space, the projection of images onto the vector from the closest nonface to the mean face

Figure 4.10 shows that the projections of faces and nonfaces onto $\mathbf{z}_1$ overlap more and more when the number of dimensions decreases. This makes it difficult to set a threshold to do classification.

4.2.4 Furthest-face scheme

Continuing the idea from last section, we tried to use a "furthest-face scheme" to find a set of discriminant hyperplanes. This scheme is identical to the closest-nonface scheme except that the furthest face, instead of the closest nonface, to mean face is sought.

In a 200-dimensional face-image-whitened space, the projections onto $\mathbf{z}_1$ are shown in Figure 4.11. The overlap between faces and nonfaces is large.

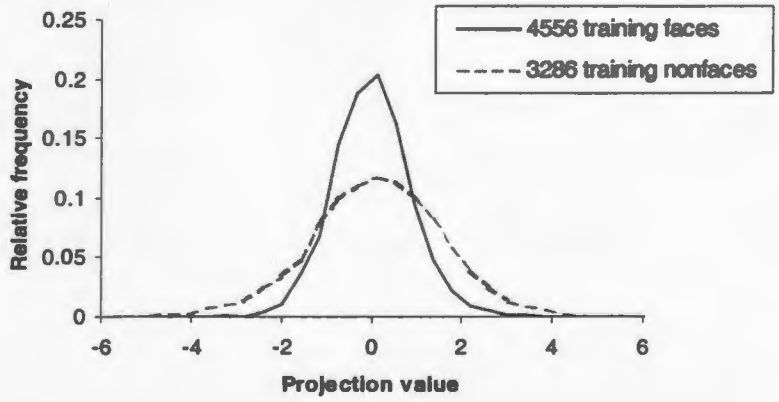


Figure 4.11 In the face-image-whitened space, the projection of images onto the furthest face to the mean face

In subsequent dimension reduction, the projection of faces and nonfaces still overlap to a large extent.

The furthest faces in each iteration are shown in Figure 4.12. The number of dimensions left is marked under each face image.

(a) 200 (b) 199 (c) 198 (d) 197 (e) 196 (f) 195 (g) 194

Figure 4.12 Furthest faces to mean face in each iteration

These furthest faces are noisy. Because in the face set 10 images are generated from one face picture, these 10 images can be divided into two sets. Each set is the mirror of the other set. The phenomenon that we observe is that Figure 4.12b is the mirrored version of Figure 4.12a. If a face is the furthest face in one iteration, its mirrored version is the furthest face in next iteration.

The experimental results of nearest-nonface and furthest-face schemes prove that these two schemes are susceptible to noise and are not good for classification.

4.3 Moving-Centre Scheme

Parametric classifiers, including probabilistic classifiers and linear classifiers, estimate unknown parameters and replace the true parameters with the estimated parameters, which may not be the optimum parameters. In this section, we propose an iterative classifier that is non-parameterised and is designed to find the optimum decision boundary.

4.3.1 Principles

We now show that if a set is convex, the decision boundary between this set and the nonset is dominated by this set.

Assume $\mathbf{m}_i$ and $\mathbf{S}_i$ are the sample mean and covariance matrix of the class ω_i. The Mahalanobis distance from a sample $\mathbf{x}$ to the class mean $\mathbf{m}_i$ is $(\mathbf{x}-\mathbf{m}_i)^T\mathbf{S}_i^{-1}(\mathbf{x}-\mathbf{m}_i)$. The Minimum Intra-Class Distance (MICD) metric compares the Mahalanobis distances to two class means and classifies an unknown sample $\mathbf{x}$ into class ω_1 if and only if

$$(\mathbf{x}-\mathbf{m}_1)^T\mathbf{S}_1^{-1}(\mathbf{x}-\mathbf{m}_1)-(\mathbf{x}-\mathbf{m}_2)^T\mathbf{S}_2^{-1}(\mathbf{x}-\mathbf{m}_2)<0$$

$$\Rightarrow \mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{W}^T\mathbf{x} + w_0 < 0 \qquad (4.7)$$

where $\mathbf{Q} = \mathbf{S}_1^{-1} - \mathbf{S}_2^{-1}$, $\mathbf{W} = 2(\mathbf{m}_2\mathbf{S}_2^{-1} - \mathbf{m}_1\mathbf{S}_1^{-1})$, and $w_0 = \mathbf{m}_1^T\mathbf{S}_1^{-1}\mathbf{m}_1 - \mathbf{m}_2^T\mathbf{S}_2^{-1}\mathbf{m}_2$.

As shown in Equation 4.7, the MICD classifier defines a decision boundary that partitions the feature space into regions for each class. The form of decision boundary depends on $\mathbf{Q}$, $\mathbf{W}$, and w_0, which rely on the class means and covariance matrices.

If the class ω_1 contains face samples and the class ω_2 contains nonface samples, the nonface class is a much broader category than the face class. The variation in ω_2 is much larger than that in ω_1. Thus $\mathbf{S}_1^{-1}$ is much larger than $\mathbf{S}_2^{-1}$. Therefore from the composition of matrix $\mathbf{Q}$, we can see that the shape of the decision boundary is dominated by $\mathbf{S}_1^{-1}$. The decision boundary is illustrated in a two-dimensional space as shown in Figure 4.13.

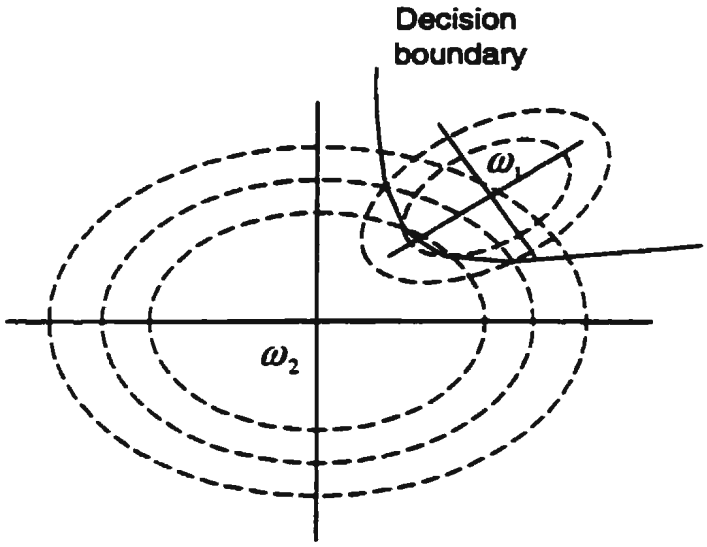


Figure 4.13 Two-dimensional illustration of the distribution of face and nonface images. Each distribution is represented by its principal axes and a collection of equidistance contours. Crossing points of equidistance ellipses are points on class boundary.

When conducting classification, we begin with the shape of the boundary defined by $\mathbf{S}_1^{-1}$, then adjust the decision boundary while keeping the shape constant until an optimum boundary is found. In the face-image-whitened space or the double-whitened space, the shape of the face class is a hypersphere. By measuring Euclidean distance all we need to specify is the centre of the sphere and the radius. The task is translated into adjusting the centre and radius of the hypersphere to reduce misclassifications. We call this scheme the "moving-centre scheme".

Figure 4.14 illustrates the moving-centre scheme.

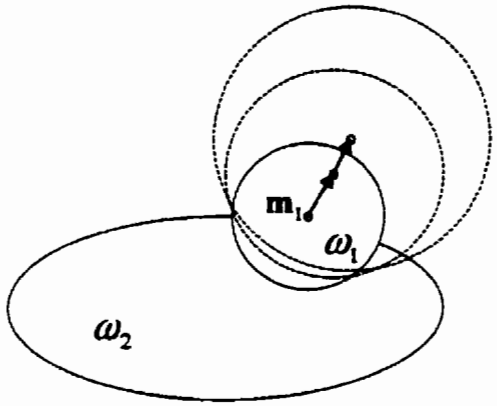


Figure 4.14 Illustration of moving-centre scheme

The distribution of class ω_1 is a hypersphere with centre $\mathbf{m}_1$. Then this centre retreats to a position where the two classes are best separated according to the Euclidean distance to the new centre.

The ideal centre $\mathbf{m}_1$ is sought by the steepest descent algorithm.

In [Cheney 1994], the steepest descent algorithm is described as follows.

A point $\mathbf{x}^*$ is sought such that

$F(\mathbf{x}^*) \leq F(\mathbf{x})$ for all $\mathbf{x} \in R^n$

At any point $\mathbf{x}$, the gradient vector $G(\mathbf{x})$ is calculated.

$$G_i(\mathbf{x}) = \frac{\partial F(\mathbf{x})}{\partial x_i} \qquad 1 \le i \le n$$

Then a one-dimensional minimisation problem is solved by determine the value t^* for which the function $\phi(t) = F(\mathbf{x} + t\, G(\mathbf{x}))$ is a minimum, then we replace $\mathbf{x}$ by $\mathbf{x} + t^* G(\mathbf{x})$ and begin anew.

In our application, the desired point $\mathbf{x}$ is $\mathbf{m}_1$ and $F(\mathbf{x})$ is the sum of error rates in the two training sets. The partial derivative $G_i(\mathbf{x})$ is approximated by the difference in $F(\mathbf{x})$ when the centre $\mathbf{m}_1$ moves along a basis vector $\mathbf{e}_i$ with a step of s, where

$\mathbf{e}_i = \{\underbrace{0,\ldots, 0}_{i-1},1,0\ldots, 0\}$

4.3.2 Experiments

In the **face-image-whitened** space that is composed of the top 170 eigenvectors of training faces, we get the mean of training faces and use it as the initial value of $\mathbf{m}_1$. We then apply the steepest descent algorithm to minimise the misclassification. The radius r is selected such that the error rates in two classes are as identical as possible. The average error rate of the two training sets after each round is shown in Figure 4.15. The displacement step $s = 0.5$.

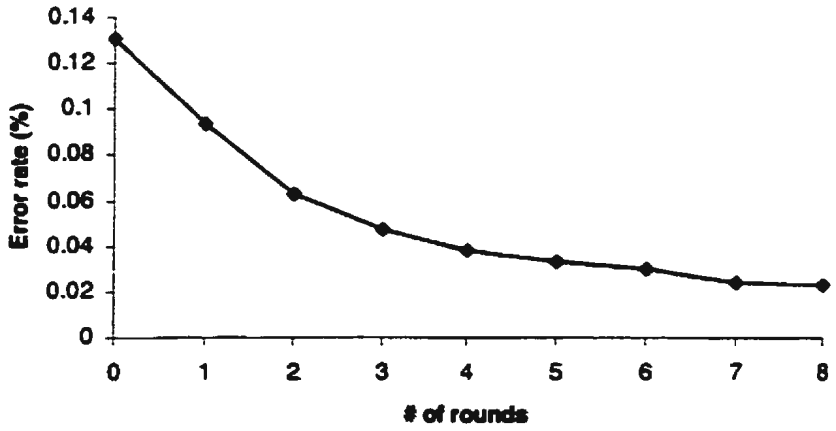


Figure 4.15 Average error rate of two training sets after each round using steepest descent algorithm in the face-image-whitened space

The error rate gradually goes down after each round. The radius versus the number of rounds is shown in Figure 4.16.

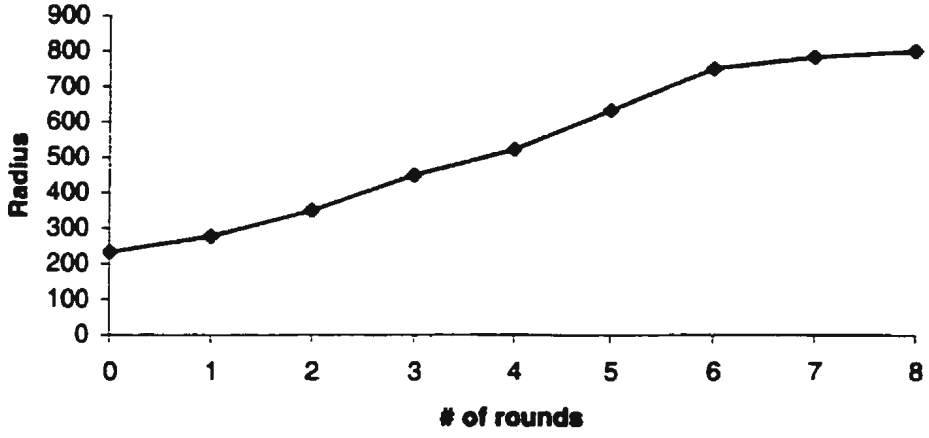


Figure 4.16 Radius after each round using steepest descent algorithm in the face-image-whitened space

The radius r at the beginning is 234.9, but after 8 rounds it increases to 798.9.

In the **double-whitened** space we run the same program on the same training data sets.

In anything-image-whitening stage, take the top $K = 100$ eigenvectors to compose the anything-image-whitened space. In the face-image-whitening stage, take all the 100 eigenvectors. The average error rate of two training sets after each round is shown in Figure 4.17. The displacement step $s = 0.516$.

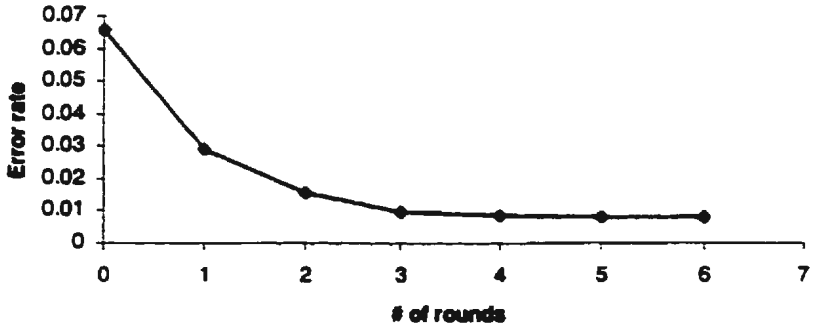


Figure 4.17 Average error rate of two training sets after each round using steepest descent algorithm in the double-whitened space

The radius r increases from 157.71 to 380.68 after 6 rounds.

Comparing the error rate in Figure 4.15 and Figure 4.17 at the end of each iteration, we can see that the double-whitened space provides a lower error rate, and thus is superior to the face-image-whitened space regarding the Euclidean distance in classification.

We then calculate the Euclidean distance for test faces and test nonfaces using the new centre and radius obtained from the steepest descent algorithm. The misclassification rate for the test face set is 0.53%, and for the nonface test set is 1.49%. These results are slightly better than the repeated FLD scheme in Section 4.2.

4.3.3 Parameter selection

In Section 4.3.2 we concluded that the double-whitened space provided a better classification than face-image-whitened space. Now we discuss the parameters involved in the steepest descent algorithm: K, the number of dimensions of the anything-image-whitened space, and the displacement step s. The number of dimensions of the face-whitening space is set to K.

The error rates in the test sets versus K are shown in Figure 4.18 while s is fixed at 0.5. This figure shows that the best selection of K is 100 or 150.

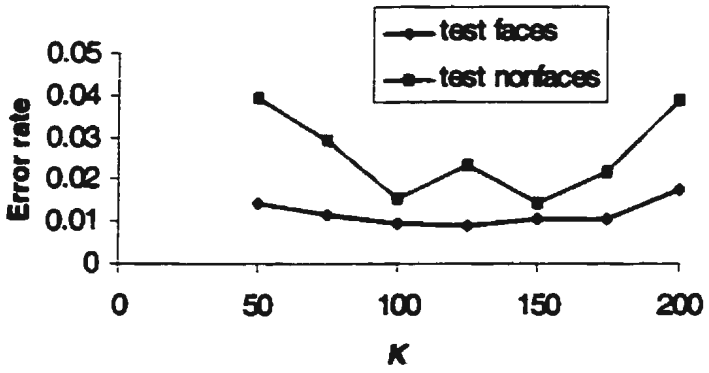


Figure 4.18 Error rates versus the number of dimensions in the double-whitened space using moving-centre scheme

The selection of the displacement step s is also important. If s is too big, the optimum centre may be jumped over. On the other hand, if s is too small, the steepest descent algorithm will stop at a local minimum.

In the double-whitened space (K =100) the relationship between the error rate of test sets and the step size s is listed in Table 4.10.

Table 4.10 Error rates versus displacement step using moving-centre scheme

s	Misclassified nonfaces (%)	Misclassified faces (%)
0.25	2.23	0.97
0.50	1.53	0.97
0.75	1.25	0.80
1.00	1.25	0.71
1.25	1.76	0.80

The smallest error rate is obtained when s = 1.00. The number of misclassified test faces is 8, and the number of misclassified test nonfaces is 32.

Figure 4.19 Eight misclassified faces using moving-centre scheme in the double-whitened space

Then we compare the results with those of the FLD classifier. In the same double-whitened space (K = 100, M = 100), if the FLD is applied, the rate of misclassifications is 10 out of 1130 faces, and 88 out of 2553 nonfaces. The misclassified test faces are shown in Figure 4.20.

Figure 4.20 Ten misclassified faces using the FLD in the double-whitened space

The moving-centre scheme is simple in calculation and has achieved better results than the FLD.

4.4 ML Classifier Based on Hyperellipsoid Distribution

A Gaussian distribution, or multinormal distribution, has often been assumed for discriminant analysis in high dimensions [Moghaddam 1997, Sung 1998]. However, in the real world many applications have feature values in a finite range. For example, in the image processing area, greyscale images are represented by values within a finite interval. In these cases, a hyperellipsoid distribution might be a good approximation of the data.

4.4.1 The hyperellipsoid distribution

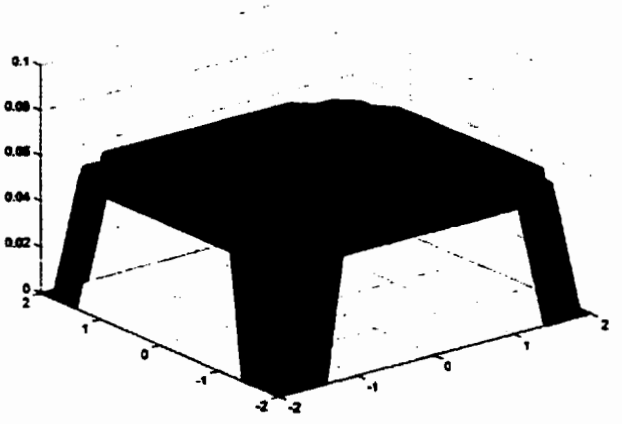
The hyperellipsoid distribution is introduced in [Tou 1973]. In [Huang 1994] the hyperellipsoid probability density function, which is zero outside of a hyperellipsoid, is used to model the data and to construct the discriminant function using the Bayes rule.

The probability density function of a hyperellipsoid distribution is

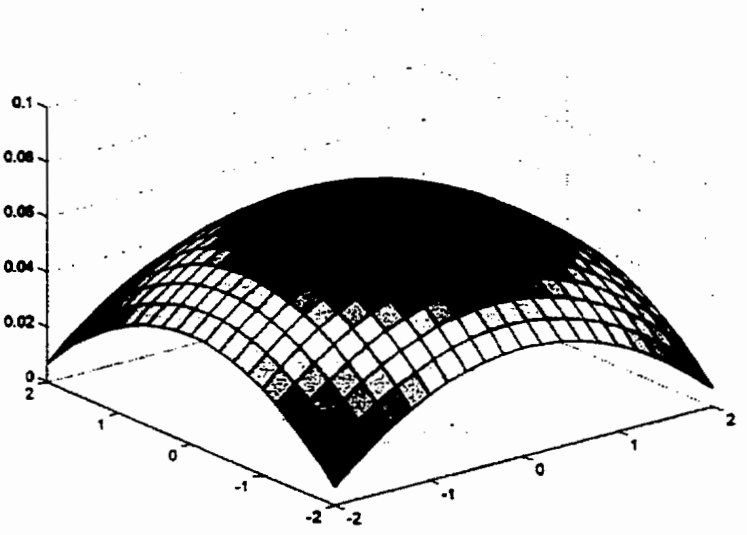
$$p(\mathbf{x}|\ \mathbf{m}, \mathbf{W}, n)$$

$$= \begin{cases} \dfrac{\Gamma(p/2+n+1)}{\pi^{p/2}\Gamma(n+1)}|\mathbf{W}|^{1/2}\left[1-(\mathbf{x}-\mathbf{m})^T\mathbf{W}(\mathbf{x}-\mathbf{m})\right]^n & \text{if}(\mathbf{x}-\mathbf{m})^T\mathbf{W}(\mathbf{x}-\mathbf{m}) \le 1 \\ 0 & \text{otherwise} \end{cases}$$

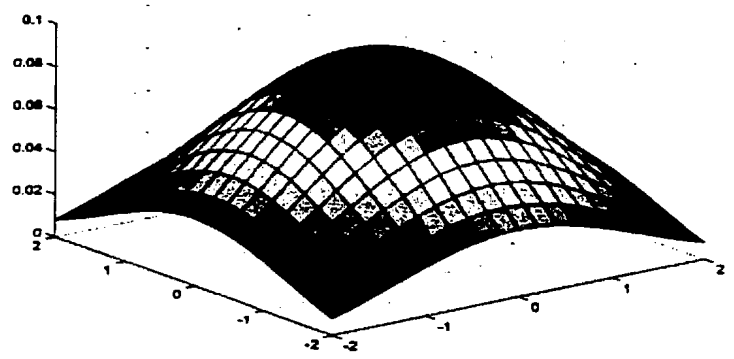
where $\mathbf{x}$ is a $p \times 1$ vector and $\mathbf{m} = E(\mathbf{X})$. Γ represents the gamma function, and the sample covariance matrix $Cov(\mathbf{X}) = \mathbf{W}^{-1}/(p + 2(n+1))$. This density is symmetric and is called Person type 2 density. Figure 4.21 shows the hyperellipsoid distribution in a 2-dimensional space when n varies.


0.1
0.08
0.06
0.04
0.02
0
2
1
0
-1
-2
-2
-1
0
1
2

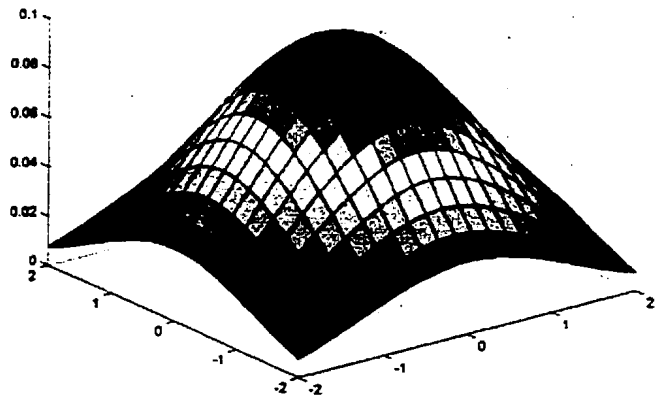
(a) $p = 2, n = 0$


0.1
0.08
0.06
0.04
0.02
0
2
1
0
-1
-2
-2
-1
0
1
2

(b) $p = 2, n = 1$



(c) $p = 2, n = 5$



(d) $p = 2, n = 10$

Figure 4.21 Hyperellipsoid distribution in a two-dimensional space

n is a parameter that controls the shape of the ellipsoid. When $n = 0$, this distribution is a hyperplane; When $n > 0$, this distribution is a hyperellipsoid; When $n \rightarrow \infty$, it becomes a multivariate normal distribution.

If there are c classes, $\omega_1, \ldots, \omega_c$, each with its respective $(\mathbf{m}_i, \mathbf{W}_i, n_i)$, $i = 1, \ldots, c$, then the Bayes rule classifies an unknown sample $\mathbf{x}$ into class ω_i if

$$P(\mathbf{x}|\ \omega_i)p(\omega_i) > P(\mathbf{x}|\ \omega_j)p(\omega_j),\ \forall j \neq i \tag{4.8}$$

Assuming all *a prior* probabilities are equal, and all the n_i's are equal, the above equation is changed into a Maximum Likelihood classifier.

A sample $\mathbf{x}$ is classified into class ω_i if

$$|\mathbf{W}_i|^{1/2n_i}\left[1-(\mathbf{x}-\mathbf{m}_i)^T\mathbf{W}_i(\mathbf{x}-\mathbf{m}_i)\right] > |\mathbf{W}_j|^{1/2n_j}\left[1-(\mathbf{x}-\mathbf{m}_j)^T\mathbf{W}_j(\mathbf{x}-\mathbf{m}_j)\right],\ \forall j \neq i \tag{4.9}$$

4.4.2 Experiments

Experiments are conducted in the 100-dimensional anything-image-whitened space (K = 100). To do the face/nonface classification, we estimate the hyperellipsoid parameters ($\mathbf{m}$, $\mathbf{W}$) for the face class and nonface class from the two training sets. Assume the parameter n's for the two classes are the same. In addition to Equation 4.9, two other classification rules are used:

In two class classification, if $p(\mathbf{x}|\ \omega_i) = 0$, i.e., $\mathbf{x}$ is outside the hyperellipsoid of class i, and $p(\mathbf{x}|\ \omega_j) \neq 0$, then sample $\mathbf{x}$ belongs to class j;

If $p(\mathbf{x}|\ \omega_i) = 0$ and $p(\mathbf{x}|\ \omega_j) = 0$, sample $\mathbf{x}$ is labelled as misclassified.

We do the face/nonface classification on 1130 test faces and 2553 test nonfaces. Figure 4.22 shows the number of misclassified images when n changes.

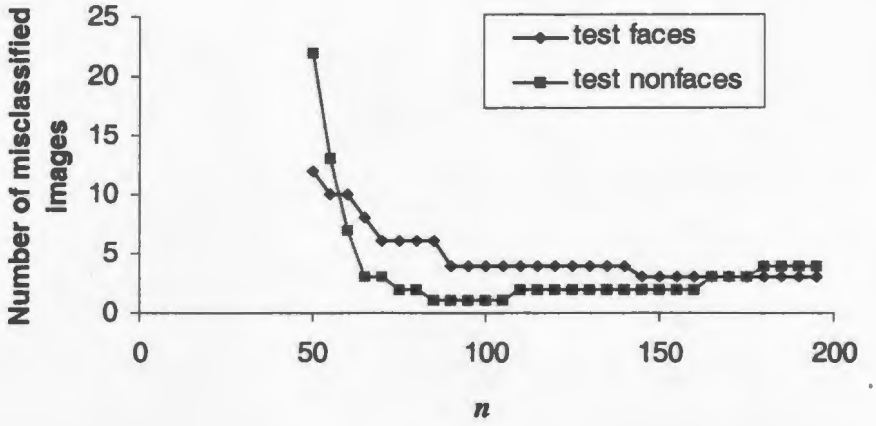


Figure 4.22 When $K = 100$, the number of misclassified images versus the parameter n of a hyperellipsoid distribution

Figure 4.22 shows that when n is from 90 to 105, this classifier achieved the best results: 4 misclassified test faces and 1 misclassified test nonface.

We are interested in the only misclassified test nonface when $n = 100$. This nonface is shown in Figure 4.23.

Figure 4.23 The only misclassified nonface when $n = 100$

It is very encouraging that this misclassified nonface does look like a real face. The 4 misclassified face images are shown in Figure 4.24.

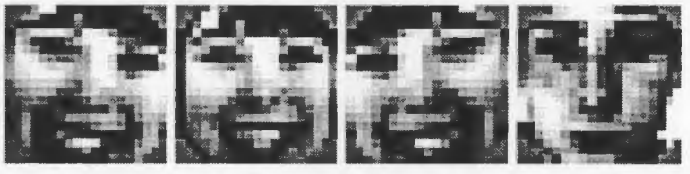


Figure 4.24 The four misclassified faces when $n = 100$

The first three faces belonging to one person are misclassified because of the hair in the forehead. The last image includes too many bright background pixels. As mentioned in Section 3.1.3, ten face images are extracted from one big face picture. This means that the other 7 face images for the first person and the other 9 face images for the second person are correctly classified.

4.5 ML Classifier Based on Gaussian Distribution

An M-dimensional Gaussian density is expressed as

$$p(\mathbf{x}|\ \omega) = \frac{\exp\left[-\frac{1}{2}(\mathbf{x}-\mathbf{m})^T\Sigma^{-1}(\mathbf{x}-\mathbf{m})\right]}{(2\pi)^{M/2}|\Sigma|^{1/2}} \tag{4.10}$$

Where $\mathbf{m}$ and Σ are the mean and covariance of this distribution.

In a two-class classification case, the ML classifier is expressed as an unknown sample $\mathbf{x}$ is assigned to class ω_1 instead of ω_2 if

$$p(\mathbf{x}|\ \omega_1) > p(\mathbf{x}|\ \omega_2) \tag{4.11}$$

Assuming the Gaussian distribution, this equation can be simplified as

$$(\mathbf{x}-\mathbf{m}_2)^T\Sigma_2^{-1}(\mathbf{x}-\mathbf{m}_2) - (\mathbf{x}-\mathbf{m}_1)^T\Sigma_1^{-1}(\mathbf{x}-\mathbf{m}_1) > \ln\frac{|\Sigma_1|}{|\Sigma_2|} \tag{4.12}$$

Let $d(\mathbf{x})$ equal the left side of the inequality

$$d(\mathbf{x}) = (\mathbf{x}-\mathbf{m}_2)^T\Sigma_2^{-1}(\mathbf{x}-\mathbf{m}_2) - (\mathbf{x}-\mathbf{m}_1)^T\Sigma_1^{-1}(\mathbf{x}-\mathbf{m}_1) \tag{4.13}$$

Then $d(\mathbf{x})$ is the difference between the Mahalanobis distance from the vector $\mathbf{x}$ to the two class means $\mathbf{m}_1$ and $\mathbf{m}_2$.

Assume class ω_1 represents the face class, and class ω_2 represents the nonface class. In the original space, the Mahalanobis distance is unavailable because the covariance

matrix of training faces is close to singular. This problem is avoided by projecting the image set to a lower-dimensional space, so that the resulting covariance matrix is nonsingular.

The face/nonface classification is performed in the following three lower-dimensional spaces.

- Anything-image-whitened space

  Figure 4.25 shows when K changes, the number of misclassified images changes accordingly. When K = 175, this classifier gives the best result: 3 misclassified test faces and 2 misclassified test nonfaces. Thus the misclassification rates for faces and nonfaces are 0.27% and 0.12% respectively.

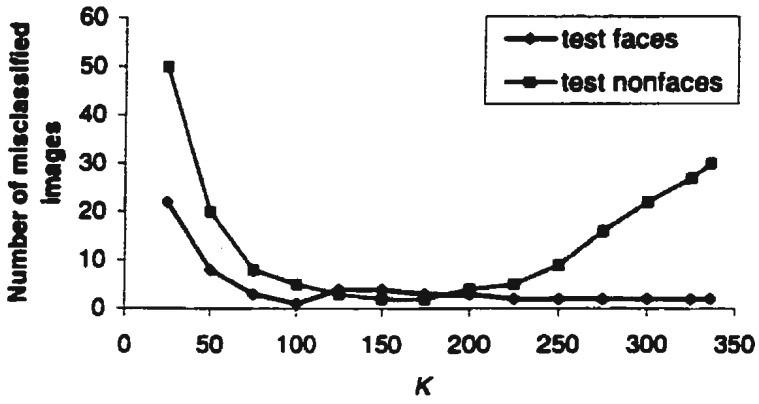


Figure 4.25 Using the ML classifier, the number of misclassified test faces and nonfaces versus the dimensionality of the anything-image-whitened space

In the case of K = 100, the ML classifier based on the hyperellipsoid distribution, as described in Section 4.4, generates better results (4 misclassified test faces and 1

misclassified test nonface) than those shown in Figure 4.25, where the results are 1 misclassified test face and 5 misclassified test nonfaces.

- Face-image-whitened space

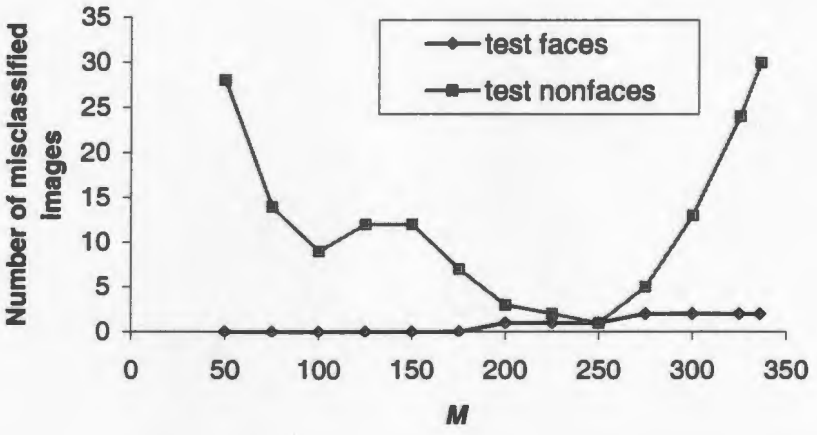
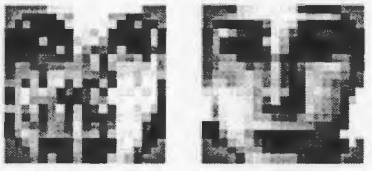


Figure 4.26 Using the ML classifier, the number of misclassified test faces and nonfaces versus the dimensionality of face-image-whitened space

When M = 250, the number of misclassified face is only one, and the number of misclassified nonfaces is also one. The misclassified nonface and face are shown in Figure 4.27.



(a) (b)

Figure 4.27 Classification results obtained using the ML classifier in the face-image-whitened space (a) misclassified nonface (b) misclassified face

Let M (the selected number of the largest eigenvalue eigenvectors of the training face images) be 250. The $d(\mathbf{x})$ of test faces and nonfaces are shown in Figure 4.28.

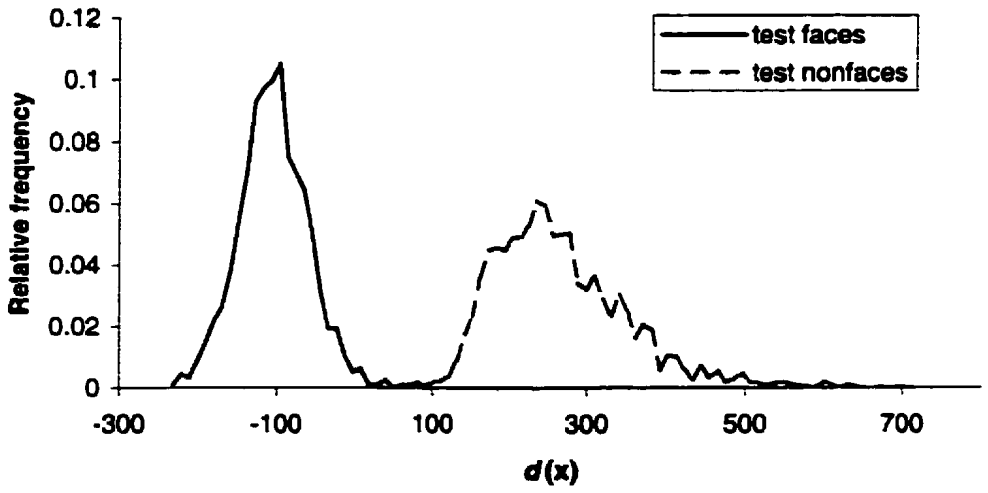


Figure 4.28 Difference of Mahalanobis distances from test samples to the two class means in the 250-dimensional face-image-whitened space

The two test sets are separated very well.

Figure 4.29 shows the misclassifications in the training sets.

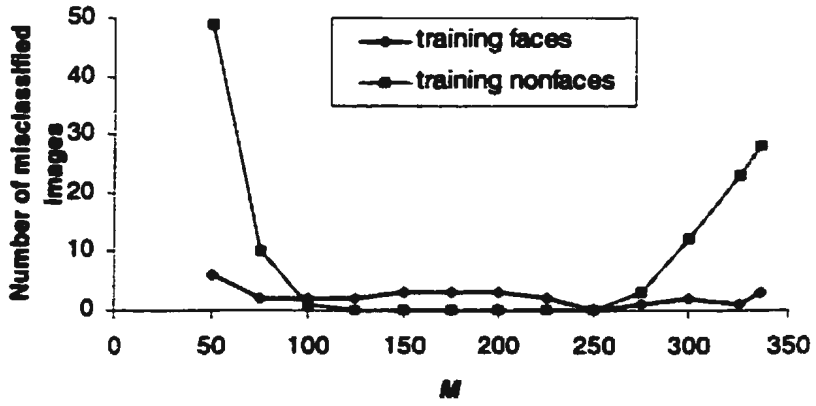


Figure 4.29 Using the ML classifier, the number of misclassified training faces and nonfaces versus the dimensionality of face-image-whitened space

When $M = 250$, there are no misclassifications. The long range (from $M = 100$ to $M = 275$) of low values suggests that the number of misclassifications is fairly insensitive to dimensionality after 100 dimensions.

- Double-whitened space

  The double-whitened space is obtained by anything-image whitening followed by face-image whitening. Figure 4.30 shows when the number of dimensions of the anything-image-whitened space is fixed at $K = 250$, the misclassifications versus M, the number of eigenvectors of training faces used to compose the double-whitened space.

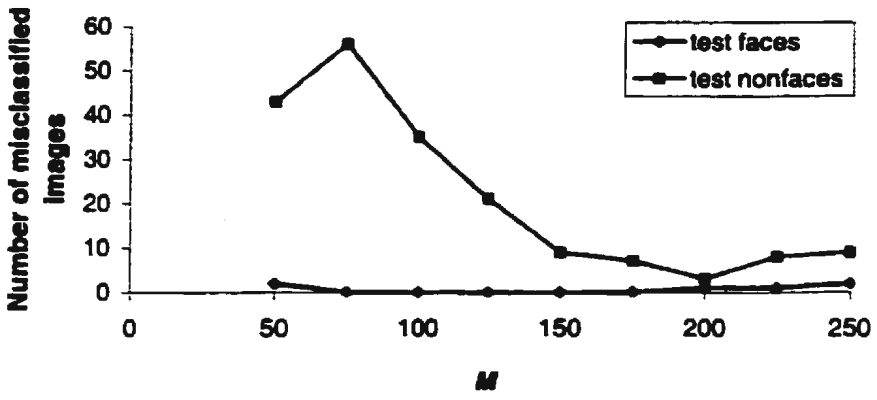


Figure 4.30 Using the ML classifier, the number of misclassified images versus the dimensionality of the double-whitened space ($K = 250$)

In the double-whitened space, the results are slightly worse than those in the face-image-whitened space. The best results, 1 misclassified face and 3 misclassified nonfaces, are achieved when $K = 250$ and $M = 200$. It proves that the double-whitening process, which increases computation demand, is unnecessary for the ML classifier.

4.6 ML Classifier Based on Principal and Complementary Spaces

Various models of training data in high-dimensional spaces have been proposed [Moghaddam 1994, Huang 1994]. In [Moghaddam 1994] a multivariate Gaussian (for unimodal distributions) model is adopted to estimate the probability density function of the training images. These probability densities are then used to formulate a maximum-likelihood classifier for object detection.

Given a set of training images $\{\mathbf{x}^t\}_{t=1}^{N_T}$, from an object class ω, the likelihood for this data, i.e., the class conditional density $p(\mathbf{x}|\omega)$, is estimated as follows:

Assume a Gaussian distribution. An orthogonal decomposition divides the vector space R^N into two mutually exclusive and complementary subspaces: the principal subspace (or feature space) $F = \{\Phi_i\}_{i=1}^{M}$ containing the principal components and its orthogonal complement $\overline{F} = \{\Phi_i\}_{i=M+1}^{N}$.

The likelihood is estimated as the product of two marginal and independent Gaussian densities.

$$p(\mathbf{x}|\omega) = \left[\frac{\exp(-\frac{1}{2}\sum_{i=1}^{M}\frac{y_i^2}{\lambda_i})}{(2\pi)^{\frac{M}{2}}\prod_{i=1}^{M}\lambda_i^{\frac{1}{2}}}\right]\left[\frac{\exp(-\frac{\varepsilon^2(\mathbf{x})}{2\rho})}{(2\pi\rho)^{(N-M)/2}}\right] = p_F(\mathbf{x}|\omega)\hat{p}_{\overline{F}}(\mathbf{x}|\omega) \qquad (4.14)$$

where $p_F(\mathbf{x}|\omega)$ is the true marginal density in F-space and $\hat{p}_{\overline{F}}(\mathbf{x}|\omega)$ is the estimated marginal density in the orthogonal complement $\overline{F}$ space. The set $\{\lambda_i\}$ are the eigenvalues.

The optimum weight ρ is defined as

$$\rho = \frac{1}{N-M}\sum_{i=M+1}^{N}\lambda_i \qquad (4.15)$$

ρ is simply the arithmetic average of the eigenvalues in the orthogonal subspace $\overline{F}$.

Assuming the prior probabilities are the same, we applied the ML classifier to face/nonface classification.

- In the anything-image-whitened space ($K = 336$)

  In every class take $M = 100$ eigenvectors. We estimate $p(\mathbf{x}|\omega_f)$ and $p(\mathbf{x}|\omega_{nf})$ for every sample $\mathbf{x}$, which is then classified into class ω_f if $p(\mathbf{x}|\omega_f) > p(\mathbf{x}|\omega_{nf})$.

  The results are 1 misclassified face and 58 misclassified nonfaces.

  If $M = 150$, we get 3 misclassified faces and 48 misclassified nonfaces.

- In the original 337-dimensional space

  We select M, the number of dimensions in the principle space, to be 150, then the test faces which have the smallest and the largest $p(\mathbf{x}|\omega_f)$ are shown in Figure 4.31.

$p(\mathbf{x}|\omega_f) = 2.36\times10^{-164}$

$p(\mathbf{x}|\omega_f) = 7.88\times10^{-11}$

Figure 4.31 Face images which have the smallest and largest $p(\mathbf{x}|\omega_f)$ in the150-dimensional eigenspace

When M varies, the number of misclassifications is shown in Figure 4.32.

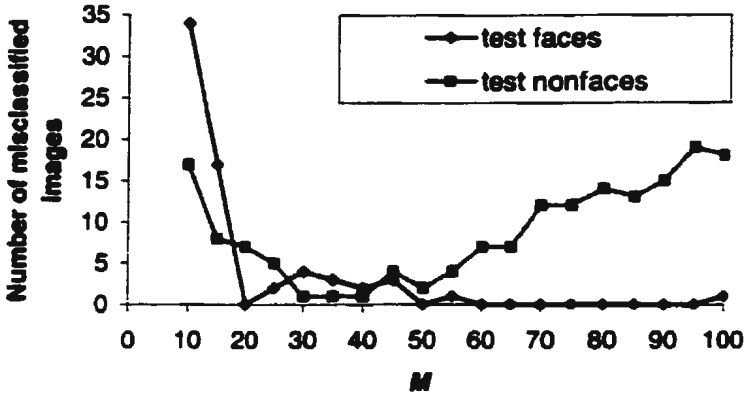


Figure 4.32 Using ML classifier in principal and complementary spaces, the number of misclassified images versus the dimensionality.

We can see that M = 50 gives the best results: no misclassified test faces and 2 misclassified nonfaces, which are shown in Figure 4.33.

Figure 4.33 The misclassified test nonfaces when $M = 50$

Although the lowest error rates in Figure 4.32 are similar to those in Figure 4.26, the small number of features required, 50, makes the ML classifier in principal and complementary spaces much more attractive than ML classifier in the face-image-whitened space only.

In [Moghaddam 1994], the classification is only based on the face density estimation $p(\mathbf{x}|\omega_f)$, i.e. the nonface density estimation $p(\mathbf{x}|\omega_{nf})$ is not used. To examine their method, we estimate $p(\mathbf{x}|\omega_f)$ for every image in the test face set and the test nonface set. Then a threshold is selected to make the misclassification rate in the two sets nearly equal. The results are

when $M = 10$, we get $e_3 = 13.45\%$ and $e_4 = 13.40\%$

when $M = 50$, we get $e_3 = 14.87\%$ and $e_4 = 13.63\%$

These results are significantly worse than those shown in Figure 4.32. Therefore, $p(\mathbf{x} \mid \omega_{nf})$ is indispensable.

4.7 ML Classifier Based on the Dominant Features

PCA has been widely used for feature extraction. Another feature extraction technique described in [Fukunaga 1991] is used to extract dominant features.

Suppose there are two classes ω_1 and ω_2, each with its mean $\mathbf{m}_i \in R^N$ and covariance matrix $\mathbf{\Sigma}_i \in R^{N \times N}$, $i = 1, 2$, estimated from the training sets. N is the number of dimensions of each sample. If $\mathbf{m}_1 \neq \mathbf{m}_2$ and $\mathbf{\Sigma}_1 \neq \mathbf{\Sigma}_2$, the following procedure is used to find the solution of extracting M dominant features from an N-dimensional vector $\mathbf{x}$.

1) Compute the eigenvalues and eigenvectors of $\overline{\mathbf{\Sigma}}^{-1}(\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$, λ_i and ϕ_i, where $\overline{\mathbf{\Sigma}} = (\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)/2$. Since the rank of the matrix is one, only λ_1 is nonzero and the other λ_i's are zero. ϕ_1 is the same as the Fisher vector W described in Section 4.1. Use the Fisher vector as the first feature and transform the input vector $\mathbf{x}$ to $y_1 = \phi_1^T \mathbf{x}$. All information of class separability due to mean-difference is preserved in this feature.
2) Find a $(N - 1)$ dimensional subspace, B, which is orthogonal to ϕ_1 by using Gram-Schmidt's algorithm. In this subspace, there is no information of class separability due to mean-difference. Project all the training samples into this subspace. The projection of input vector $\mathbf{x}$ in this subspace is $\mathbf{y} = B\mathbf{x}$. In this subspace, the covariance matrix of class ω_1 is $\mathbf{\Sigma}_{1Y}$. Likewise, the covariance matrix of class ω_2 is $\mathbf{\Sigma}_{2Y}$

3) In the $(N-1)$ dimensional subspace, compute $\Sigma_{2Y}^{-1}\Sigma_{1Y}$ and its eigenvalues and eigenvectors, μ_i's and Ψ_i's.

4) Select the Ψ_i's which correspond to the $(M-1)$ largest $(\mu_i + 1/\mu_i + 2)$ terms, and transform $\mathbf{y}$ to $z_i = \Psi_i^T \mathbf{y}$ $\quad (i = 1,\ldots, M-1)$.

5) Form an M-dimensional vector as $[y_1, z_1, \ldots, z_{M-1}]^T$. This vector is the extracted feature vector for the input vector $\mathbf{x}$.

After all the training and testing samples are represented by their corresponding feature vectors, the ML classifier is used to do the face/nonface classification. In the original greyscale space, the classification results with respect to M are shown in Figure 4.34.

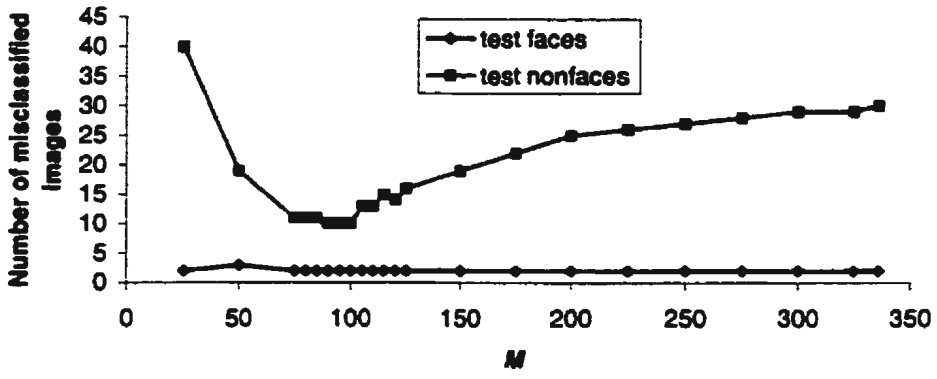


Figure 4.34 Number of misclassifications in the test sets versus the number of dominant features using the ML classifier in the original space

The results are promising. The fewest misclassifications, 2 misclassified test faces and 10 misclassified nonfaces, are obtained when M is from 90 to 100. The misclassification numbers are low and the best case is achieved with a small number of

features. Compared with Figure 4.25 and Figure 4.26, the smaller number of features required is the evident advantage of this feature extraction technique.

In a 250-dimensional face-image-whitened space, the best result is 4.4% error rate in the test set when the number of features is 100.

In a 250-dimensional anything-image-whitened space, the best result is 2.83% error rate in the test set when the number of features is 125, 225 or 250.

The relatively worse results in the latter two spaces are caused by the discarding of low energy components in the feature extraction process in these two spaces.

This is the first time that this feature extraction technique has been used in combination with the ML classifier in face/nonface classification.

Now we examine the classification results in the training sets in the original greyscale space as shown in Figure 4.35.

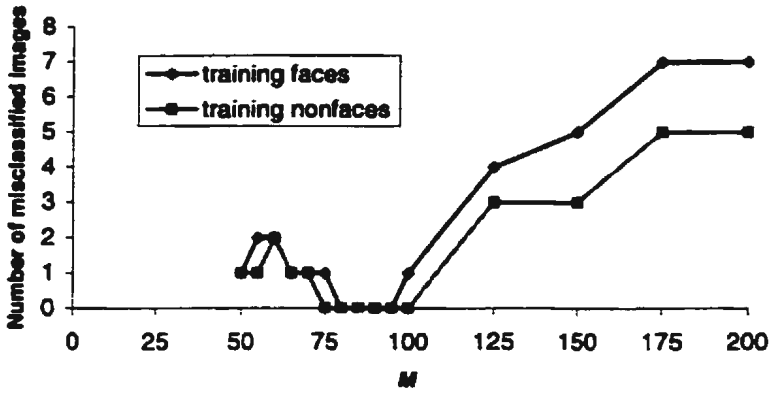


Figure 4.35 Number of misclassifications in the training sets versus the number of dominant features using the ML classifier in the original space

In this figure, the following equation, instead of Equations 4.12 and 4.13, is used to obtain the error rates.

$$d(\mathbf{x}) > s \tag{4.16}$$

where s is a separation point which is selected such that the error rate in the test face set and the error rate in the test nonface set are as close as possible.

The error rates in Figure 4.35 and Figure 4.29 are quite similar. This proves that the classification performances of the dominant feature extraction technique and the face-image-whitening scheme do not have statistical difference. However, Figure 4.35 shows that when the number of dimensions changes from 80 to 95, there are no misclassifications in the two training sets. Nevertheless, in Figure 4.29 250 features are required to get zero misclassifications. Hence, the dominant feature technique is superior to the face-image-whitening scheme in terms of number of features required.

4.8 Nearest Neighbour Classifier

The most straightforward nearest neighbour classifier is a non-parametric classifier and does not require any training. It can be conveniently used as a benchmark for all the other classifiers since it appears to always provide a reasonable classification performance in most applications. Further, as the nearest neighbour classifier does not require any user-specified parameters, its classification results are implementation independent.

In an image space, an unknown sample is classified into the class to which its nearest neighbour belongs. Therefore, we must make sure there are enough samples in each class.

In face/nonface classification, we use Euclidean distance as the measure. In the original 337-dimensional space, we yield no misclassified face out of 1130 test faces, but 138 misclassified nonfaces out of 2553 test nonfaces. The error rate of test nonfaces is 138/2553 = 5.4 %. These results reveal that the face samples are close to each other, but the nonface samples are rather scattered.

If all the images are projected into the anything-image-whitened space, the number of misclassifications is listed in Table 4.11. K is the number of dimensions of the anything-image-whitened space.

Table 4.11 Number of misclassified test images using the nearest neighbour classifier in the anything-image-whitened space

K	Misclassified nonfaces		Misclassified faces	
	Number	Error rate	Number	Error rate
50	113	4.43%	3	0.27%
100	30	1.18%	0	0%
150	16	0.63%	6	0.53%

The best results are achieved when K = 150. The biggest problem with the nearest neighbour classifier is the formidable time it consumes. All the experiments on nearest neighbour classifier, the following k-nearest neighbour classifier and k / l nearest neighbour classifier are implemented by using MATLAB on a Pentium II 400 MHz and 128 MB RAM computer. The MATLAB program took 3.84 hours in K = 50 case, and 4.32 hours in K =150 case.

Besides the long time it consumes, nearest neighbour classifier is sensitive to noise and outliers.

4.9 k-Nearest Neighbour Classifier

The k-nearest neighbour classifier is an extension of the nearest neighbour classifier. In a two-class classification case, it is assumed that class ω_1 contains samples $\{\mathbf{x}_i\}_{i=1}^{n_1}$, and class ω_2 contains samples $\{\mathbf{y}_j\}_{j=1}^{n_2}$. We seek the k nearest neighbours of an unknown

sample $\mathbf{x}$ in the composite set of samples in these two classes. $\mathbf{x}$ is classified into the class to which the majority of its k nearest neighbours belong. k must be an odd number. This classifier gives some protection against noise and gives better classification on boundary points.

The k-nearest neighbour classifier is applied to face/nonface classification.

- In the original greyscale space

  We do the k-nearest neighbour classification for test faces and test nonfaces. This is implemented by looking for the k-closest matches of a test image in these two training data sets.

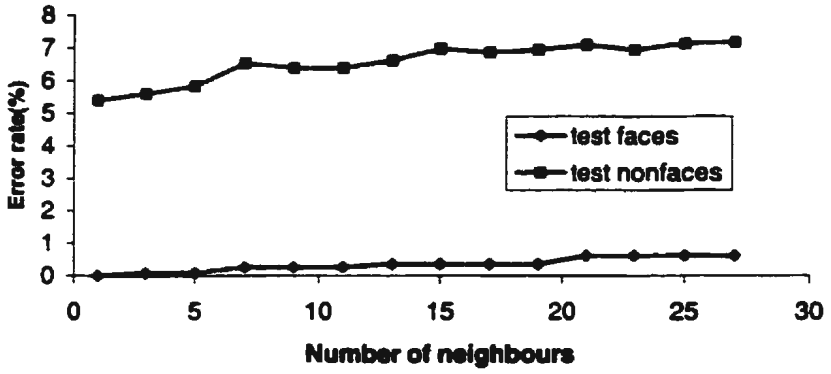


Figure 4.36 Error rates in the test sets versus the number of neighbours using the k-nearest neighbour classifier in the original space

The error rates slightly increase with k. Hence the best result is obtained when k = 1.

- In the 150-dimensional anything-image-whitened space

  For every image in the training face set and the training nonface set, we find its k nearest neighbours in these two sets as a whole using Euclidean distance, and then

do the face/nonface classification. The error rates versus k are shown in Figure 4.37. It took 9.30 hours for the MATLAB program to generate these results.

Figure 4.37 Error rates in the training sets versus the number of neighbours using the *k*-nearest neighbour classifier in the 150-dimensional anything-image-whitened space

Figure 4.37 shows that the number of neighbours, k, doesn't matter much when $k \geq 5$.

In the same anything-image-whitened space, the classification results for test sets, which cost 4.22 hours, are shown in Figure 4.38.

Figure 4.38 Error rates in the test sets versus the number of neighbours using the *k*-nearest neighbour classifier in the 150-dimensional anything-image-whitened space

The minimum of the sum of error rates is obtained at the point $k = 5$ with the error rate $14 / 2553 = 0.55\%$ for test nonfaces, and $2 / 1130 = 0.18\%$ for test faces.

- In the 150-dimensional face-image-whitened space

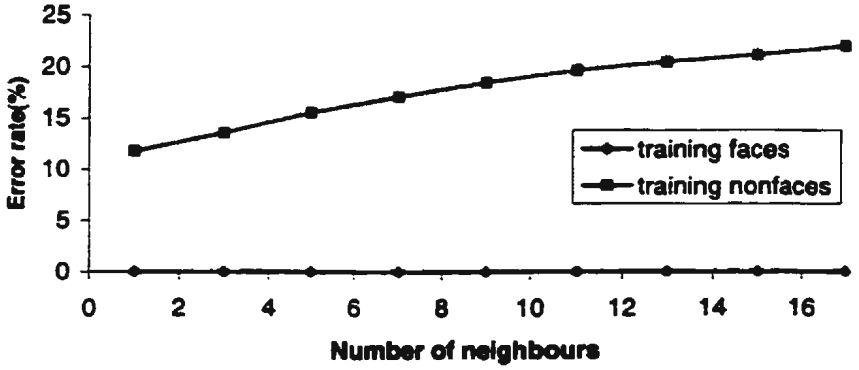


Figure 4.39 Error rates in the training sets versus the number of neighbours using the *k*-nearest neighbour classifier in the 150-dimensional face-image-whitened space

When k increases, the error rate of training faces, e_1, increases, but the error rate of training nonfaces, e_2, keeps stable at nearly zero.

Comparing the performance of the k-nearest neighbour classifier in the three feature spaces, we can see that the performance improves in the order of the face-image-whitened space, the original greyscale space, and the anything-image-whitened space. In the face-image-whitened space and the original space, the error rates increase with the number of neighbours, k. The experimental results disclose that the face class in the face-image-whitened space and the original space is more compact than that in the anything-image-whitened space. The sample cluster in face class is dense while the sample cluster in nonface class is sparse.

4.10 k / l Nearest Neighbour Classifier

To deal with the situation of dense face cluster and sparse nonface cluster, we apply the k/l nearest neighbour classifier. Similar to the k-nearest neighbour classifier, the k/l nearest neighbour classifier seeks the k nearest neighbours of an unknown sample $\mathbf{x}$ in the composite set of samples in class ω_A and ω_B. However, $\mathbf{x}$ is classified into class ω_A if l or more than l of its k nearest neighbours belong to class ω_A.

Suppose ω_A represents the face class and ω_B represents the nonface class. We first use the k / l nearest neighbour classifier to classify the training samples. Then we determine the combination of k / l that give the lowest sum of training face misclassification rate, e_1, and the training nonface misclassification rate, e_2.

- In the original greyscale space

For a specific k, the optimum l which gives the lowest $e_1 + e_2$ is listed in Table 4.12.

Table 4.12 Using the k/l nearest neighbour classifier, the number of misclassifications in the training sets in the original space

k	l	# misclassified training nonfaces	# misclassified training faces
1	1	177	0
3	3	78	1
5	5	49	12
7	7	41	23
9	9	35	32
11	11	27	43
13	13	23	54
15	**14**	**43**	**23**
17	16	41	29
19	18	38	37
21	20	36	49
23	21	49	29
25	23	44	38

The best result of 23 misclassified faces and 43 misclassified nonfaces is achieved at $k = 15$ and $l = 14$. The l value is very close to k. This phenomenon implies that the nonfaces at the boundary between the two classes tend to have much more face neighbours than nonface neighbours.

Applying this classifier in the condition of $k = 15$ and $l = 14$ to the two test sets, we get 13 misclassified test faces and 47 misclassified test nonfaces.

- In the 150-dimensional face-image-whitened space

  The same experiment is conducted in the 150-dimensional face-image-whitened space. The apparently worse results in Table 4.13 suggest that the face-image

whitening removed some useful information contained in the discarded dimensions. Moreover, since the l value is identical to k, in this space the face samples are more dense and compact but the nonface samples become more dispersed.

Table 4.13 Using the k / l nearest neighbour classifier, the number of misclassifications in the training sets in the 150-dimensional face-image-whitened space

k	l	# misclassified training nonfaces	# misclassified training faces
1	1	387	1
3	3	200	6
5	5	139	12
7	7	107	23
9	9	90	40
11	11	73	53
13	13	61	61
15	15	52	71
17	17	48	88
19	19	43	97
21	21	38	100

The lowest $e_1 + e_2$ is achieved at $k = 15$ and $l = 15$. In that combination of k and l, $e_1 = 52 / 3286 = 1.58\%$, and $e_2 = 71 / 4556 = 1.56\%$

- In the 150-dimensional anything-image-whitened space

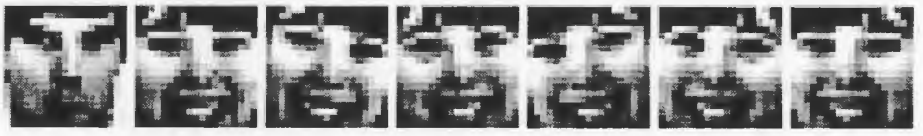
  The experiments in the 150-dimensional anything-image-whitened space give much better results as listed in Table 4.14.

Table 4.14 Using the k / l nearest neighbour classifier, the number of misclassifications in the training sets in the anything-image-whitened space

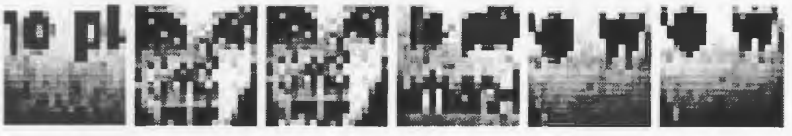
k	l	# misclassified training nonfaces	# misclassified training faces
1	1	38	2
3	3	10	12
5	4	**11**	**10**
7	6	5	16
9	7	7	17
11	8	10	17
13	8	12	13
15	10	10	14
17	9	13	10
19	11	11	12
21	12	10	14

The lowest $e_1 + e_2$ is achieved at $k = 5$ and $l = 4$.

Applying this classifier in the condition of $k = 5$ and $l = 4$ to the two test sets, we get 7 misclassified test faces and 6 misclassified test nonfaces as shown in Figure 4.40.



(a)



(b)

Figure 4.40 Using the k / l nearest neighbour classifier in the 150-dimensional anything-image-whitened space the misclassifications (a) in the test face set, and (b) in the test nonface set

Figure 4.40a clearly shows a deficiency in the training set: no training faces have dark hair on the forehead.

The results in Table 4.14 are better than those achieved by using the FLD or the repeated FLD, but not as good as those obtained by using the ML classifier.

4.11 Possible Classifiers

In this section we propose three classifiers that have achieved limited success.

4.11.1 The Euclidean distance to the mean face in the double-whitened space

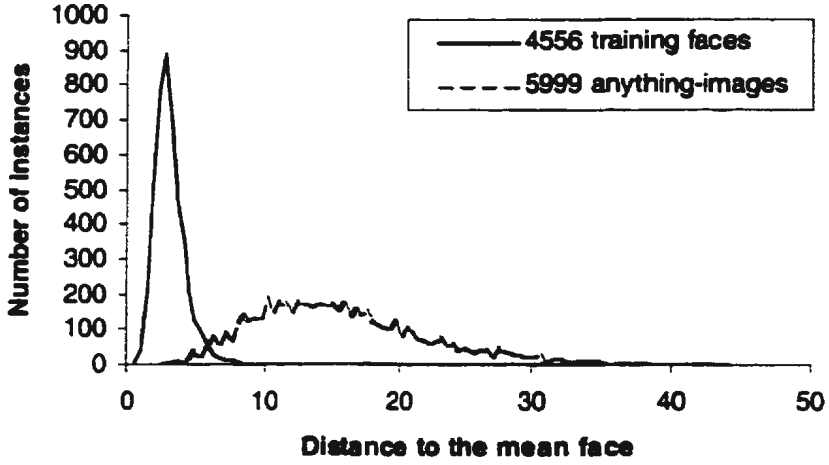
Now we look back at Figure 3.17, which shows that in the anything-image-whitened space, the directions corresponding to the smallest eigenvectors of face class should be the directions in which the variance of face space divided by the variance of the anything-image space is least.

If the dimensionality of the original space is N, the maximum available number of dimensions of the anything-image-whitened space is $K = N - 1$. The face-whitening step after anything-image whitening selects the M largest eigenvalue eigenvectors of face images to compose a double-whitened space, and discard the remaining $L = N - 1 - M$ eigenvectors. However, because these discarded eigenvectors represent directions in which the variance of face space divided by the variance of the anything-image space is least, we utilize the projections of images onto these discarded eigenvectors to derive a Euclidean distance measure. The Euclidean distance measure in the double-whitened space can be written as

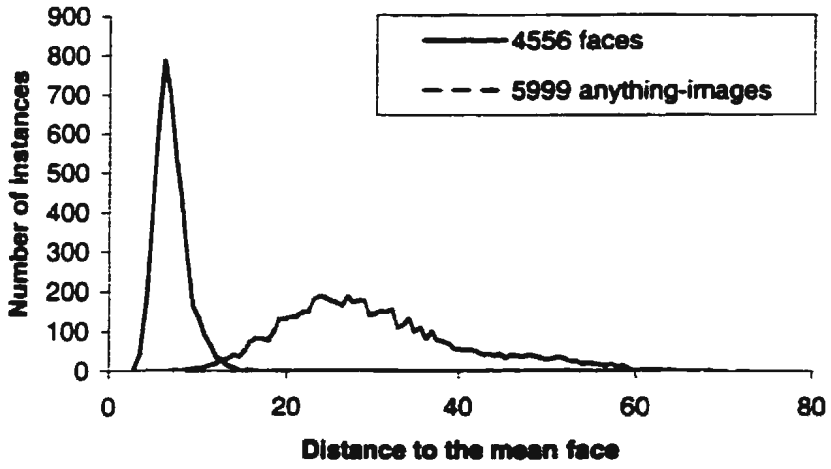
$$d(\mathbf{x}) = \left(\sum_{i=M+1}^{N-1} (\mathbf{x} - \mathbf{m})^T \phi_{f,i} \lambda_{f,i}^{-\frac{1}{2}} \right)^{\frac{1}{2}} \tag{4.17}$$

where $\mathbf{x}$ is an image in the anything-image-whitened space, i.e., $\mathbf{x} = \Lambda_a^{-\frac{1}{2}} \Phi_a^T \mathbf{I}$ and $\mathbf{I}$ is that image in the original space; $\mathbf{m}$ is the mean of all the face images in the anything-image-whitened space; $\phi_{f,i}$ is the i-th eigenvector of face images in the anything-image-whitened space. Note that the eigenvalues $\Lambda_f = \{ \lambda_{f,1}, \lambda_{f,2}, ..., \lambda_{f,N-1} \}$ are sorted in descending order.

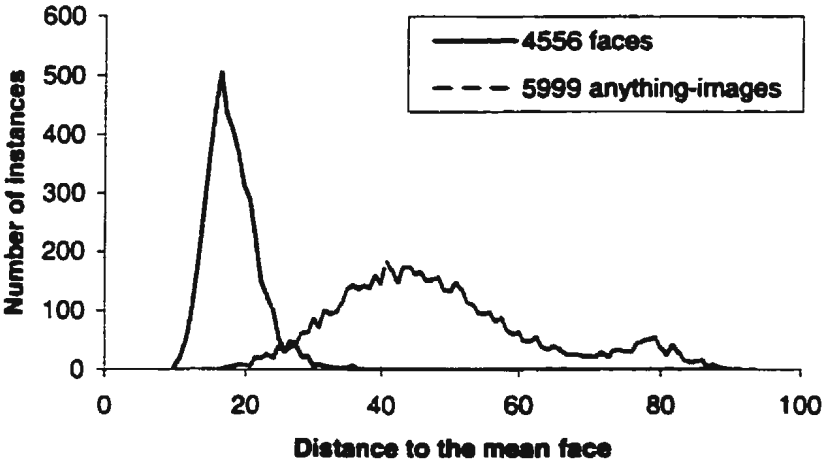
Figure 4.41 shows the distribution of Euclidean distances to the mean face using different values of L.



(a) $L = 11$



(b) $L = 50$



(c) $L = N - 1$

Figure 4.41 The Euclidean distance of training samples to the mean face in the double-whitened space using the dimensions corresponding to the L lowest eigenvalue eigenvectors of face images

From the Figure 4.41a to Figure 4.41c, the overlap between the face images and anything-images varies according to L. Among these three graphs, Figure 4.41b shows

the smallest overlap between faces and anything-images. Because none of the anything-images is a real face, the smaller the overlap, the better that the two image sets are separated.

Then we set up a threshold θ. An unknown image $\mathbf{x}$ is classified into the face class if $d(\mathbf{x}) <= \theta$; otherwise, it is classified into the nonface class.

In the case of each L, the number of misclassified face images and anything-images is obtained. The threshold θ is chosen such that the number of misclassified face images is kept at 50.

Figure 4.42 shows the number of misclassified anything-images varies with L. When $L = 60$, the misclassified anything-images are the fewest, 61.

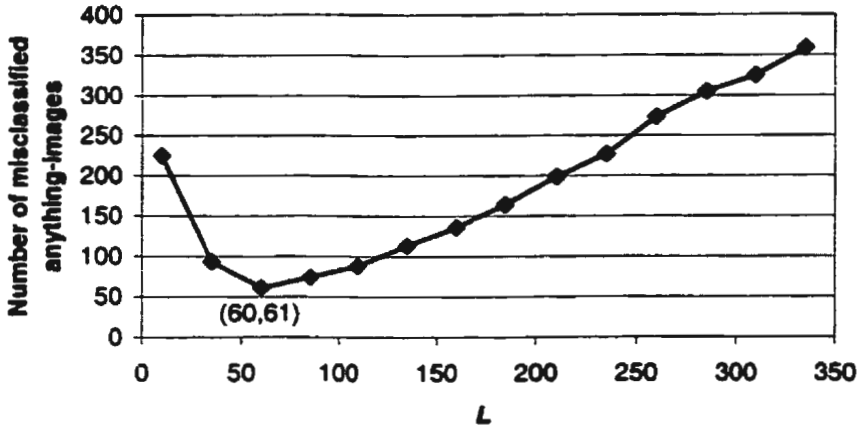


Figure 4.42 Number of misclassified anything-images versus the number of the lowest eigenvalue eigenvectors used

We used the two test sets to test the effectiveness of this Euclidean distance scheme, Their distance to the mean face according to Equation 4.17 is shown in Figure 4.43.

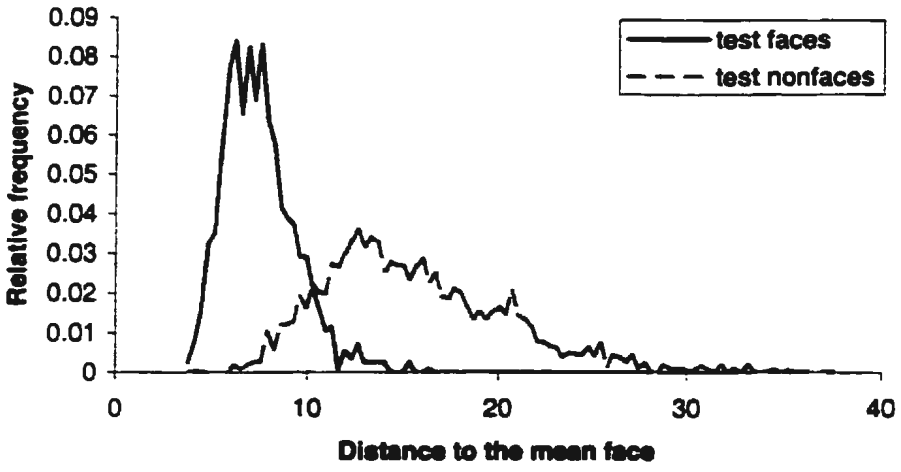


Figure 4.43 Distance of test samples to the mean face in the double-whitened space using the lowest 50 eigenvectors of training faces

Compared with Figure 4.41b, Figure 4.43 shows more overlap between the test faces and nonfaces. However, the test sets are distinct from the training sets, and the test nonfaces are closer to true faces in appearance than anything-images. If the threshold is set to 10, the error rate is 9.03% for test faces, and 9.99% for test nonfaces.

Figure 4.44 shows the Euclidean distances from test faces and nonfaces to the mean face in the original space. Comparing Figure 4.44 with Figure 4.43, we can see that the double-whitened space separates test faces and nonfaces much better.

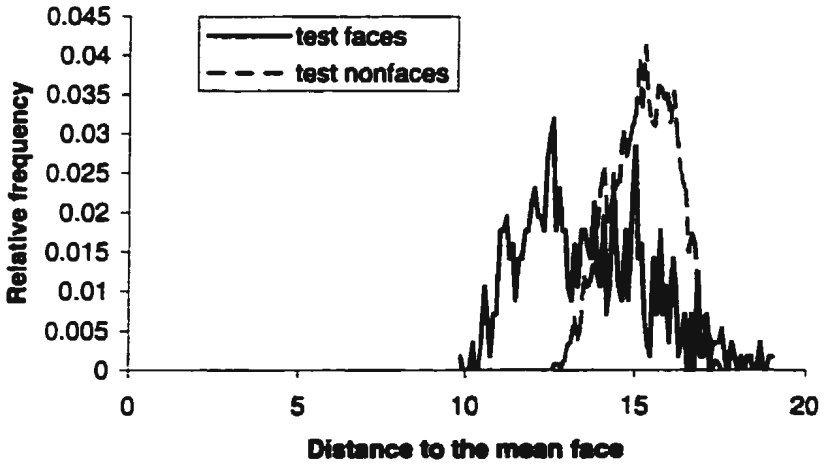


Figure 4.44 The Euclidean distance of test faces and nonfaces to the mean face in the original greyscale space

4.11.2 Miniball algorithm

In Section 4.4, we modelled the distribution of the face and nonface class as two hyperellipsoids. In this section, we explore the possibility of modelling the face class as a hypersphere in the original space, anything-image-whitened space and face-image-whitened space.

We would probably get a better understanding of the way to define a hypersphere from sparse data in high dimensions if we first find the "least extreme face". This is not the mean, rather it is the face which is the closest possible to the most distant face. Figure 4.45 illustrates this idea in 2D.

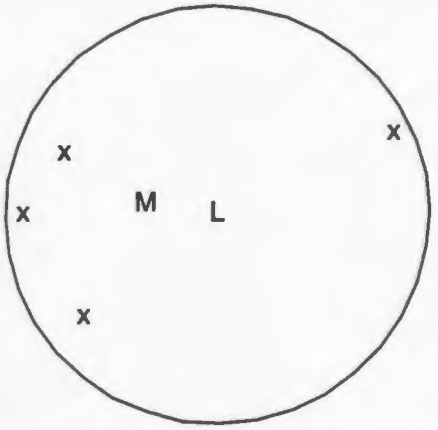


Figure 4.45 2D illustration of the miniball scheme

"x" is a sample; "M" is the mean of all the samples; "L" is the "least extreme" because the training point furthest away from it is as close as possible. With sparse data, this is probably a better estimate of the real mean than the sample mean. We also find that "L" is the centre of the smallest hypersphere that encloses all the faces.

This method was investigated by using a publicly available miniball program in C [Bernd]. The program can find the centre and radius of a smallest enclosing ball of a set of points in a multidimensional Euclidean space. It works well, except that the author said it was slow with greater than 30 dimensions.

The program was run on 1000 training faces in the 337-dimensional original greyscale space. The computed centre of this minimum enclosing ball is shown in Figure 4.46.

Figure 4.46 Centre of a minimum enclosing ball for 1000 faces in the original space

The miniball centre is a theoretical least extreme face, and is not a real face in the training set. The miniball centre maintains the basic features of a human face, such as two dark eyes and the mouth. This image is noisier than the mean face shown in Figure 3.5.

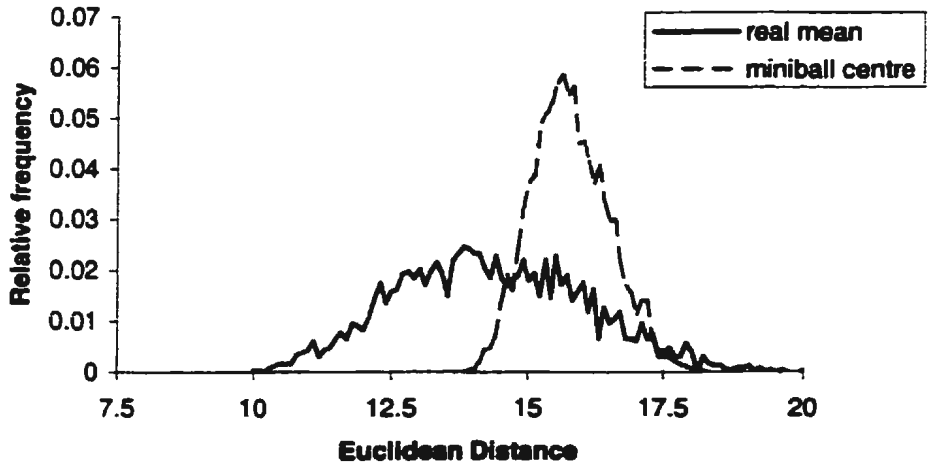


Figure 4.47 Euclidean distance to the real mean and the miniball centre of training faces

Figure 4.47 shows the Euclidean distance from the training faces to the real mean and to the miniball centre for the training faces. It is not surprising that the longest distance from a sample to the miniball centre is shorter than the longest distance from a sample to the real mean. However, the average distance to the miniball centre is longer than that to the real mean. This miniball centre is not dependent on the majority of the training faces, but on the face samples along the border of the face cluster. Therefore, including noisy face samples in the training set is not recommended.

In the anything-image-whitened space composed of the top 125 eigenvalue eigenvectors of anything-images, we obtain the miniball centre of 1519 training faces,

which are randomly selected from 4556 training faces. The Euclidean distance from the training images to the miniball centre and the real mean is shown in Figure 4.48.

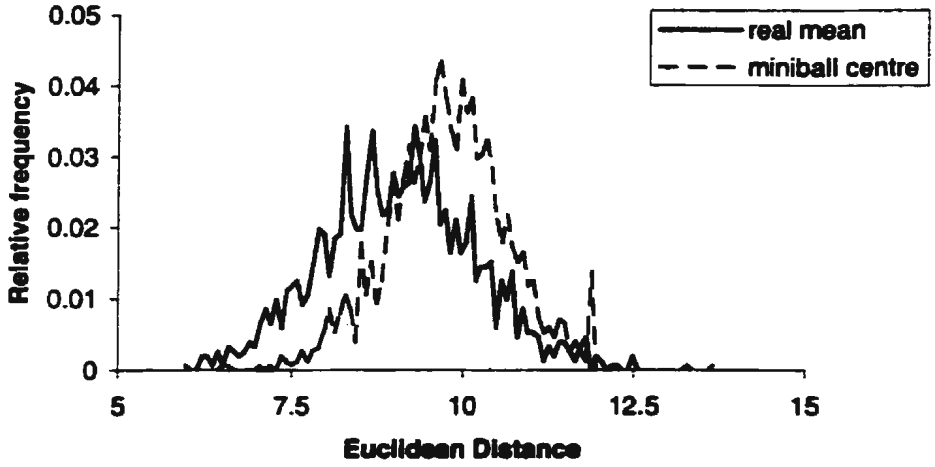


Figure 4.48 In the 125-dimensional anything-image-whitened space the Euclidean distance to the real mean and the miniball centre of training faces

Then we calculate the Mahalanobis distance in the normal way, except using the miniball centre as a replacement for the real mean. The resulting graph is shown in Figure 4.49.

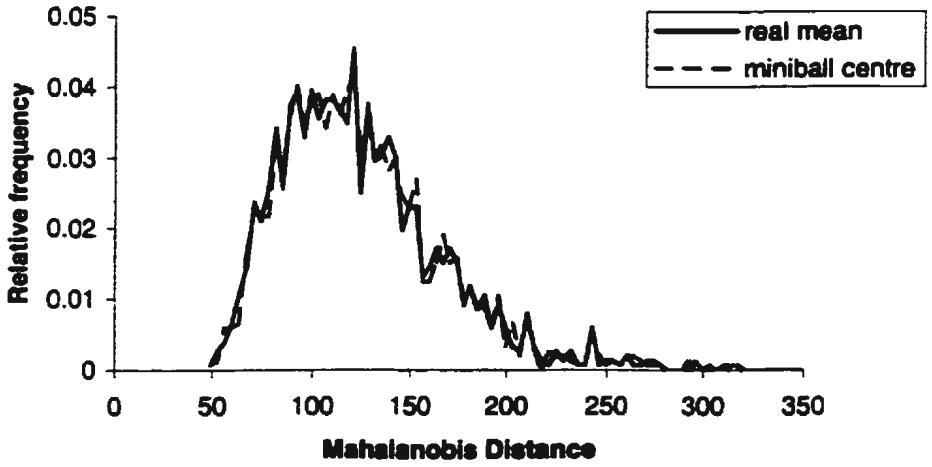


Figure 4.49 In the 125-dimensional anything-image-whitened space the Mahalanobis distance to the real mean and the miniball centre of training faces

Actually the Mahalanobis distance from one face image to the miniball centre is about the same (± 2 %) as its distance to the real mean. This explains why in Figure 4.49 the two curves are nearly identical.

When three other vectors (a vector whose elements are all zeros, a vector whose elements are all ones, and a vector whose elements are all 100s) are used as the substitute for the mean, the resulting Mahalanobis distances are almost the same.

We can draw the conclusion that replacing the sample mean with the centre of the smallest enclosing ball does not improve classification results.

The miniball centre and Mahalanobis distance calculation was repeated on the face images in the face-image-whitened space and the double-whitened space, the above conclusion is also true in these two feature spaces.

Now let's look at the time that the miniball finding program takes on a Pentium II 400M Hz computer.

If the face samples are in the anything-image-whitened space, the elapsed time varies with the number of samples and the number of dimensions as shown in Table 4.15.

Table 4.15 The time that miniball program takes when the number of samples and the number of dimensions vary

# of samples	# of dimensions	Time (minutes)
1000	100	5
1519	125	14
4000	100	200
4000	200	900

On the other hand, if the face images are in the face-image-whitened space, with the restriction to the 1519 samples and 125 dimensions, the time spent is 13 seconds. Please note that Table 4.15 shows in the anything-image-whitened space and with the same dimensionality and samples, the time spent is 14 minutes. The huge difference in time can be explained that in the face-image-whitened space, the face set is compact and spherical, so the miniball centre can be easily calculated.

4.11.3 Minimising the variance of face class while maximising the variance of nonface class

The FLD selects a subspace that maximises the ratio of the determinant of the between-class scatter matrix, $\mathbf{S}_B$, of the projected samples to the determinant of the within-class scatter matrix, $\mathbf{S}_w$, of the projected samples. We now consider a subspace that instead of maximising $\mathbf{S}_W^{-1}\mathbf{S}_B$, minimises $\sigma_f^2 / \sigma_{nf}^2$, where σ_f^2 is the variance of the projected values of faces and σ_{nf}^2 is the variance of the projected values of nonfaces. This

means finding the largest eigenvalue eigenvectors of $\mathbf{S}_f^{-1}\mathbf{S}_{nf}$, where $\mathbf{S}_f$ is the face covariance matrix, and $\mathbf{S}_{nf}$ is the nonface covariance matrix. Once this subspace is found, it might define good directions in which to measure.

In the original space after finding the largest eigenvalue eigenvector of $\mathbf{S}_f^{-1}\mathbf{S}_{nf}$, we project the faces and nonfaces onto it and get the distribution of projections shown in Figure 4.50. The face samples compose a compact cluster in this subspace while the nonface samples spreads out.

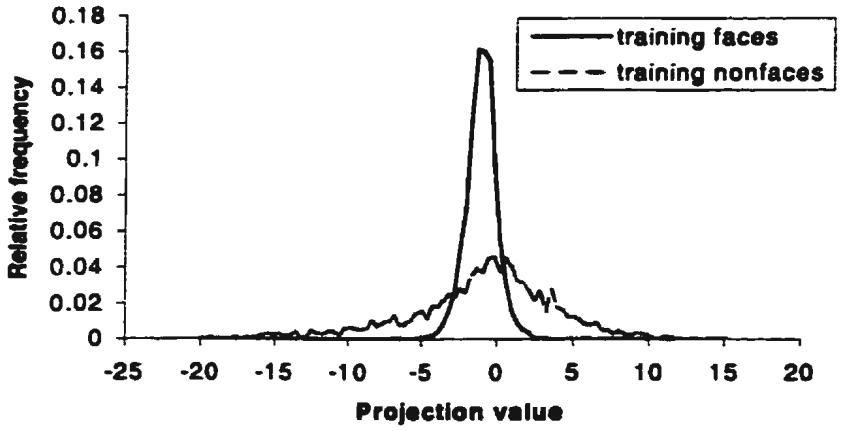


Figure 4.50 In the original space, the projection value onto the largest eigenvalue eigenvector of $S_f^{-1}S_{nf}$

If the 336 eigenvalues of $\mathbf{S}_f^{-1}\mathbf{S}_{nf}$ are sorted in descending order, the eigenvalues after the 237th are less than one in value. Suppose the top M eigenvectors are used to compose a lower-dimensional subspace. All the data sets are projected into this subspace. Then the ML classifier is applied. Figure 4.51 shows the misclassification rate versus the number of dimensions of this subspace.

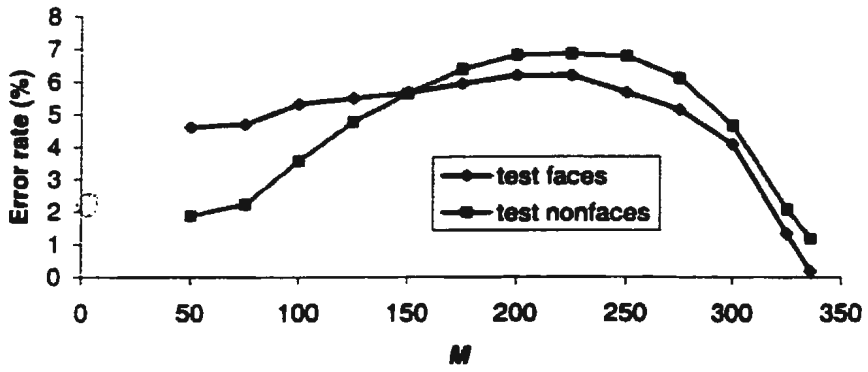


Figure 4.51 Error rates versus number of features

Figure 4.51 shows that the selection of this subspace does not decrease error rate as desired, but increases error rate. When the ML classifier is applied, both the information in the training face set and nonface set is important. If only the largest eigenvalue eigenvectors of $\mathbf{S}_f^{-1}\mathbf{S}_{nf}$ are used to compose the feature space, the information with respect to nonface set is lost.

4.12 Evaluation on Various Classifiers

After performing face/nonface classification tests on various classifiers including the ones proposed here, we evaluate these classifiers in terms of error rates since the classification error is the ultimate measure of the performance of a classifier.

Table 4.16 lists the error rates obtained using various classifiers on 2553 test nonfaces and 1130 test faces. The number of the section where each classifier is presented is also shown.

Table 4.16 Error rates (%) using various classifiers. In a result cell, first row shows the number of dimensions, and second row shows the error rate of test nonfaces and test faces (in this order).

Section	Scheme	Original greyscale space		Anything-image-whitened space		Double-whitened space		Face-image-whitened space	
4.1	Repeated FLD			150D					
				1.6	1.6				
4.4	Hyperellipsoid data modelling, ML classifier			100D					
				0.04	0.36				
4.3	Moving-centre scheme					(100D,100D)*		170D**	
						1.25	0.71	2.35	2.35
4.1	FLD	337D		100D*		(250D, 150D)*		250D*	
		3.53	1.59	3.45	0.88	2.19	1.33	3.29	0.80
4.5	Gaussian distribution, ML classifier			175D*		(250D, 200D)*		250D*	
				0.12	0.27	1.18	0.09	0.04	0.09
4.6	Principal plus complement space, ML classifier					100D		50D*	
						2.27	0.09	0.08	0
4.7	Dominant feature extraction, ML classifier	90D							
		0.39	0.18						
4.8	Nearest neighbour classifier	337D		150D*					
		5.41	0	0.63	0.53				
4.9	k-nearest neighbour classifier	337D($k = 1$)		150D($k = 5$)				150D ($k = 1$)**	
		5.41	0	0.55	0.18			11.8	0.02

4.10	*k / l* nearest neighbour classifier	337D (*k* =15, *l* =14)		150D (*k* = 5, *l* = 4)				150D (*k* = 15, *l* =15)**	
		1.15	1.84	0.24	0.62			1.58	1.56
4.11.1	Euclidean distance to mean face					50D*			
						9.99	9.03		

* The listed number of dimensions is the number that gives the lowest sum of error rates e_4 and e_3.

** The results are the error rates in the training sets, i.e., e_2 and e_1

The classifiers listed in Table 4.16 can be divided into three categories:

1) Linear discriminant

   This category includes the FLD and the repeated FLD. A discriminant hyperplane or several hyperplanes are sought.

2) Hyperquadratic discriminant

   This category includes the ML classifier based on the hyperellipsoid distribution or the Gaussian distribution. These are parametric approaches. Since the Mahalanobis distance uses covariance information, it has the ability to suppress the effect of parameters responsible for within-class variation.

3) Nearest neighbour

   This category includes the nearest neighbour classifier, the *k*- nearest neighbours, and the *k / l* nearest neighbour classifier. These are non-parametric approaches and require an abundance of training samples.

The experimental results in this chapter prove that a hyperquadratic discriminant is preferable to a linear discriminant. Our best result of e_4 = 0.04% and e_3 = 0.09% is achieved by using the ML classifier in the face-image-whitened space. The performance

of the k- nearest neighbour classifier ranks between the hyperquadratic discriminant and the linear discriminant.

In these experiments the number of training samples is at least ten times the number of dimensions. Due to the relatively big training set for the small feature sets, the parameters of hyperquadratic classifiers are fairly reliably estimated. The hyperquadratic classifiers assume normal distribution with different covariance matrices and this attribute makes hyperquadratic classifiers flexible.

FLD is a linear classifier using MSE optimization between the classifier output and the desired labels. It is similar to the Bayes decision rule for Gaussian distributions with identical covariance matrices. However, as shown in Figure 3.21 and Figure 3.22, the face class and nonface have quite different covariance matrices. Moreover, the single hyperplane derived by using FLD is not enough for separating intermingled face class and nonface class. This explains why FLD is outperformed by hyperquadratic classifiers in terms of error rate.

The reason why the nearest neighbour classifiers have higher error rates than hyperquadratic classifiers is probably that we do not have sufficient nonface training samples. Although the set of face samples can be finite, i.e. the possible face images can compose an imaginably finite set, the number of possible nonfaces is infinite. This phenomenon is revealed from the experimental results of k / l nearest neighbour classifier.

Besides the error rate as a measure of the performance of a classifier, other performance measures include the cost of measuring features and the computational requirements of the decision rule. With regard to computational requirements, FLD requires least time, followed by hyperquadratic classifiers and finally the nearest

neighbour classifiers which need the computation of the distances between a test pattern and all the patterns in the training set.

4.13 Summary

In this chapter we proposed two representations and two classifiers whose ability of classifying is demonstrated through face/nonface classification. These four schemes are

- *Repeated FLD*

  The repeated FLD algorithm generates a group of Fisher vectors between two classes. These Fisher vectors are obtained by iteratively reducing the training samples, adding new training samples, or rotating the coordinate system and removing a dimension. These Fisher vectors which are applied to classification sequentially outperform a single Fisher vector. The advantage of the rotate-coordinate-system-and-remove-dimension scheme is that we can progressively tune the method to reject particular kinds of non-faces.

- *Maximum Likelihood classifier based on hyperellipsoid distribution*

  High-dimensional data is usually modelled as a Gaussian distribution. However, the face images take values in a finite range. Consequently, the hyperellipsoid distribution is a better approximation of underlying data. In face/nonface classification, the face class and nonface class are each modelled as having a hyperellipsoid distribution. The ML classifier based on it generates good classification results.

- *Maximum Likelihood classifier based on dominant feature extraction*

  A dominant feature extraction technique is first applied to face/nonface classification. The Fisher vector is extracted as the first feature that preserves all information of class separability caused by mean-difference. Then a subspace

orthogonal to the Fisher vector is found. In this subspace, the dominant eigenvectors of both the face set and nonface set are extracted as other features. The ML classifier based on the feature vectors obtained by using this technique generates satisfying results. The dominant feature technique is superior to the face-image-whitening scheme in terms of number of features required.

- *Moving-centre scheme*

  If one class has large variance but the other class has much smaller variance, the classification boundary is defined by the covariance matrix of the class that has smaller variance. The moving-centre scheme takes advantage of this phenomenon. In face/nonface classification based on Euclidean distance, the centre of the face class is modified through steepest-descent algorithm to find a position where the misclassification rate is the lowest. This scheme is simple in calculation and achieved better results than the FLD.

The proposed classifiers and six existing classifiers, including the FLD, the ML classifier based on Gaussian distribution, the ML classifier in principal plus complement space, 1-nearest neighbour classifier, k-nearest neighbour classifier, and k / l nearest neighbour classifier, are tested in the original greyscale space, anything-image-whitened space, double-whitened space, and/or face-image-whitened space.

In terms of error rate, the probabilistic classifiers perform the best, followed by nearest neighbour classifiers and linear classifiers.

The dimensionality reduction by whitening provides better results. For example, the FLD in the face-image-whitened space is more effective than that in the original space. The dimensionality reduction also eases the problem of singularity in high-dimensional space.

Chapter 5

Optical Flow Used in Representing Face Images

Face images can be characterised directly in terms of pixel intensities. Contemporary face image processing techniques are almost all based on greyscale images. If faces are represented by greyscale only, the face cluster is not convex. The face detection is done by multi-hyperplane [Rowley 1998] or inserting nonface clusters [Sung 1998]. One possible solution is to find a space that is convex. The two crucial elements in this solution are the prototype and the distance measure, which will give a convex space. We propose a possible solution using optical flow as well as deformation.

Optical flow, which captures motion within the face, has been used widely in facial expression analysis directly based on pixel intensities, or on a detailed anatomical and physical model of the face [Mase 1991]. In these applications, optical flow was calculated between various face images of the same person.

However, there are some rigid features common to all face patterns, such as the two dark eyes, the bright nose ridge, and the spatial layout of facial organs. This suggests that a face image of any kind (smiling, tilted, etc.) be represented by a "standard" face, which must have the basic characteristics of all the training samples, and the optical flow between the standard face and the input face.

Moreover, because expression involves motion within the face and pose is the motion of the whole face in a coherent way, a face representation including optical flow might be useful for not only face expression analysis, but also pose estimation and face detection.

5.1 Approach Overview

The proposed approach is aimed to provide a versatile feature space that can be used for all kinds of face-image-processing tasks. The whole process is as follows:

1. Choose an image representation of appropriate size.
2. Define a standard face as the face template. After comparing the later results using various images as the face template, we select the mean of training faces as the template.
3. Represent every image using the measures: motion vectors obtained from optical flow analysis, and deformation residue (difference between the face template and the deformed input). Note that the representation can include other appropriate measures, such as pixel greyscales and edge strength.
4. Perform statistical feature extraction PCA on the large set of measurements to derive a decorrelated space of relatively low dimensionality. In discarding features with low energy, PCA removes noise and less expressive features.

In this feature space the face images compose a convex cluster according to our experiments which will be described later. The convexity of the face cluster greatly facilitates the selection of a classifier.

5.2 Face Representation Including Motion Vectors and Deformation Residue

We used two different sets of training faces to obtain the feature space. Set 1 contains 4556 faces at 19 × 19 pixel resolution, and set 2 contains 1550 faces at 38 × 38 pixel resolution. Here we suggest a face representation based on motion vectors and deformation residue. This approach is illustrated in Figure 5.1.

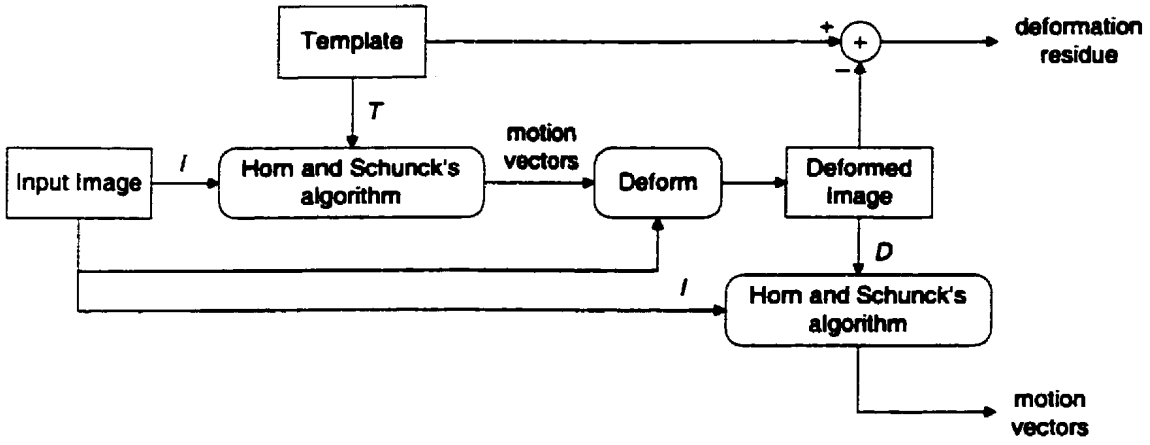


Figure 5.1 Method of generating motion vectors and deformation residue to represent an input image

This representation is obtained in the following steps.

1. Represent an input image *I* in greyscale.
2. Get the mean of a training face set. Define the histogram equalised mean image as the face template, denoted as *T*, as shown in Figure 5.2.

(a) (b)

Figure 5.2 Face templates (a) 19 × 19 pixels (b) 38 × 38 pixels

3. Use Horn and Schunck's algorithm [Horn 1981] to generate the motion vectors from the input image to the face template. Then based on the motion vectors acquire the deformed input image D.
4. Deform the deformed image D back to its original input image I to obtain motion vectors associated with each point in the template. The reason for doing this is that we want the motion vectors anchored in the template. The motion vectors obtained at this step and the deformation residue, $D - T$, are regarded as the combined representation of input image.

Now an input image is represented by the deformation residue and the motion vectors as shown in Figure 5.1. The motion vectors contain pixel wise correspondences between the input image and the deformed input image. The deformed image for an input image is the image in which the grey levels of the image are moved to the corresponding positions in the face template. The motion vectors describe how the 2D shape may change and the deformation residue vector describes how the brightness values may change. The aim of introducing deformation residue is to achieve invariance to small amounts of nonlinear deformations.

The calculation of optical flow and the deformation process are introduced next.

5.2.1 Calculation of optical flow

We used a slightly modified Horn and Schunck's algorithm [Horn 1981, Bässmann 1995] for optical flow calculation. [Bässmann 1995] contains a simplified illustration and implementation of the original algorithm in [Horn 1981].

Let $E(x, y, t)$ denote the image intensity at a point (x, y) in an image at time t. In its adjacent image, if the object to which this pixel is related has moved to another position, the intensity of this pixel does not change. This can be described by

$$E(x, y, t) = E(x + \delta x, y + \delta y, t + \delta t) \tag{5.1}$$

δx, δy, and δt represent the spatial and temporal displacement of the object. Simple mathematical deduction including a Taylor expansion of the right term yields

$$E_x u + E_y v + E_t = 0 \tag{5.2}$$

Here, E_x, E_y, and E_t are the partial derivatives of the image intensity $E(x, y, t)$ with respect to x, y, and t, which can be directly computed from the changes in intensity;

u and v are the partial derivatives of the motion vector at point (x, y) with respect to x and y, i.e., $u = \frac{dx}{dt}$ $\quad v = \frac{dy}{dt}$

E_x, E_y, and E_t are taken from two source images E_0 and E_1:

$$E_x = \frac{\partial E}{\partial x} \qquad E_y = \frac{\partial E}{\partial y} \qquad E_t = \frac{\partial E}{\partial t} \tag{5.3}$$

In practical calculation, the partial derivatives E_x, E_y, and E_t at each pixel are approximated by the masks shown in Figure 5.3.

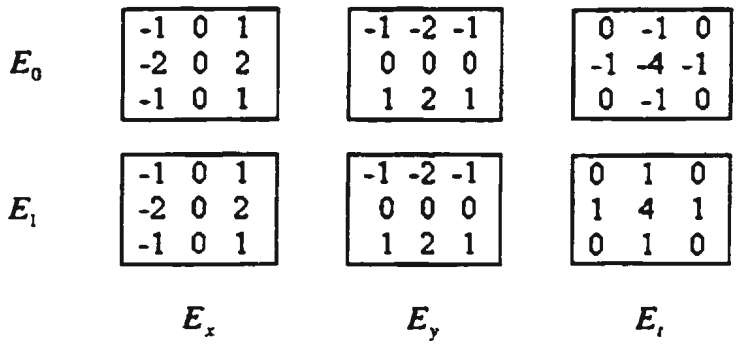


Figure 5.3 Masks to approximate partial derivatives

The partial derivatives E_x, E_y and E_t are computed with the aid of the greyscale differences in a 3×3 neighbourhood. Since there are two images, the differences are computed for each of these images separately. Then the mean of the two resulting differences is utilized as the derivative. Note that we used a different set of masks from those in [Bässmann 1995]. These masks are Sobel operators, which are symmetric about the centre.

Because the motion vector has two components u and v, it cannot be determined locally by only one constraint Equation (5.2). Until now, there have been several different algorithms proposed to estimate the optical flow velocity field by adding other constraints. In our research, Horn and Schunck's algorithm [Horn 1981, Bässmann 1995] which includes a "smoothness constraint" is used. The basic idea behind this constraint is that a point has motion similar to its adjacent points. Hence the motion field changes smoothly in the image. Based on the smoothness constraint, Horn and Schunck used the spatial change of the movement components and thus defined two errors

$$\varepsilon_b = E_x u + E_y v + E_t \tag{5.4}$$

$$\varepsilon_c^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \tag{5.5}$$

These errors are computed for each pixel of the source images, and the overall error is defined as

$$\varepsilon^2 = \iint \left(\varepsilon_b^2 + \alpha^2 \varepsilon_c^2\right) dx dy \tag{5.6}$$

where α is a constant which controls the influence of ε_c on the overall error. The bigger the value of α, the more the emphasis on smoothness.

By minimising the overall error, we obtain the iterative formula:

$$u^{(n+1)} = \bar{u}^{(n)} - \frac{E_x(E_x\bar{u}^{(n)} + E_y\bar{v}^{(n)} + E_t)}{\alpha^2 + E_x^2 + E_y^2} \tag{5.7}$$

$$v^{(n+1)} = \bar{v}^{(n)} - \frac{E_y(E_x\bar{u}^{(n)} + E_y\bar{v}^{(n)} + E_t)}{\alpha^2 + E_x^2 + E_y^2} \tag{5.8}$$

The iteration ends if the number of iterations reaches a certain number or the overall error ε is smaller than a threshold. At the end of this process, the motion field is obtained.

The new values $u^{(n+1)}$ and $v^{(n+1)}$ are obtained following the $(n + 1)$-th iteration from the values ($\bar{u}^{(n)}$ and $\bar{v}^{(n)}$), which are calculated from the results of the preceding iteration ($u^{(n)}$ and $v^{(n)}$).

The values ($\bar{u}^{(n)}$ and $\bar{v}^{(n)}$) are weighted means. The weights are shown in Figure 5.4. Note that the centre pixel is not included in the calculation of $\bar{u}^{(n)}$ and $\bar{v}^{(n)}$.

1/12	1/6	1/12
1/6	-1	1/6
1/12	1/6	1/12

Figure 5.4 The Laplace operator

When we compute the optical flow between two arbitrary face images, these two images are treated in the same way as two consecutive frames of an image sequence. In this thesis, when it is said that the optical flow is from image A to image B, it means that image A and B act as E_0 and E_1 respectively in the optical flow calculation.

In this work, the initial values, $u^{(0)}$ and $v^{(0)}$, are set to zero. α is set to 50. The program stopped when the number of iterations reached 100.

5.2.2 Deformation algorithm

The process of moving every pixel in an image along the motion vectors is called *deformation*.

Assume two images are denoted as S and T respectively. $t(i, j)$ is the intensity of T at point (i, j). Likewise, $s(i, j)$ is the intensity of S at point (i, j). i and j are integers. We generate the motion vectors from S to T. If the motion vectors in horizontal and vertical directions at (i, j) are denoted as $m_u(i, j)$ and $m_v(i, j)$, then a point (i, j) in S corresponds to a point $(i + m_u(i, j), j + m_v(i, j))$ in T. By pulling the points of S along the motion vectors, we obtained the deformed image, which will tend to look like T in appearance.

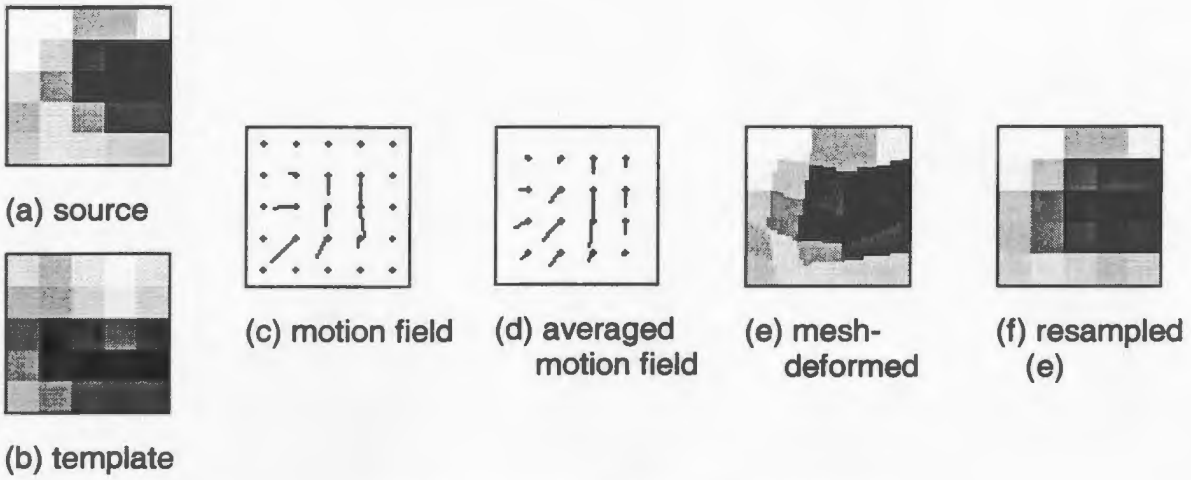


Figure 5.5 The deformation process

Suppose we have two $n \times n$ ($n = 5$) pixel images as shown in Figure 5.5a and b. The whole deformation process is as follows.

1. Compute the motion field when Figure 5.5a is regarded as the image S and Figure 5.5b is regarded as the image T. The resulting motion field is displayed in Figure 5.5c.

   The small needles in Figure 5.5c indicate the magnitude and direction of motion vectors between these two images.

2. Calculate the average motion among every four adjacent points in the motion vector picture and mark the average motion at the centre, which we call the pixel midpoint, of those four points. The resulting motion vector picture is shown in Figure 5.5d.

3. Enlarge the original source image Figure 5.5a by a factor (we use 10), and pull the midpoints along the averaged motion vectors. This process leads to the mesh-deformed image Figure 5.5e.

4. Equally divide the mesh-deformed image into $n \times n$ blocks and get the average greyscale of each block. Thus we obtain the deformed image, denoted as D, in Figure 5.5f.

Comparing the deformed image Figure 5.5f with the source image Figure 5.5a, we see that the dark part in the source image has been moved down to make it resemble the template image Figure 5.5b.

Note that the images in Figure 5.5a, b, and f are of size 5×5 pixels, while the image in Figure 5.5e is of size 50×50 pixels.

5.2.3 Resulting motion vectors and deformation residue for an image

The optical flow analysis and deformation algorithm are applied to face images of two resolutions.

- 19×19 pixel resolution

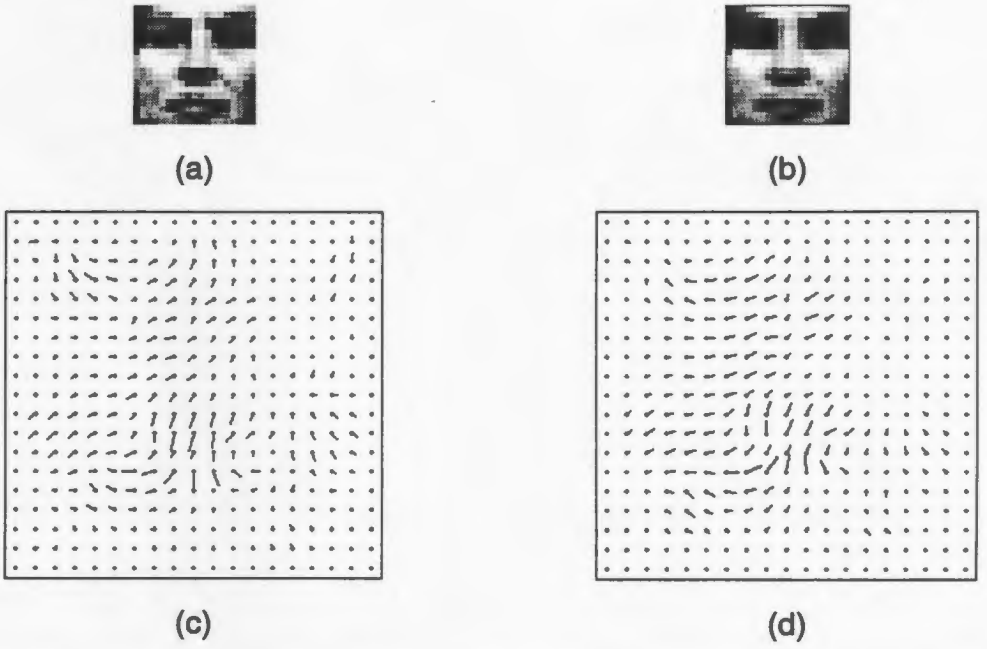


Figure 5.6 (a) an input image of 19×19 pixel resolution, (b) the deformed input image, (c) the motion field from (a) to the face template Figure 5.2a, (d) the motion field from (b) to (a).

Figure 5.6a shows an input image, and Figure 5.6c shows the motion field from this input image to the face template. The motion vectors are shown with heads at the grid points of pixels in the face sample, and tails at the corresponding points in the template. These motion vectors indicate how a particular image can be adjusted to conform best to the face template. Figure 5.6c shows substantial motion in the nose area. The motion vectors along the four borders are always zero. Because the masks in Section 5.2.1 are symmetric, the needle image between one image and the template is exactly the flipped version of the needle image between its flipped image and the template.

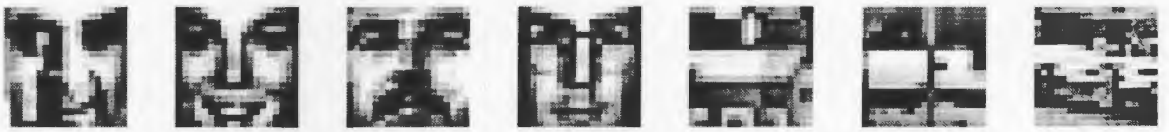
Using the deformation method, we deform the input image and generate the deformed image in Figure 5.6b. The deformed input face becomes more similar in appearance to the face template than the input image.

Figure 5.6d shows the motion field from the deformed input image to the input image. Now the input image can be represented by the deformation residue, the greyscale difference between the deformed image (as in Figure 5.6b) and the face template (as in Figure 5.2a), or the motion vectors (as in Figure 5.6d) from the deformed image to the input image.

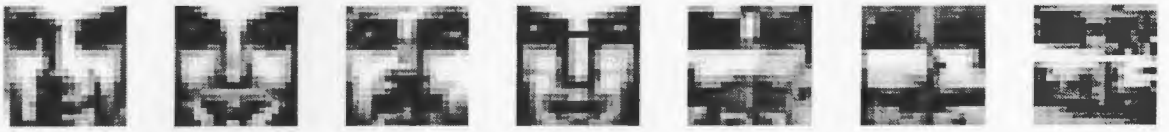
To remove the effect of background, the corner pixels are removed from the deformation residue part; both the corner pixels and border pixels are removed from the motion vector part. For 19×19 pixel images, the number of remaining dimensions of the deformation residue part is 337, and the number of remaining dimensions of the motion vector part is $285 \times 2 = 570$.

Figure 5.7a shows four faces and three non-faces in our data set for the 19×19 pixel case. These face images includes smiling faces, faces with heavy shadow,

moustache, or eyeglasses. Figure 5.7b shows their deformed images. By and large, the deformed images tend to resemble the face template. Although the non-faces become more like face too, the underlying motion vectors are not structured in a way that the motion vectors of faces are. This will be exhibited in our face detection experiments in the next chapter.



(a) the original images



(b) the deformed images

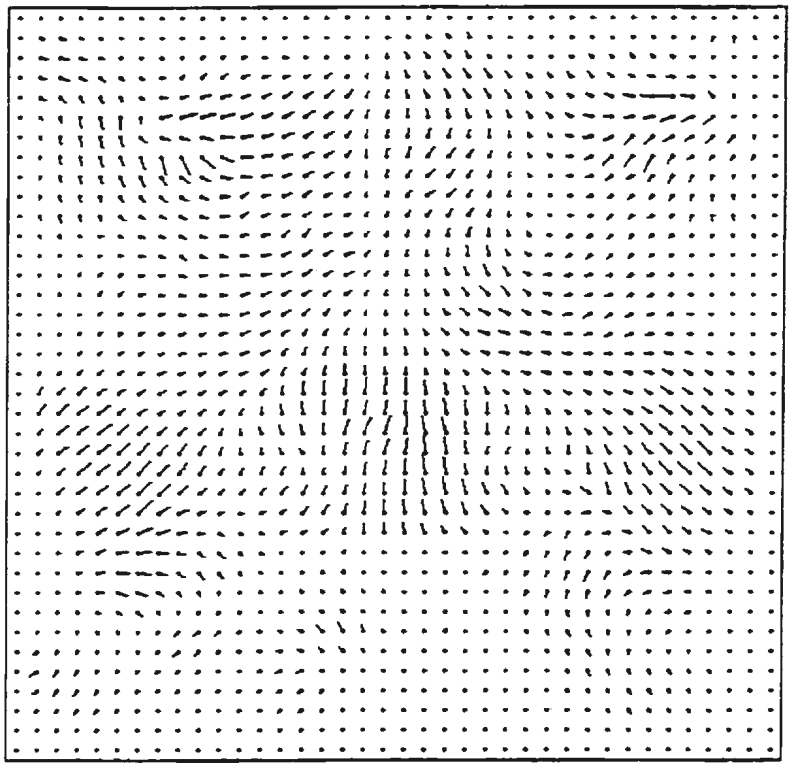
Figure 5.7 Deformed faces and nonfaces

- 38×38 pixel resolution

  Figure 5.8 shows a 38×38 pixel input image, the deformed input image and the calculated motion field from the deformed image to the input image.

(a)

(b)



(c)

Figure 5.8 (a) an input image of 38 × 38 pixel resolution, (b) the deformed input image, (c) the motion field from (b) to (a)

Note that the input image in Figure 5.8a is the same as that in Figure 5.6a except for size. Therefore, the needle image Figure 5.8c shows the similar motion to that in Figure 5.6d. For example, upward motion in the nose area is observed in both needle images.

For 38 × 38 pixel images, after the corner pixels are removed, the number of remaining dimensions of the deformation residue part is 1360; after the corner and border pixels are removed, the number of remaining dimensions of the motion vector part is $1256 \times 2 = 2512$.

5.3 Feature Space

In the single space composed of the concatenated motion vectors and deformation residue vectors, PCA is performed to derive a decorrelated space of relatively low dimensionality. In discarding features with low energy, PCA ensures that wrong assumptions about the importance of measures are suppressed. This PCA derived space is the feature space that will be used in face-image processing experiments in the next chapter.

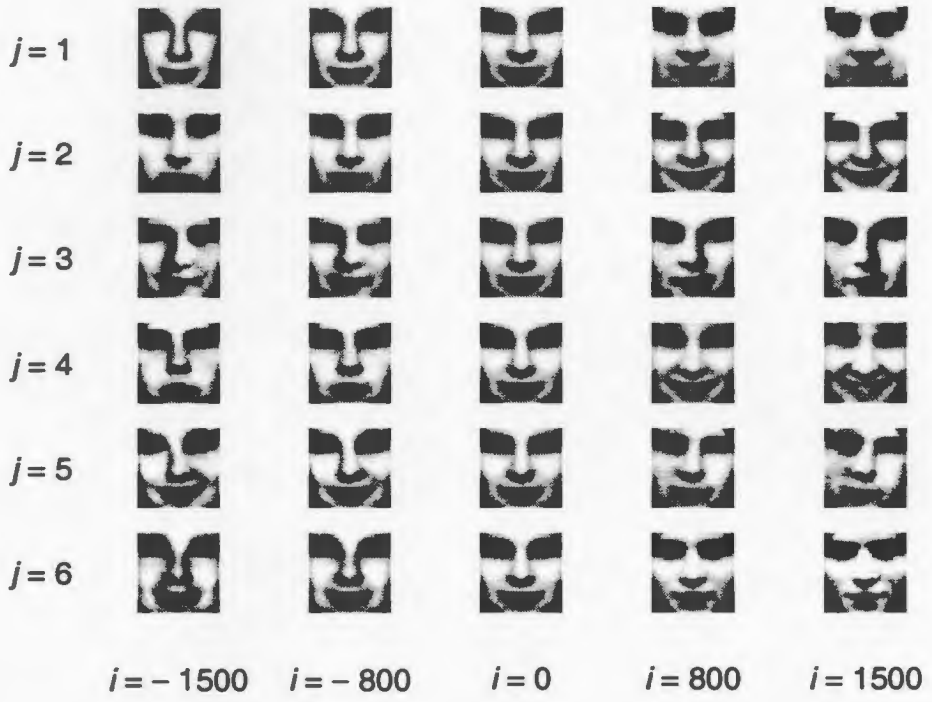


Figure 5.9 Images generated by morphing the mean face along the first 6 eigenvectors of the face space. *j* is the serial number of eigenvectors, and *i* is a value used to generate the morphed images

The faces associated with the six highest eigenvalue eigenvectors for the 38 × 38 pixel case are shown in Figure 5.9. Videos showing movement along these eigenvectors are available at http://www.engr.mun.ca/~qing/video.

One morphed image is generated by deforming a greyscale image ($T + \phi_{rj} \times i$) by the vectors $\phi_{mj} \times i$ where T is the greyscale of the face template, ϕ_{rj} is the residue part of the j-th eigenvector and ϕ_{mj} is the motion vector part of that eigenvector.

Note that these top eigenvectors have been selected for certain kinds of variation in the training set: the first eigenvector tracks gross shape, the third, lighting direction, and the fifth, tilt. On the other hand, facial expression - specifically, smiling - is tracked by both the second and fourth eigenvectors (in the latter case, it is correlated with nose shape!). Interestingly enough, the gender is tracked by the fifth eigenvector too.

The images in Figure 5.10 are generated by using the first two eigenvectors and the face template. One morphed image is generated by deforming a greyscale image ($T + \phi_{r1} \times i_1 + \phi_{r2} \times i_2$) by the vectors $\phi_{m1} \times i_1 + \phi_{m2} \times i_2$. The horizontal direction shows images deformed along the first eigenvector, while the vertical direction shows the changes along the second eigenvector. It is encouraging that all the morphed images in Figure 5.10 look like real faces and cover a wide variation in both face expression and shape.

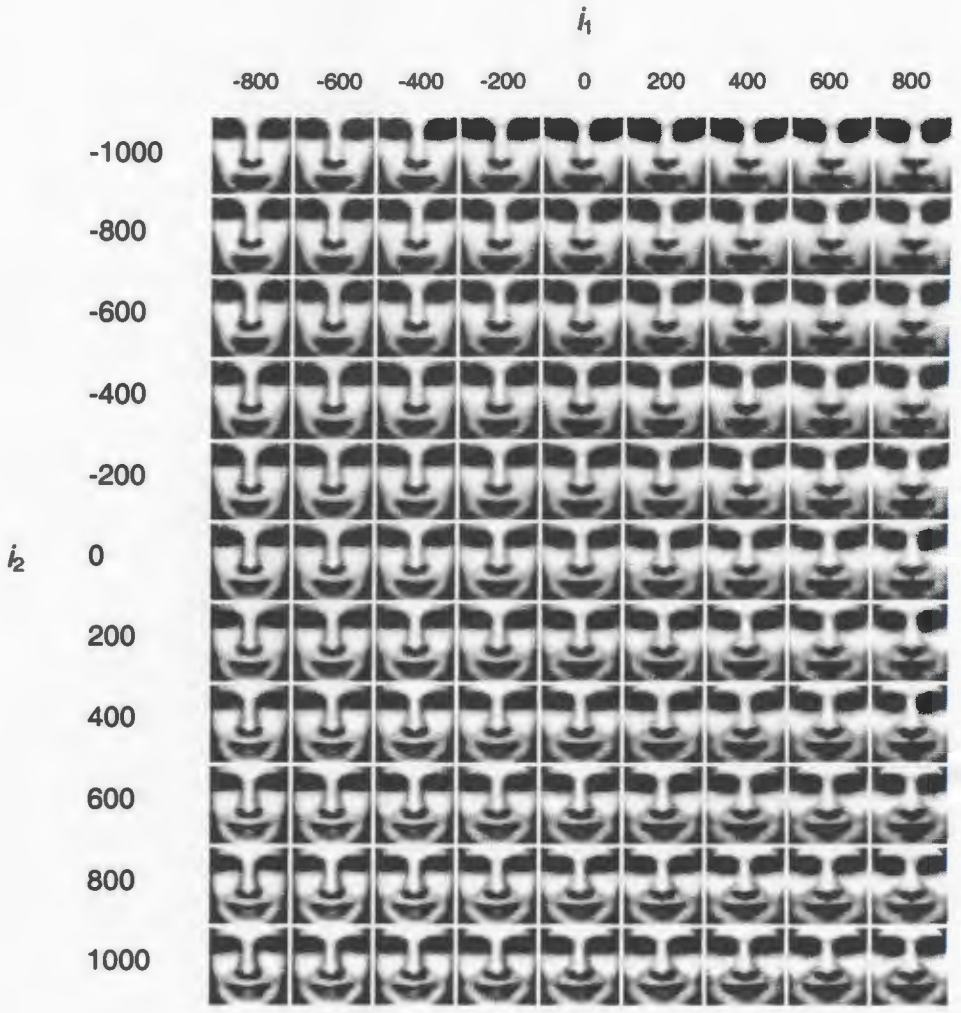
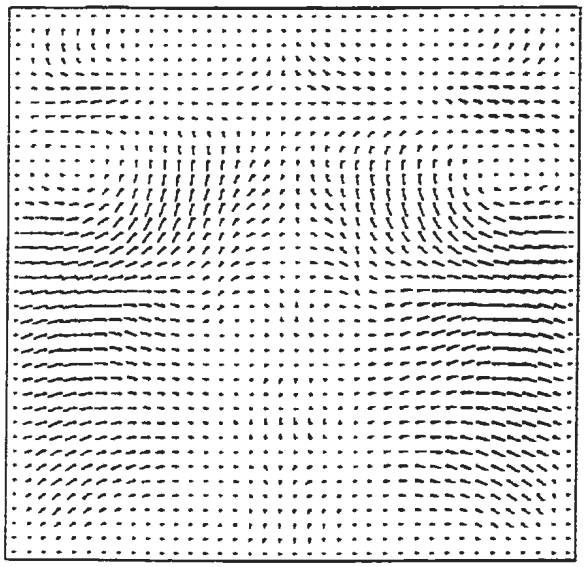
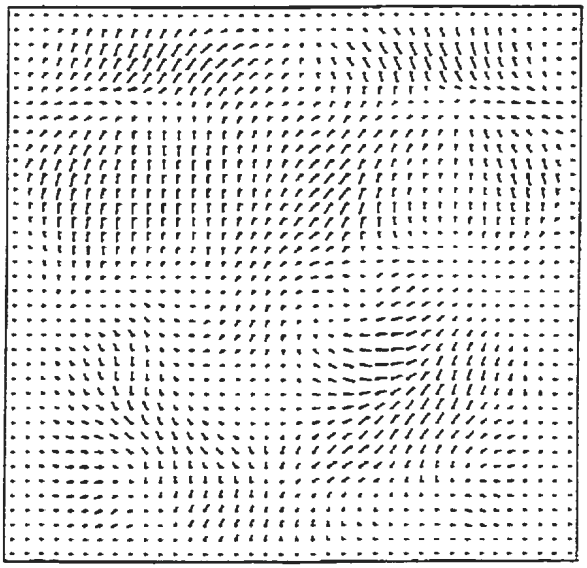


Figure 5.10 Morphed Images using the first two eigenvectors

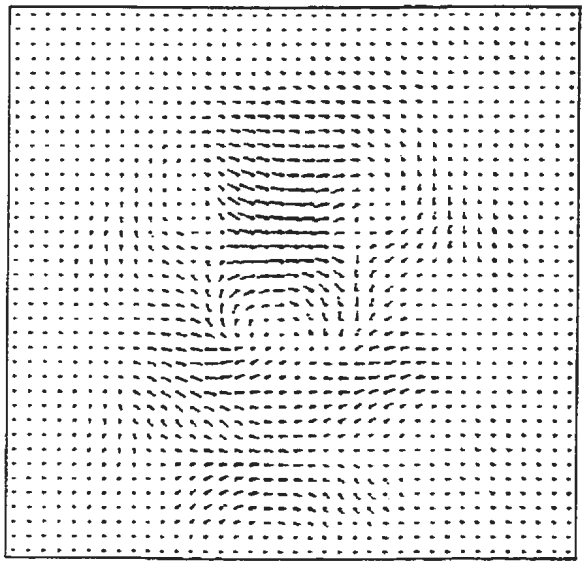
Figure 5.11 gives us a close look at the motion vector part of the top six eigenvectors which are used to generate the morphed images in Figure 5.9.



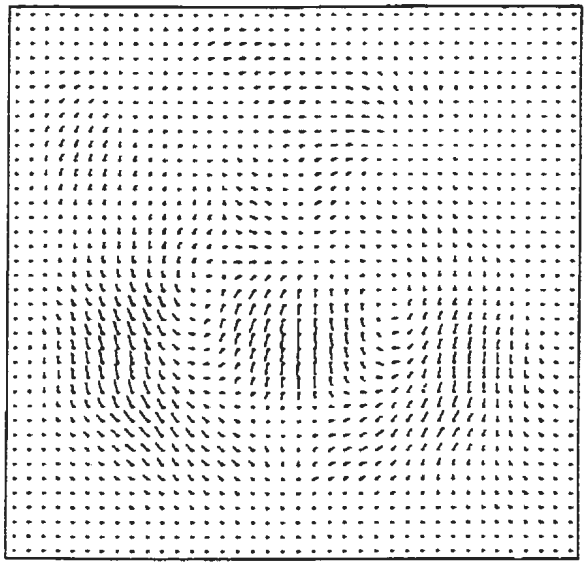
(a) ϕ_{m1}



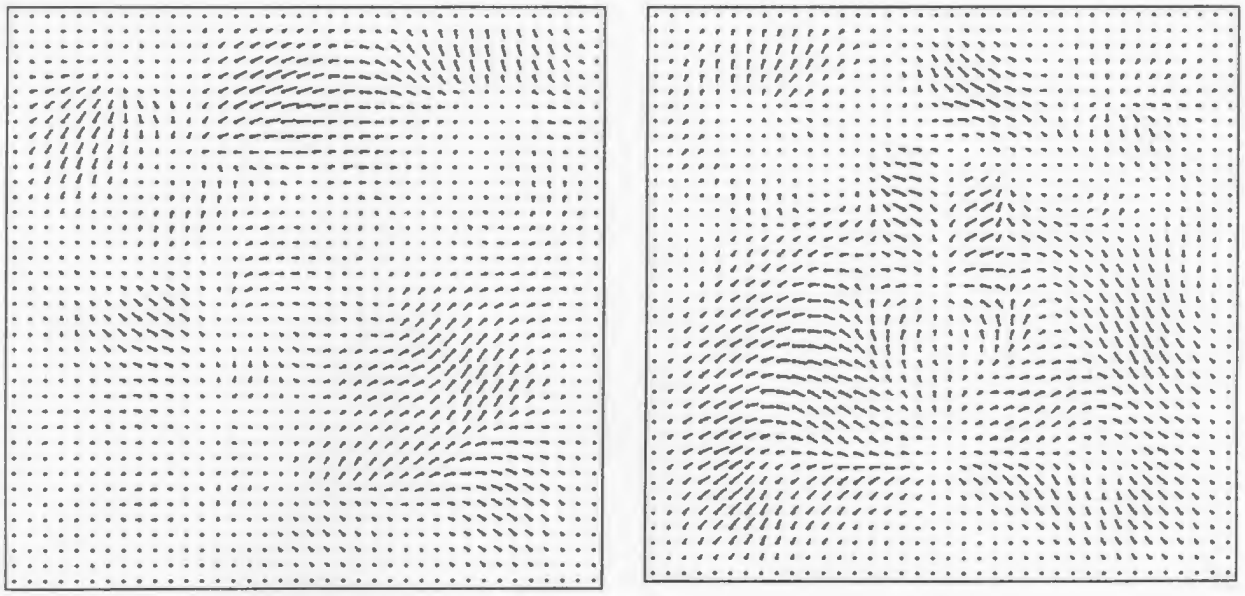
(b) ϕ_{m2}



(c) ϕ_{m3}



(d) ϕ_{m4}



e) ϕ_{m5} f) ϕ_{m6}

Figure 5.11 Needle images showing the motion vector part of the first six eigenvectors

We can see clearly how the pixels in the cheek area move up to give a smiling expression. In addition, in Figure 5.11d, the nose pixels are moving down while the cheek pixels move up. This explains the reason that in Figure 5.9 the fourth eigenvector tracks both the nose shape and the smiling expression.

Figure 5.12 exhibits the deformation residue part of the first six eigenvectors. ϕ_{r1} represents the eigenvector associated with the largest eigenvalue, and so on. Clearly, the variation in the residue part in each eigenvector matches its motion vector part.

(a) ϕ_{r1} (b) ϕ_{r2} (c) ϕ_{r3} (d) ϕ_{r4} (e) ϕ_{r5} (f) ϕ_{r6}

Figure 5.12 Deformation residue part of the first six eigenvectors

It is worthwhile to explore the effect of the motion vector part of eigenvectors on morphing the face template. The images in Figure 5.13 are generated by deforming a greyscale image ($T + \phi_{r5} \times i$) by the vectors $\phi_{m5} \times k$, where ϕ_{r5} and ϕ_{m5} are the deformation residue part and motion vector part of the 5th eigenvector respectively. The condition is that i is fixed at 1500, and k changes from −3000 to 3000 at a step of 1500.

(a) $k = -3000$ (b) $k = -1500$ (c) $k = 0$ (d) $k = 1500$ (e) $k = 3000$

Figure 5.13 Effect of the motion vector part of the 5th eigenvector on morphing the face template

Note that the image in Figure 5.13d is the same as the image in the condition of $j = 5$ and $i = 1500$ in Figure 5.9. From Figure 5.13a to Figure 5.13e, the variation between the morphed image and the face template increases gradually. Figure 5.13a is the one which looks most like the face template, but Figure 5.13e is the one least like the template. This tells that when k is negative, the motion vector part compensates for the variation caused by the deformation residue part in morphing. When k is positive, the motion vector part and the deformation residue part work together to increase the variation. The image in Figure 5.13a looks like the face template because the effect of the deformation residue part of the eigenvector is offset by the motion vector part.

5.4 Convexity of Space

The convexity of a measurement space is what we have sought for. We now examine the convexity of the proposed space composed of the motion vectors and deformation residue without PCA. The convexity of a measurement space is tested by taking random

pairs of face images and finding their mean in the measurement space. If the mean images look like a face, then the measurement space is convex.

Figure 5.14 shows the means of image pairs when images are represented by original scales, and when images are represented by the motion vectors and deformation residue.

Image pair					
Mean image in the original greyscale space					
Mean image in the motion vectors and deformation residue space					

Figure 5.14 Means of image pairs in the original greyscale space and in the motion vectors and deformation residue space

The mean images of image pairs in the proposed space are more smooth and face-like than those in the original greyscale space. These mean images are solid evidence that the proposed space is more convex than the original greyscale space.

The leftmost pair of face images shows 20° of rotation. Their mean image in the motion vector and deformation residue space is not quite face-like, especially in the eye areas. To make the representation space more convex, another method of calculating motion vectors is proposed. The Horn and Schunck's algorithm in Figure 5.1 is replaced by the whole process shown in Figure 5.15.

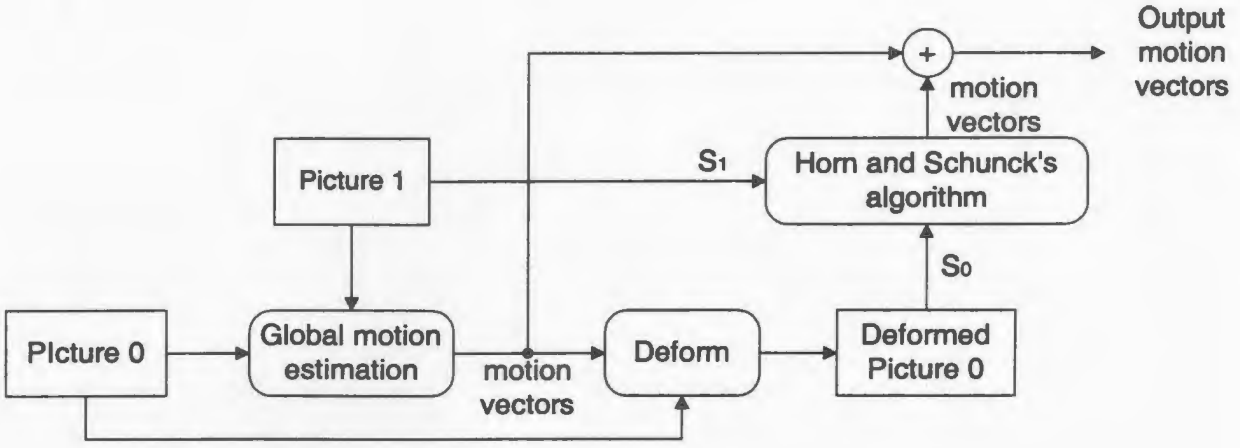


Figure 5.15 Generation of motion vectors including global motion

The "Output motion vectors" between two input images "Picture 0" and "Picture 1" are the sum of the global motion and the local motion. The global motion estimation program estimates the global translation, rotation, and independent scaling in horizontal and vertical directions. Local motion is generated by Horn and Schunck's algorithm.

Using the motion vectors including global motion and thus generated deformation residue, we obtain the mean images, as shown in Figure 5.16, of the above image pairs.

Figure 5.16 Means of image pairs in the motion vectors (including global motion) and deformation residue space

The leftmost image becomes more like the real face. The quality of the other four mean images is also better than that in Figure 5.14.

The method of representing images by the motion vectors including global motion is used in the face detection and face recognition experiments described in the next chapter.

5.5 Reconstruction of Face Images

It is important to note that the representation we achieve is reversible - a given face image can be reconstructed from its motion vector and deformation residue parameters.

The projection of a vector $\mathbf{x}$ into the eigenspace is

$$y_i = \phi_i^T (\mathbf{x} - \mathbf{m}) \tag{5.9}$$

where $i = 1, ..., M$. M is the number of dimensions of the eigenspace. $\mathbf{m}$ is the mean of samples from which the eigenspace is derived. ϕ_i is the i-th largest eigenvalue eigenvector of the covariance matrix of samples.

The reconstructed vector in the original space is

$$\mathbf{z} = \left(\sum_{i=1}^{M} y_i \phi_i \right) + \mathbf{m} \tag{5.10}$$

If $\mathbf{z}$ is compared with $\mathbf{x}$, the amount of reconstruction error can be estimated.

In the preceding section, an image is represented by a vector $\mathbf{x}$ consisting of two parts: a motion vector part and a deformation residue part. We project the vector $\mathbf{x}$ into the feature space obtained by performing PCA on training faces. The reconstructed image is generated by deforming the image $(T + \mathbf{z}_d)$ by motion vectors $\mathbf{z}_m$, where T is the face template, $\mathbf{z}_m$ and $\mathbf{z}_d$ correspond to the motion vector and deformation residue part of the reconstructed vector $\mathbf{z}$. The image size is 38×38 pixels. Let M be 500.

(a) (b) (c)

Figure 5.17 Original image and reconstructed images. (a) original image, (b) reconstructed image with not-scaled deformation residue, (c) reconstructed image with scaled deformation residue.

The value range for the motion vectors is [-2, 2], but the value range for the deformation residue is [-255,255]. If the image vector is directly represented by the combination of the motion vector part and the deformation residue part, the reconstructed image of Figure 5.17a is Figure 5.17b. The $\mathbf{z}_m$ part is very small (maximum 0.2) and negligible. If the image vector is represented by the motion vector part and the scaled deformation residue part (scale is 1/100), the reconstructed image is shown in Figure 5.17c. In this case, the maximum value of $\mathbf{z}_m$ part is 1.2, so it does contribute to deformation.

Figure 5.17c is smoother than Figure 5.17b.

5.6 Selection of the Face Template

As manifested in the previous sections in this chapter, the selection of the face template is crucial to the subsequent calculation of motion vectors and deformation residue. Our face template, shown in Figure 5.2, is simply the mean face of thousands of face images. However, it might not be the optimal template. We have tried replacing this template with other faces, for example,

1) the actual face image from the training set that is closest to the mean template in Euclidean distance in terms of pixel intensities,
2) the high frequency emphasised mean face,
3) the mean face of dozens of face samples with same face expression, such as smiling,
4) the mean face of thousands of deformed face images (this step can be repeated iteratively).

These solutions are implemented and compared. The comparison is done by deforming every image in a face set (435 images of 19 by 19 pixels) to the new template and getting the average of mean squared error (MSE) between the new template and the deformed images. The image that has lowest average MSE over all other images deformed to it should be chosen as the template.

The average MSE for some of the different templates are listed in Table 5.1.

Table 5.1 Average MSE between a template and images deformed to it

Template	MSE
The histogram equalised mean of all the images	1537.6
The high frequency emphasised version of above	2004.2
The face image closest to the mean in MSE	2066.7
The image second closest to the mean in MSE	2399.0

The results show that the simple mean of all the face samples achieved the minimum MSE. Until now, no other better substitute for this template has been found.

5.7 Summary

The chapter describes an approach to generating a face feature space that is convex.

At first, the mean of a large set of face samples is selected as a face template.

Then we represent the relationship between members of a training set and the face template using motion vectors obtained through optical flow analysis and deformation residue (difference between the face template and the deformed input image).

Horn and Schunck's algorithm is used for the calculation of motion vectors, which indicate how a particular image can be adjusted to conform best to another.

The process of pulling the pixels of an image along its motion vectors is called deformation. The deformation is performed by calculating the average motion vectors among four surrounding pixels, generating mesh deformed image, and re-sampling.

Finally, the selection of features is then done in a conventional statistical way using PCA applied to all the measurement dimensions. The principal components capture the outstanding variations across the training set.

The convexity of the obtained feature space is tested and demonstrated. Another set of motion vectors that include global motion is described and used in convexity tests.

Chapter 6

Experiments

In order to assess the viability of the proposed representation of face images in a focused way, a number of experiments are performed on face detection, expression analysis, pose estimation, and face recognition. The computer programs performing all these tasks are developed in Powersoft Power++, MATLAB, or C/C++ in Windows or Unix environment.

6.1 Classifying Face and Nonface Images

Experiments on face/nonface discrimination are chosen because face/nonface discrimination underlies face finding. In addition, existing methods for this tend either to use many linear discriminants [Rowley 1998], or a complicated non-convex space trained by insertion of nonface clusters within face clusters [Sung 1998]. The following experimental results demonstrate that this task can be simplified by only using one linear classifier or hyperquadratic classifier.

The classifications are performed on 19×19 pixel images and 38×38 pixel images. Three types of representation (original greyscale, motion vectors, deformation residue) are used individually, or in combination. The original greyscales take values in the range $[0, 255]$. The deformation residue takes values in the range $[-255, 255]$.

Every classification experiment requires four data sets: training faces, training nonfaces, test faces, and test nonfaces. All the four sets of 19×19 pixel images are

identical to the sets used in Chapter 4 except that no normalisation is applied to let an image have a zero mean and unit variance. The training face set and test face set of 38 × 38 pixel images are also identical to the sets used in Chapter 4 except for size, while the training nonfaces and test nonfaces are different.

The results of using the FLD and ML classifier for face/nonface classification are presented.

6.1.1 FLD

We compared the classification results obtained by the FLD when every image is represented by the original greyscales, motion vectors, deformation residue, or the combination of motion vectors and deformation residue.

Two methods are used to obtain the discriminant for an unknown sample $\mathbf{x}$.

1) Holistic method

   Regarding the available representations as a whole for every image, we derive the single Fisher vector between training faces and training nonfaces, and then project the sample $\mathbf{x}$ onto the Fisher vector.

2) Collection method

   Regarding the horizontal components of motion vectors, vertical components of motion vectors, and the deformation residue of an image as independent of each other, we derive a Fisher vector W_i and the separation point w_i from the training faces and nonface in the i-th representation space. The discriminant for sample $\mathbf{x}$ in the i-th space is

$$d_i(\mathbf{x}) = W_i^T \mathbf{x} + w_i \tag{6.1}$$

   Therefore, the final discriminant is

$$d(\mathbf{x}) = \sum_{i=1}^{n} d_i(\mathbf{x}) \tag{6.2}$$

where n is the number of available spaces.

Table 6.1 presents the number of misclassified images in the test sets based on these four different representations. If n is specified, the collection method is used. Otherwise, the holistic method is used.

Table 6.1 Number of misclassifications using the FLD in a test set containing 2553 nonfaces and 1130 faces with a resolution of 19 × 19 pixels

Representation	Misclassified nonfaces	Misclassified faces
Original greyscales	75	12
Motion vectors	29	11
Motion vectors ($n = 2$)	44	16
Deformation residue	34	3
Motion vectors and deformation residue	14	2
Motion vectors and deformation residue ($n = 3$)	20	1

This table shows that if images are represented by motion vectors or deformation residues alone, the classification results are better than those using greyscale representation. Moreover, if motion vector representation and deformation residue representation are used together, the results are much better. This proves the effectiveness of the new representation of images. The holistic method is slightly better than the collection method in this condition. Although the motion vectors and deformation residue representation increases the dimensionality of feature space, the substantial improvement on error rates makes the effort worthwhile.

Table 6.2 presents the results on 38 × 38 pixel images.

Table 6.2 Number of misclassifications using the FLD in a test set containing 1516 nonfaces and 1130 faces with a resolution of 38 × 38 pixels

Representation	Misclassified nonfaces	Misclassified faces
Original greyscales	67	3
Motion vectors ($n = 2$)	41	2
Deformation residue	29	1
Motion vectors and deformation residue ($n = 3$)	15	0

The representation of motion vectors and deformation residue again performs the best. The error rate in the larger pictures is lower than that in the smaller pictures. This observation is consistent with other researchers' results that the bigger the search-window size, the lower the error rates. Nevertheless, those better results are achieved at the expense of computation time and storage space. A 38 × 38 pixel image occupies four times of the storage space of a 19 × 19 pixel image. The computation time of classification on 38 × 38 pixel images is also roughly four times of that on 19 × 19 pixel images.

In the preceding chapter, the motion vectors including global motion were described. Using these motion vectors and hence obtained deformation residue, we do the face/nonface classification and get the results in Table 6.3.

The results in Table 6.3 are worse than those shown in Table 6.1. The reason for the poor performance of the motion vector (including global motion) representation is that these motion vectors are the sum of the global motion vectors and local motion vectors. In some cases, the global motion is more significant than local motion. While local

motion is specific to the structure of an image, the global motion only accounts for the orientation of that image. Consequently, when the global motion dominates, faces and nonfaces cannot be discriminated by the motion vectors. The explanation of the relatively poor performance of deformation residue representation is that because of the global motion included, the deformed images are rotated to be upright. Thus the nonface images look more like faces, which makes the classification problem harder.

Table 6.3 Number of misclassifications using the FLD in a test set containing 2553 nonfaces and 1130 faces with a resolution of 19 × 19 pixels. Global motion is included.

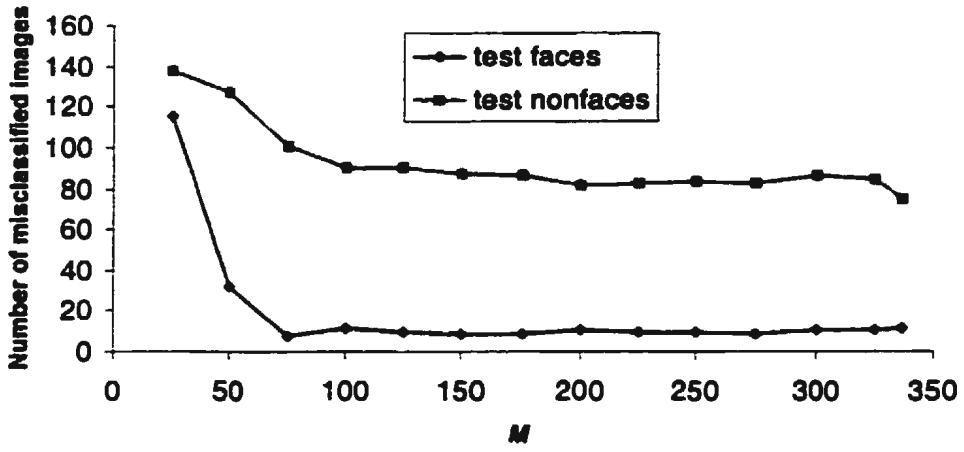
Representation	Misclassified nonfaces	Misclassified faces
Motion vectors	48	13
Deformation residue	59	15

In the following sections, the motion vectors do not include global motion unless specified.

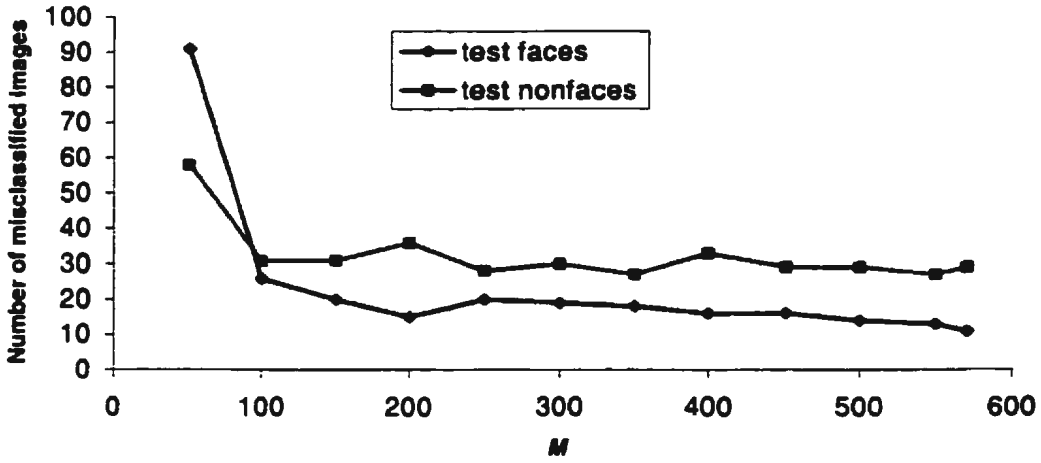
6.1.2 PCA plus FLD

We attempt to improve the results presented in the preceding section by extracting the eigenvectors of the representation space of the training faces and then performing the FLD in the subspace composed of the M largest eigenvalue eigenvectors.

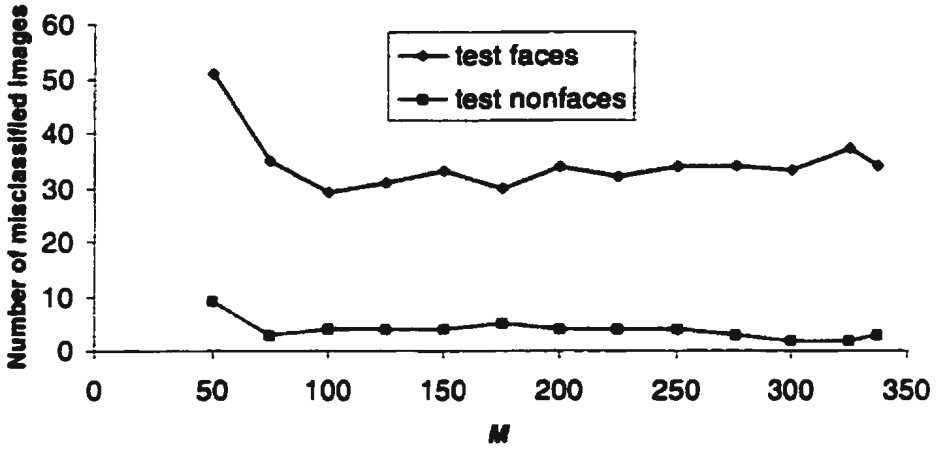
The face/nonface classification is first performed on 19 × 19 pixel images which are represented by the original greyscales, motion vectors, deformation residue, and the combination of motion vectors and deformation residue. Only the holistic method is adopted. The number of misclassified test images versus M is shown in Figure 6.1.


Number of misclassified images
160
140
120
100
80
60
40
20
0
test faces
test nonfaces
0
50
100
150
200
250
300
350
M

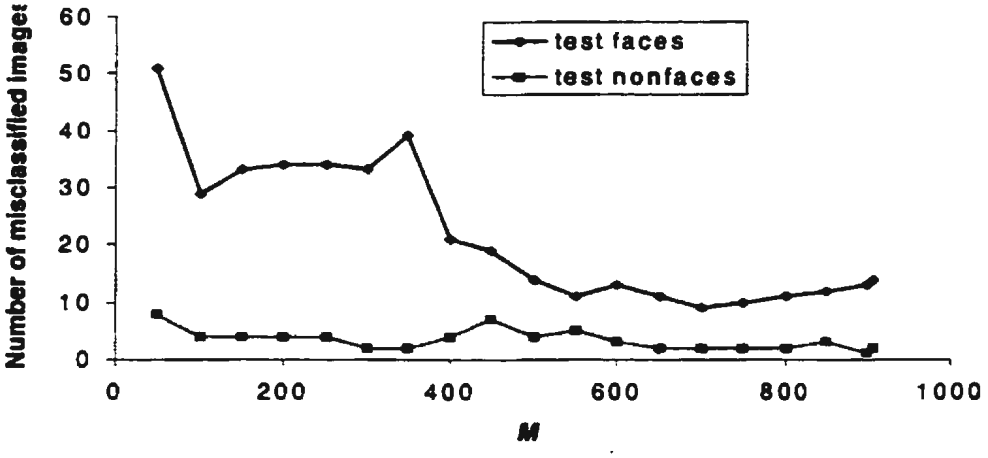
(a)


Number of misclassified images
100
90
80
70
60
50
40
30
20
10
0
test faces
test nonfaces
0
100
200
300
400
500
600
M

(b)



(c)



(d)

Figure 6.1 Using the FLD, the number of misclassified test images versus the number of dimensions of eigenspace when images are represented by the (a) original greyscales, (b) motion vectors, (c) deformation residue, (d) motion vectors and deformation residue

This figure shows that when M is greater than one third of total available dimensions, the number of misclassifications remains steady, i.e., the PCA does not improve the

results significantly. In Figure 6.1c, the best result of 29 misclassified test faces and 4 misclassified test nonfaces is achieved at $M = 100$. In Figure 6.1d, the best result of 9 misclassified test faces and 2 misclassified test nonfaces is achieved at $M = 700$.

Then we perform face/nonface classification using the PCA plus FLD on 38×38 pixel images. The number of dimensions for the original greyscale or deformation residue representation is 1360, while the number of dimensions for the horizontal or vertical components of motion vector representation is 1256. When the holistic method for the FLD is used, we extract the top 500 eigenvectors from the training faces and compose a 500-dimensional subspace and then derive a single Fisher vector. When the collection method is used, we take the top 500 eigenvectors in each representation space (original greyscale, horizontal component of motion vectors, vertical component of motion vectors, and deformation residue), and get the Fisher vector for each space. The face/nonface classification results are listed in Table 6.4.

Table 6.4 Using the PCA plus FLD, number of misclassifications (%) in a test set containing 1516 nonfaces and 1130 faces with a resolution of 38×38 pixels.

Representation	Misclassified nonfaces	Misclassified faces
Original greyscales	25	3
Motion vectors	26	14
Motion vectors ($n = 2$)	13	2
Deformation residue	7	1
Motion vectors and deformation residue	1	4
Motion vectors and deformation residue ($n = 3$)	2	1

These results are encouraging. The space derived from the larger-dimensional input set is significantly better than the greyscale only case. Moreover, the number of misclassifications in Table 6.4 is much lower than that in Table 6.2. This indicates that the deformation information is being selected effectively in the PCA stage.

6.1.3 ML classifier

In Chapter 4, the ML classifier based on the Gaussian distribution performed better than the FLD on face/nonface classification when the images are represented by greyscales (normalised). In this section we explore the ML classifier when images are represented by greyscales (not normalised), motion vectors, deformation residue, and the combination of motion vectors and deformation residue.

Table 6.5 lists the face/nonface classification results on 19 × 19 pixel images. When the ML classifier is used, the values of $|\Sigma_1|$ and $|\Sigma_2|$ are both infinite. Therefore, Equation 4.16, which makes the error rates in two test sets as close as possible, is adopted.

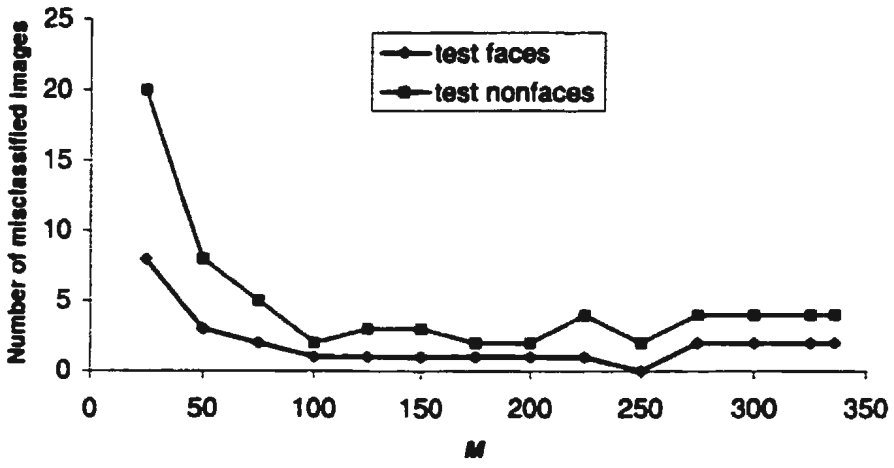
Table 6.5 Number of misclassifications using the ML classifier in a test set containing 2553 nonfaces and 1130 faces with a resolution of 19 × 19 pixels

Representation	Misclassified nonfaces	Misclassified faces
Original greyscales	4	2
Motion vectors	65	29
Deformation residue	8	3
Motion vectors and deformation residue	13	6

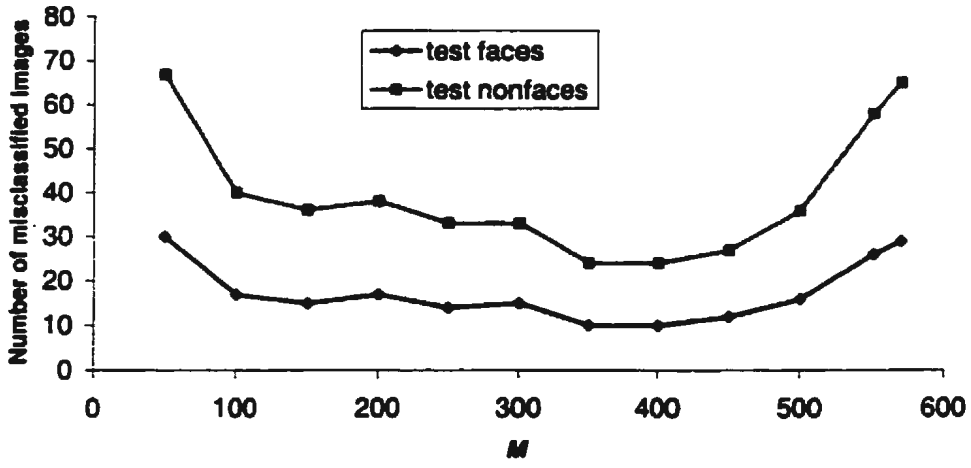
As expected, the ML classifier achieves lower error rates than the FLD on the same data sets. Contrary to what is shown in Table 6.1, the motion vector and/or deformation residue representation does not outperform the greyscale representation.

6.1.4 PCA plus ML classifier

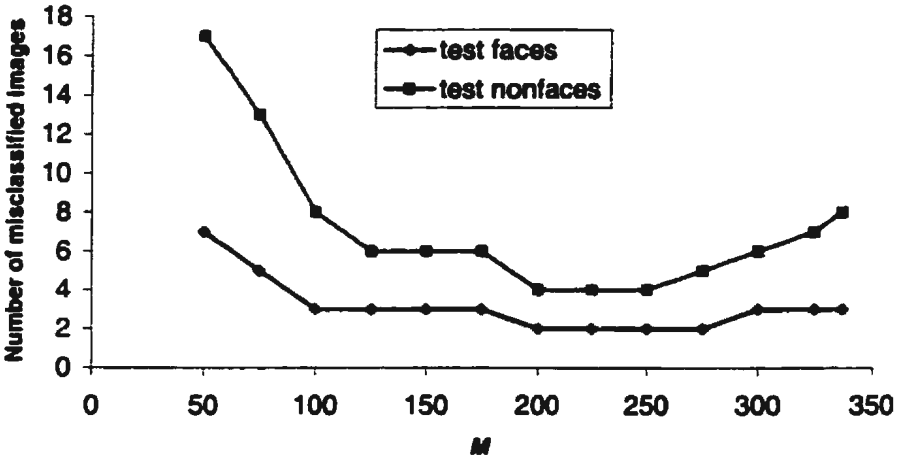
Similar to Section 6.1.2, the M largest eigenvalue eigenvectors of the covariance matrix of training faces are extracted and used to compose a lower-dimensional subspace. All the data sets are projected into this subspace. Then the ML classifier is used to do the face/nonface classification. Figure 6.2 shows the number of misclassified test images using the PCA plus ML classifier on various representations of 19×19 pixel images



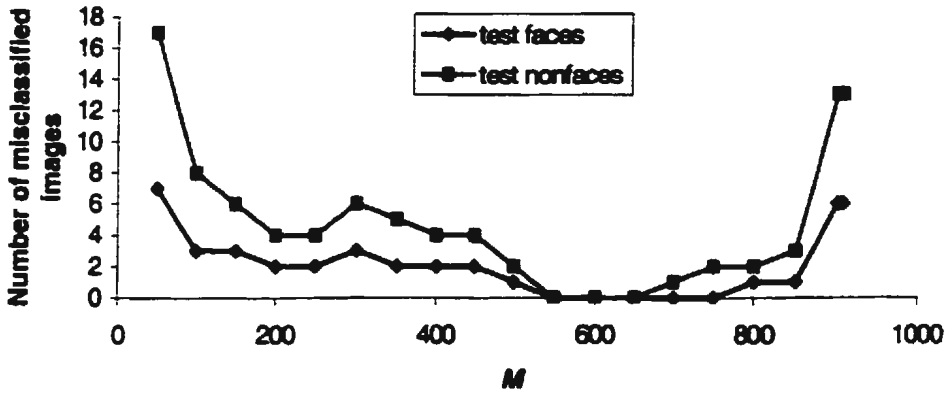
(a)


Number of misclassified images
80
70
60
50
40
30
20
10
0
test faces
test nonfaces
0
100
200
300
400
500
600
M

(b)


Number of misclassified images
18
16
14
12
10
8
6
4
2
0
test faces
test nonfaces
0
50
100
150
200
250
300
350
M

(c)



(d)

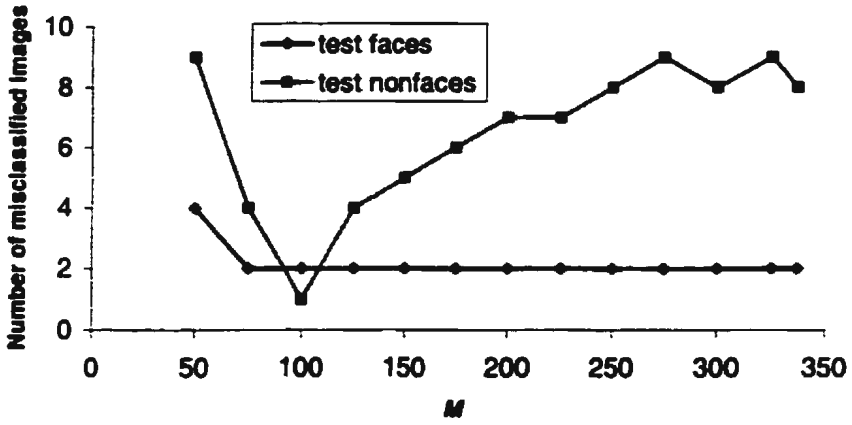
Figure 6.2 Using the ML classifier, the number of misclassified test images versus the number of dimensions of eigenspace when images are represented by the (a) original greyscales, (b) motion vectors, (c) deformation residue, (d) motion vectors and deformation residue.

Equation 4.16 is used to obtain the results. By and large, the number of misclassifications varies greatly in accordance to M. It is noteworthy that in Figure 6.2d when M is between 550 and 650, there are no misclassifications! This is the best result that we have obtained on these data sets. It proves again that the combination of motion vector and deformation residue representation is superior to the greyscale representation.

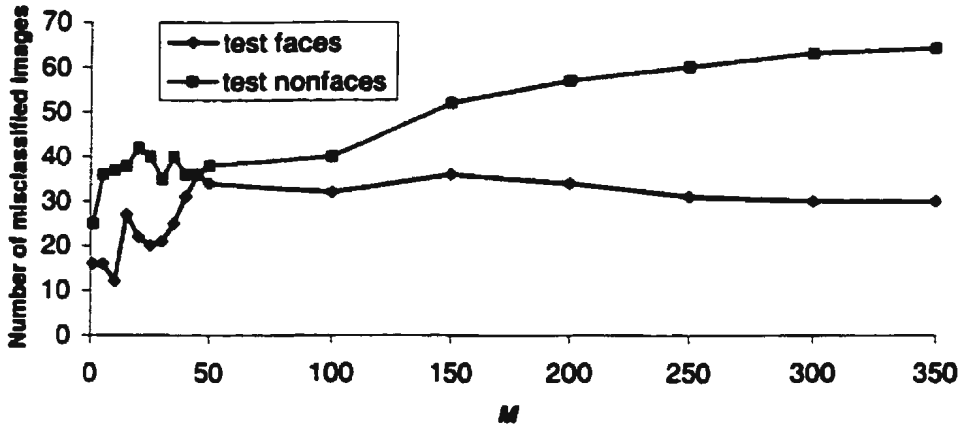
6.1.5 ML classifier based on dominant features

In Section 4.7, the dominant feature extraction technique is applied for the first time to face/nonface classification when images are represented by pixel greyscales. This technique is applied further to face/nonface classification when images are represented by motion vectors and deformation residue.

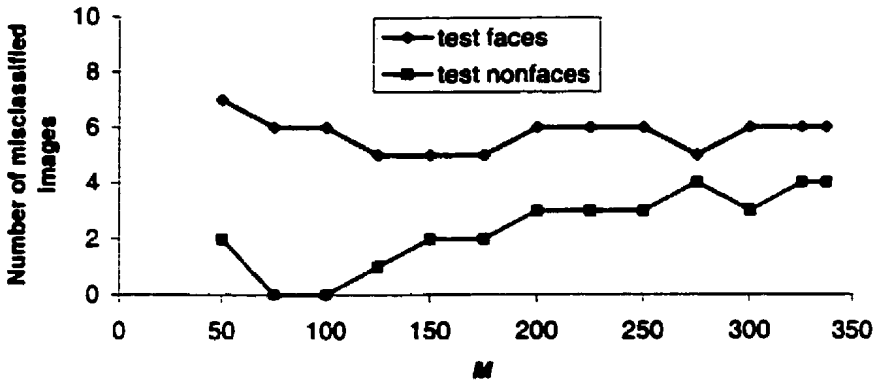
Figure 6.3 presents the results obtained on 19×19 pixel images.


Number of misclassified images
10
8
6
4
2
0
test faces
test nonfaces
0
50
100
150
200
250
300
350
M

(a)


Number of misclassified images
70
60
50
40
30
20
10
0
test faces
test nonfaces
0
50
100
150
200
250
300
350
M

(b)



(c)

Figure 6.3 Using the ML classifier, the number of misclassified test images versus the number of dominant features when images are represented by the (a) original greyscales, (b) motion vectors, (c) deformation residue.

The training and test data sets used to generate Figure 6.3a are the same as those used in Figure 4.34 except that no normalisation on images is performed here. The best result, 2 misclassified faces and 1 misclassified face, is obtained with 100 dominant features in Figure 6.3a. In general, the results in Figure 6.3a are much better than those in Figure 4.34, so we conclude that the image normalization is detrimental to ML classifier based on dominant features.

Note that in Figure 6.3, Equation 4.16, which makes the error rates equal in the test face set and test nonface set, is not applied. If this equation were used, the results of using all available 337 dominant features would be the same as those results shown in Table 6.5.

Figure 6.3b shows that the more the dominant features of motion vectors included, the worse the classification results. The best result is achieved when only the first feature, the Fisher vector, is used.

Figure 6.3b and Figure 6.3c suggest that if the first dimension from motion vector discrimination is used with M dominant features of deformation residue, the classification results will be improved. Figure 6.4 shows the ML classification results on these features.

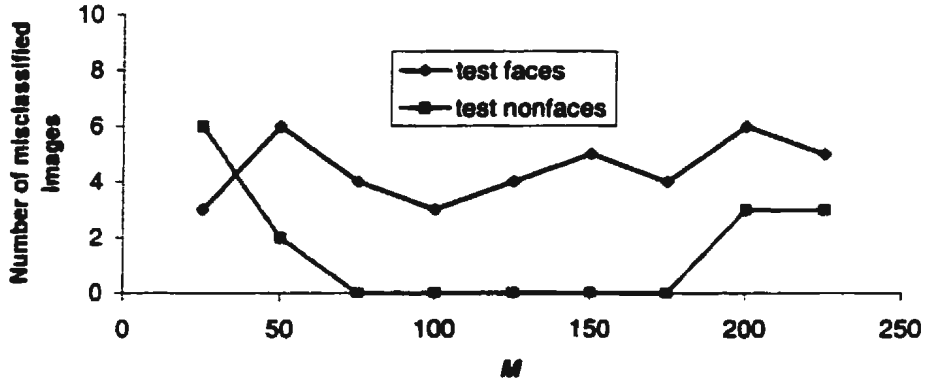


Figure 6.4 Using the ML classifier, the number of misclassified test images versus the number of dominant features of deformation residue. One dimension from motion vector discrimination is used as one of the features.

The results verify our assumption. Previously, only using 100 features of deformation residue, the result is 6 misclassified test faces and no misclassified nonface. With the added dimension from motion vector discrimination, the result is improved to 3 misclassified test faces and no misclassified nonface.

6.1.6 Face detection in still images

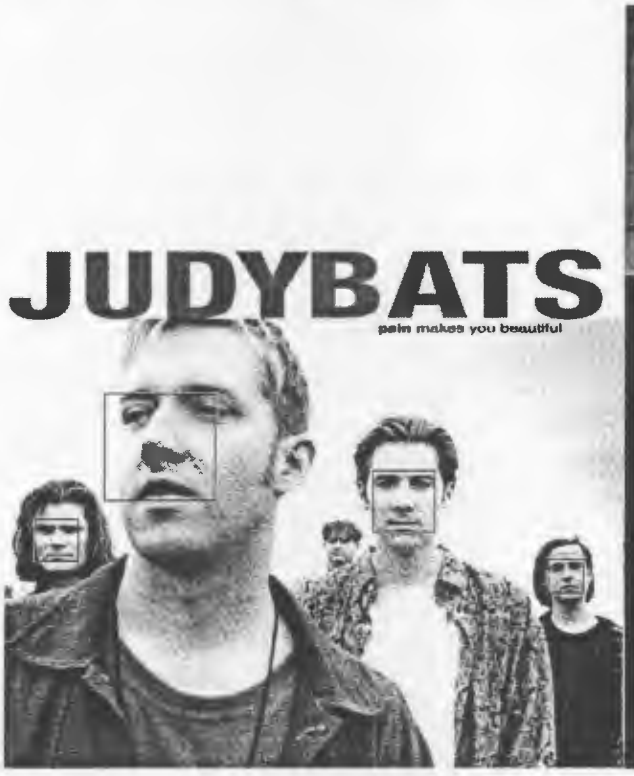
In Figure 6.5 we show some results for face finding using the single linear discriminant in our 500-dimensional PCA space described in Section 6.1.2. The combination of motion vectors and deformation residue is used as the image representation. These experiments use a multi-scale scanning process that finds faces not less than 38×38 pixels in size.

Because most faces are detected at multiple nearby positions or scales, while false detections often occur with less consistency, the method of merging overlapping detections [Rowley 1998] is used. This process is performed by first removing the overlapping detections at the same layer. Then all the detections at the previous layers corresponding to the region of current detection are checked, and if the matching value of the current detection is the smallest, all the detections at the previous layers are removed. Otherwise, only the current detection is eliminated.

(j)

(k)

(l)



(m)

(n)

(o)

(p)

(q)

Figure 6.5 Face detection output

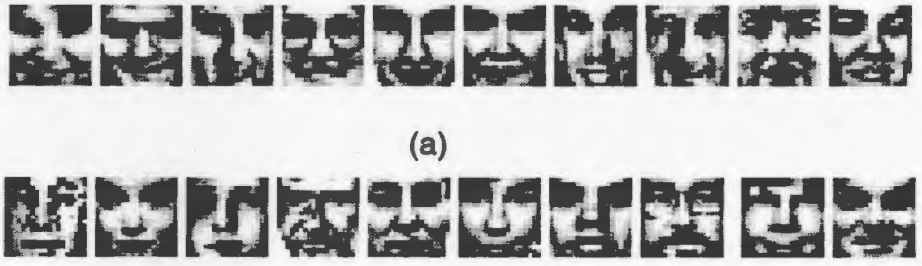
The results include one missed face in Figure 6.5c and two false positives in Figure 6.5j and Figure 6.5t. The reason for missing the child's face in Figure 6.5c is probably the occlusion, the tilt of the face, and the noise of that picture. The false positive in Figure 6.5j looks quite like a real face. Our face detection algorithm performs quite well on other images, for example, although the person in Figure 6.5c has closed eyes and tilted head, his face is successfully detected.

The computation time of this face detection algorithm depends on the size and complexity of the picture. It takes roughly 20 minutes on Windows 98 platform on Figure 6.5o, which is an originally 320 × 240 pixel image. This long processing time limits the practical application of the method with today's technology.

6.2 Classifying Smiling and Nonsmiling Images

We perform experiments on identifying facial expressions where the training images are divided into two classes: 324 smiling faces and 596 nonsmiling faces. Smiling/nonsmiling classification is chosen because, as illustrated by the eigenvector pictures in Figure 5.9, smiling is captured in several dimensions.

Examples of 19 × 19 pixel smiling and nonsmiling face images are shown in Figure 6.6. All the images are in upright position.



(a)

(b)

Figure 6.6 Examples of 19 × 19 pixel (a) smiling faces, and (b) nonsmiling faces

This figure illustrates the degree of subjective judgement required. For example, some people may judge the last face of Figure 6.6b as smiling.

6.2.1 FLD

The smiling/nonsmiling classification results using the FLD are reported in Table 6.6. No normalisation or PCA is performed.

Table 6.6 Number of misclassifications using the FLD in a test set containing 268 nonsmiling faces and 134 smiling faces with a resolution of 19 × 19 pixels.

Representation	Misclassified nonsmiling faces	Misclassified smiling faces
Original greyscales	24	26
Motion vectors	38	47
Motion vectors only ($n = 2$)	24	22
Deformation residue only	26	40
Motion vectors and deformation residue	115	75
Motion vectors and deformation residue ($n = 3$)	17	29

It is clearly shown that the representation in which the motion vectors and deformation residue are utilised together performs the best again. It is also shown that the collection method is superior to the holistic method. A possible explanation is that the holistic method involves larger dimensions than the collection method, but the number of training samples is small. The relatively small number of training samples is not sufficient to estimate the within-class scatter matrix.

On the other hand, the error rates are higher than those of face/nonface classification. This is caused by the intrinsic ambiguity of face images; for example, a face can be regarded as both smiling and non-smiling even by human beings. Clearly our space has not achieved separation of smiling as a feature. But do smiling faces fall within a well-defined linearly discriminable region of the space?

6.2.2 ML classifier

The smiling/nonsmiling classification results using the ML classifier are reported in Table 6.7.

Table 6.7 Number of misclassifications using the FLD in a test set containing 268 nonsmiling faces and 134 smiling faces with a resolution of 19 × 19 pixels

Representation	Misclassified nonsmiling faces	Misclassified smiling faces
Original greyscales	112	82
Motion vectors	137	69
Deformation residue	110	98

About 50% of the images are misclassified. In the two-class classification case, this means that the ML classifier does not provide any discrimination between these two kinds of face images. The underlying reason is that the distribution of the classes is not Gaussian and the ML classifier is therefore over-trained by the training set. Hence, the FLD is more useful than the ML classifier on smiling/nonsmiling faces discrimination.

6.3 Pose Estimation Experiments

As described in Section 3.1.3, in our face image database 10 images are generated from one original face picture. Among the 10 images, 6 images have no rotation; one image has an in-plane rotation degree of −5°, one image −10°, one image 5°, and the other image 10°. "−" sign means rotating left, so no sign means rotating right.

Given an image **x** with a rotation degree in the set {0, –5°, –10°, 5°, 10°}, we find its k nearest neighbours in terms of Euclidean distance in the face image database. The pose of **x** is estimated as the majority of its k nearest neighbours' poses.

The experiment is conducted in four feature spaces:

- **Space 1: original greyscale space**

  We only use the greyscales of original images and do not perform PCA.

- **Space 2: eigenspace of motion vector and deformation residue representation (scaled deformation residue part)**

  For 4650 38 × 38 resolution face images, we get the top 500 eigenvectors of motion vector and deformation residue representation (the deformation residue part is divided by 100), then project all the samples to this eigenspace

- **Space 3: eigenspace of normalised motion vector and deformation residue representation**

  For 4650 38 × 38 resolution images, we get their motion vector and deformation residue representation. The deformation residue part is not scaled by any factor, but the mean and variance of all the elements in one dimension are calculated, and thus the samples in every dimension are made zero mean and unit variance. We take the top 500 eigenvectors to compose the eigenspace.

- **Space 4: the eigenspace of greyscale representation**

  Space 4 also utilizes the original greyscale space, but the top 100 eigenvectors of 4650 images are extracted to compose an eigenspace. All the images are projected into this eigenspace.

We randomly select one search probe, as shown in Figure 6.7, from our face database.

Figure 6.7 A search probe

The 20 closest matches for the search probe are presented in Figure 6.8 where from left to right the matching distance increases.

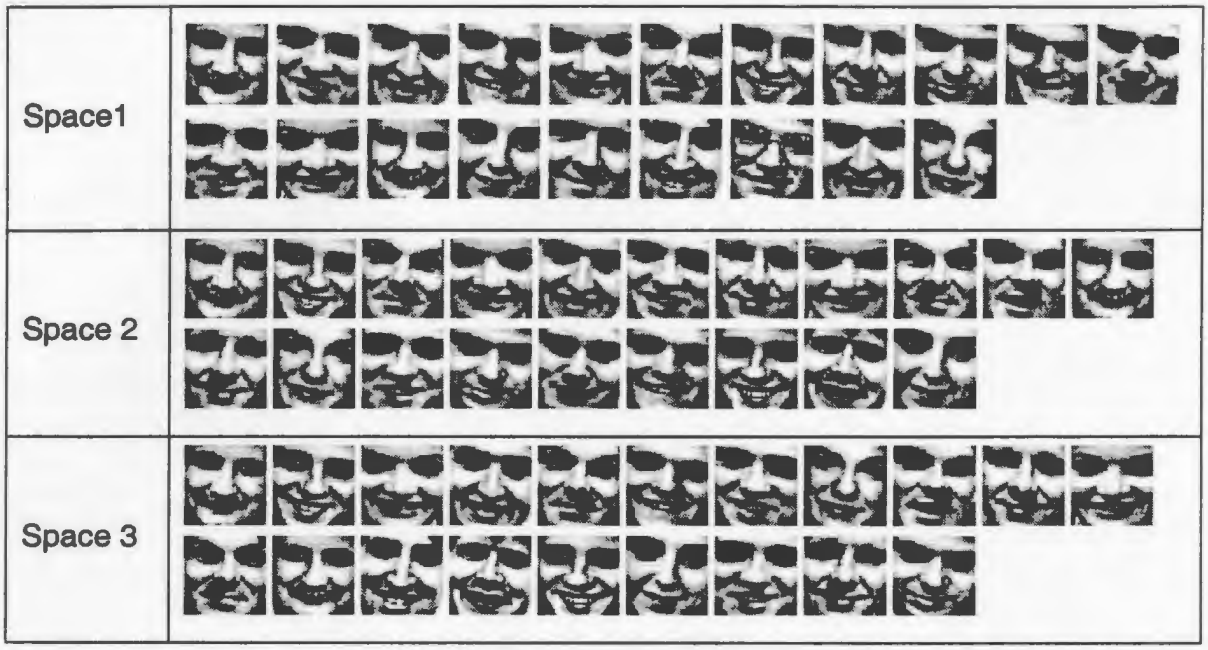


Figure 6.8 20 closest matches for a search probe in three different spaces

The images in Figure 6.8 are almost all rotated right and smiling. It proves that in this feature space, the distance metric correlates to the face pose and expression.

Figure 6.8 shows the closest matches for only one image. Now we measure the closest matches for every image of the 4650 images in the database. Because the 4650 images are composed of 10 images of each of 465 persons, we can find out among the top 10 matches of an input image, how many are the images of the same person as the input image. These images are termed as "correct person matches", while the images of the same pose as the input image are termed as "correct pose matches". Note that although

we use the term "correct person matches", it actually means the matches of the same photograph given that all 10 images of each person are based on a single photograph.

Out of $4650 \times 10 = 46500$ matches, the number of correct matches in each space is listed in Table 6.8.

Table 6.8 Number of correct pose and person matches in four spaces

	Space 1	Space 2	Space 3	Space 4
Correct person matches	8848	10418	10542	9174
Correct pose matches	36648	35299	35018	36504

Space 1 and 4 tend to match pose first, while space 2 and 3 emphasise more the person match than the pose match. This proves that motion vector and deformation residue representation is better than luminance representation in face recognition, but worse than luminance representation in pose estimation. There is not much difference in their overall performance.

Space 4 provides a higher face recognition rate than Space 1, but about the same on pose estimation. Therefore we can draw a conclusion that the eigenspace is slightly superior to the original greyscale image in face recognition.

6.4 Face Recognition Experiments

Face recognition is a broad term that can be specified into two tasks. The first task is to determine if the individual shown in the presented face image has already been seen. The second task is the classification, which is to assign the face image to a certain class corresponding to a known person. Here, the classification task is implemented. All the experiments have been executed on the faces provided by the ORL Face Database.

6.4.1 ORL face database

The ORL database contains a set of faces taken between April 1992 and April 1994 at the Olivetti Research Laboratory in Cambridge, U.K. There are ten different images of 40 different persons. For some of the persons, the images were taken at different times. There are variations in facial expression (open/closed eyes, smiling/nonsmiling), and facial details (glasses/no glasses). All the images were taken against a dark background with the persons in an upright frontal position, with tolerance for some tilting and rotation of up to about 20 degrees. There is some variation in scale of up to about 10%. 200 images out of the 400 images are shown in Figure 6.9. The images are greyscale with a resolution of 92×112 pixels.

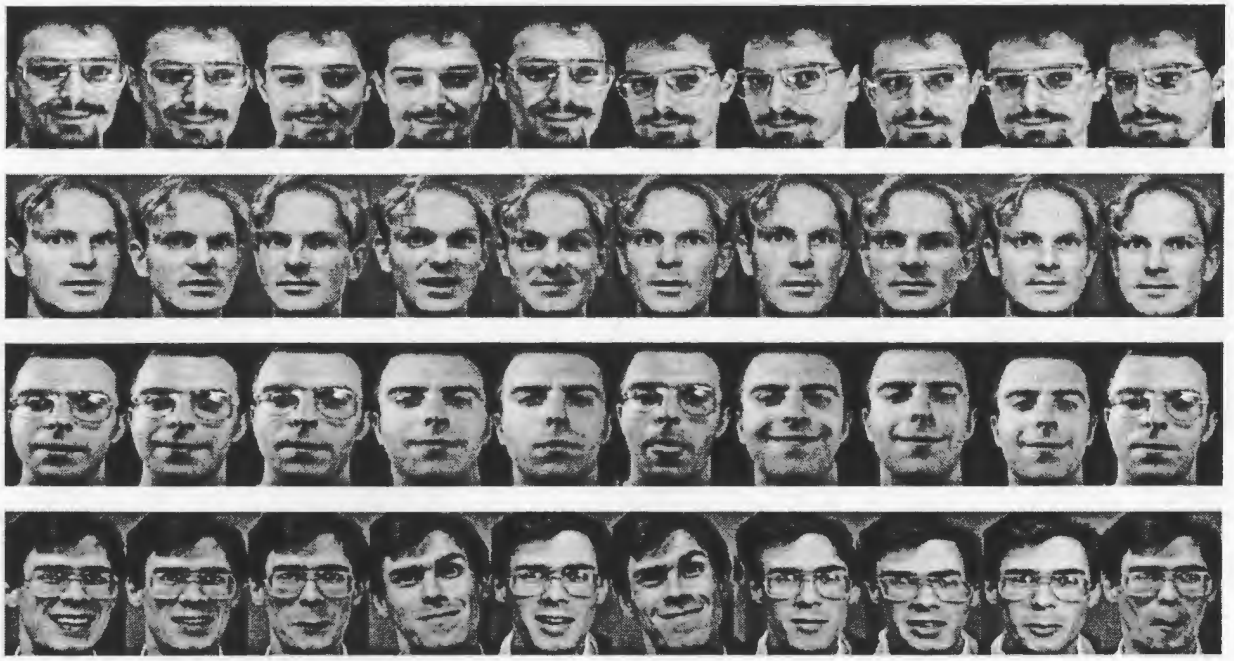


Figure 6.9 The images of 20 persons in the ORL face database

6.4.2 Methods and results

Brunelli and Poggio [Brunelli 1993] mentioned that 36×36 pixels per face is enough for face recognition by human beings. Here the face recognition experiments are performed on 38×38 pixel face images.

The ORL face database contains 40 classes and each class contains 10 samples. For training and testing, we adopted a leave-one-out scheme, which makes maximal use of the available data for training. In this procedure, the classification process was performed multiple times, each time using all the samples but one in a class for training and the remaining one sample for testing. The procedure is repeated for each of the 10 samples.

Every image can be represented by the original greyscale, motion vectors, deformation residue, or their combination. Note that no whitening as described in Chapter 4 is adopted. In each case, the PCA subspace is composed of the top M eigenvectors extracted from 4650 faces, which are used as the training face set in the face/nonface

classification experiments. By using the covariance matrix obtained from a large set rather than covariance matrices computed for individual classes, we ensure that we get a good estimate despite the limited number of training examples available. Then the set of PCA features are generated for each extracted image from the ORL database and stored. When an image query is presented, it is projected to this subspace. The nearest neighbour classifier based on a simple Euclidean distance in this feature space is adopted for classification.

The 38 × 38 pixel face images are extracted manually or automatically from the ORL face database. The face classification results on these images are presented in the next section.

6.4.2.1 Experimental results on manually extracted face images

The method of extracting the face region from a picture was described in Section 3.1.1. After manually marking the two eye centres, nose tip, two mouth corners and mouth centre of every face image, we get the extracted faces of 38 × 38 pixel resolution. Preprocessing including shade removal and histogram equalisation is applied. Examples are shown in Figure 6.10.

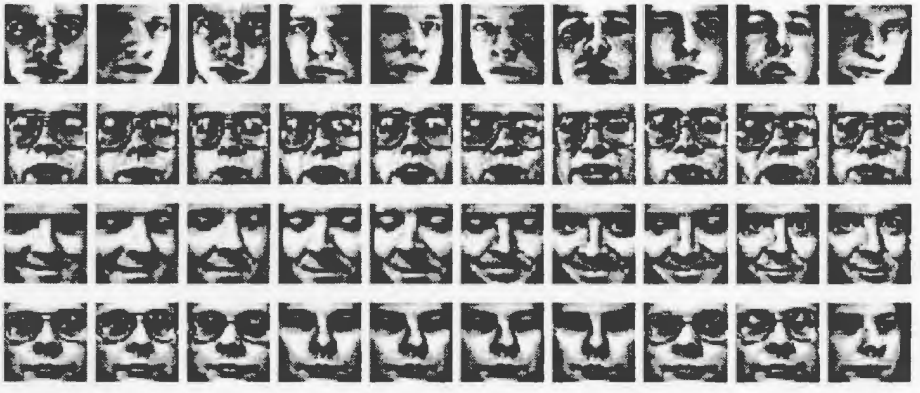


Figure 6.10 Examples of the manually extracted 38 × 38 pixel face images

Table 6.9 lists the nearest-neighbour classification results when face images are represented by different attributes. No PCA is performed.

Table 6.9 Number of correct best matches out of 400 classifications on manually extracted face images. No PCA is performed

Representation	# of correct best matches	Correct rate (%)
Original greyscales	386	96.5
Motion vectors	373	93.3
Deformation residue	388	97.0
Normalised motion vectors and deformation residue	385	96.3

The last representation in Table 6.9, normalised motion vectors and deformation residue, means that the mean and variance of all the elements in one dimension are calculated, and thus the samples in every dimension are made zero mean and unit variance. This representation is the same as the space 3 in pose estimation experiments in Section 6.3.

In the deformation residue representation, the experimental results are that out of 400 classifications the number of correct classifications is 388, which gives us a correct rate of 97%. Correct classification in top 5 matches is 98.5%.

The results are very encouraging. Because the extracted face images only contain the face region between and including the eyebrow and mouth, the classification results are not affected by the hair or beard in face images.

Figure 6.11 shows a search image and its top 10 matches. The first 4 matches are correct, the remaining 6 matches are incorrect but these 6 images show the same pose.

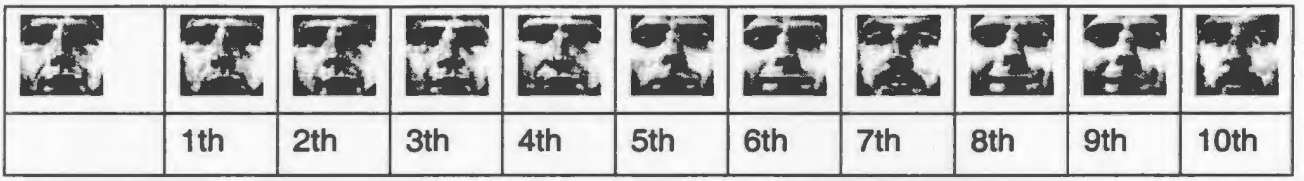


Figure 6.11 A search image and its top 10 matches.

However, 400 – 388 = 12 images are misclassified. 9 of those 12 images and their incorrect best matches are shown in Figure 6.12

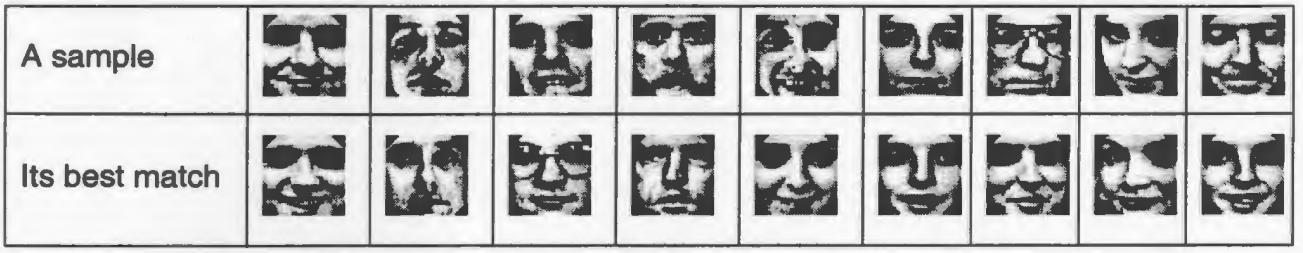


Figure 6.12 Incorrect best matches for 9 images.

These incorrect best matches show the same expression or pose as the search images. This verifies that the individual structure, pose, and expression are intermingled together and hard to separate.

Figure 6.13 shows the correct classification rate if PCA is performed in each representation space.

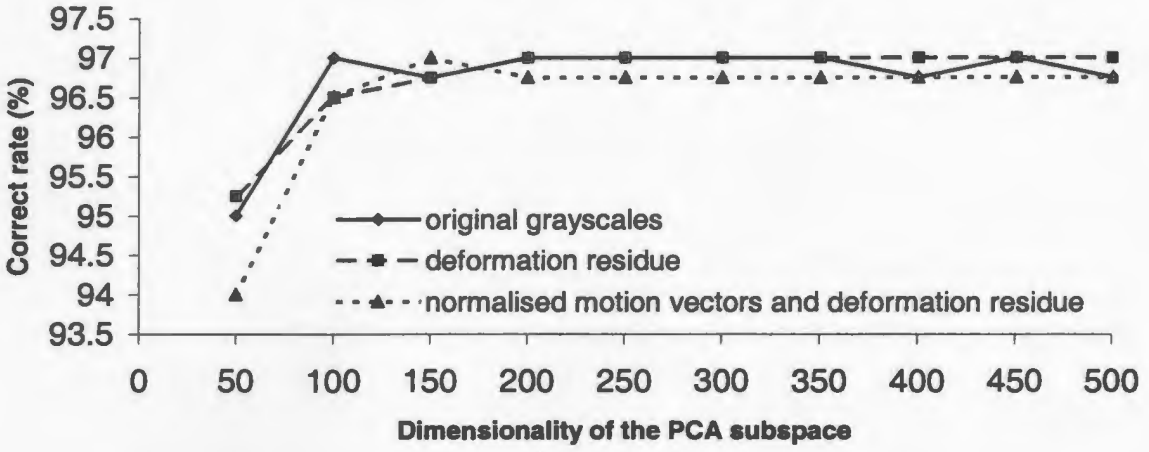
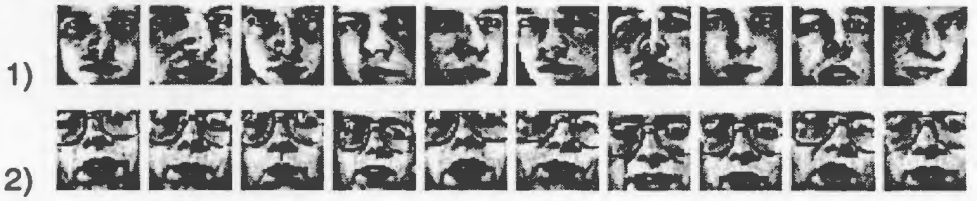


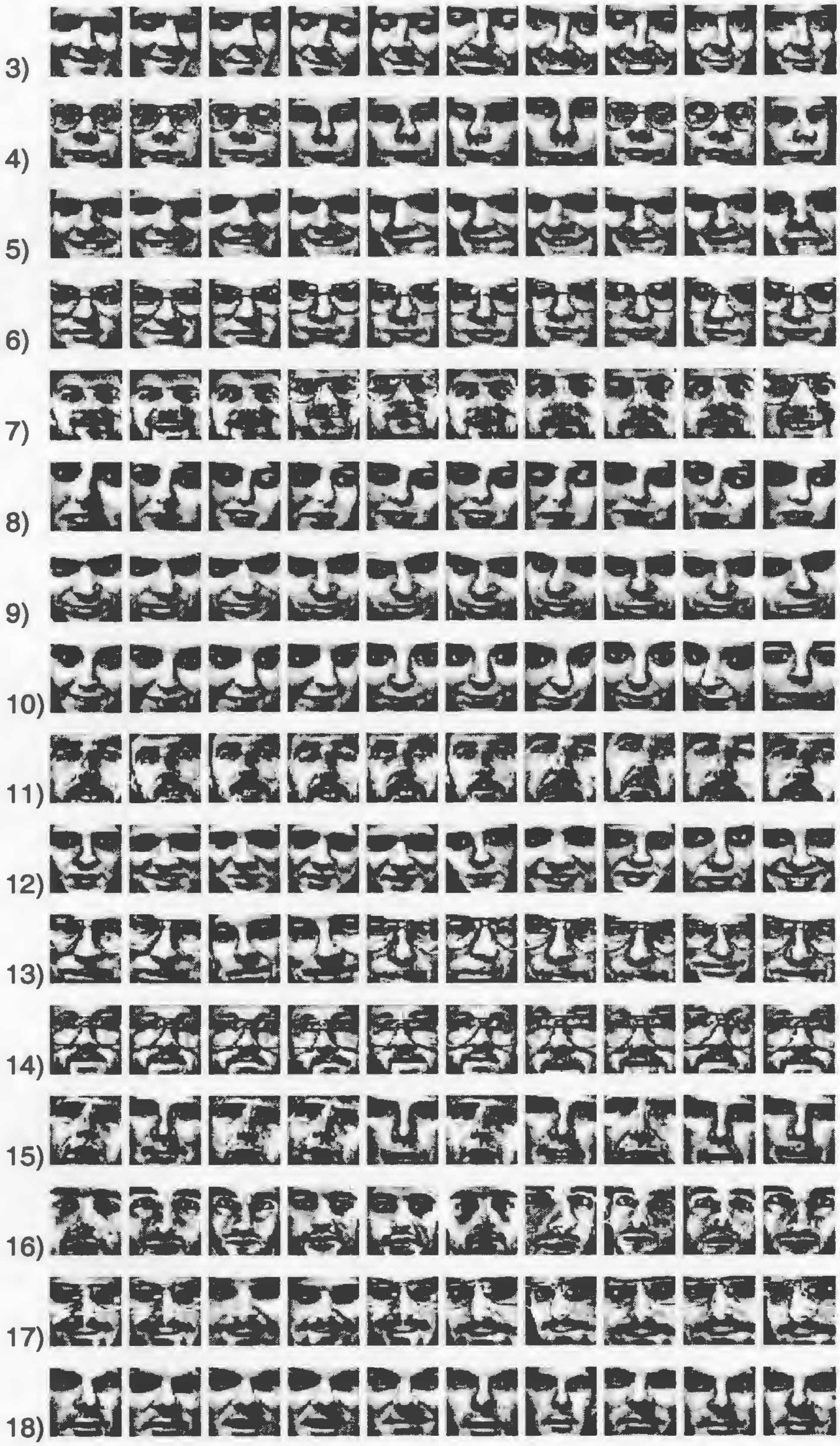
Figure 6.13 Using PCA plus the nearest neighbour classifier, the correct face recognition rate on manually extracted face images versus the dimensionality of the PCA subspace

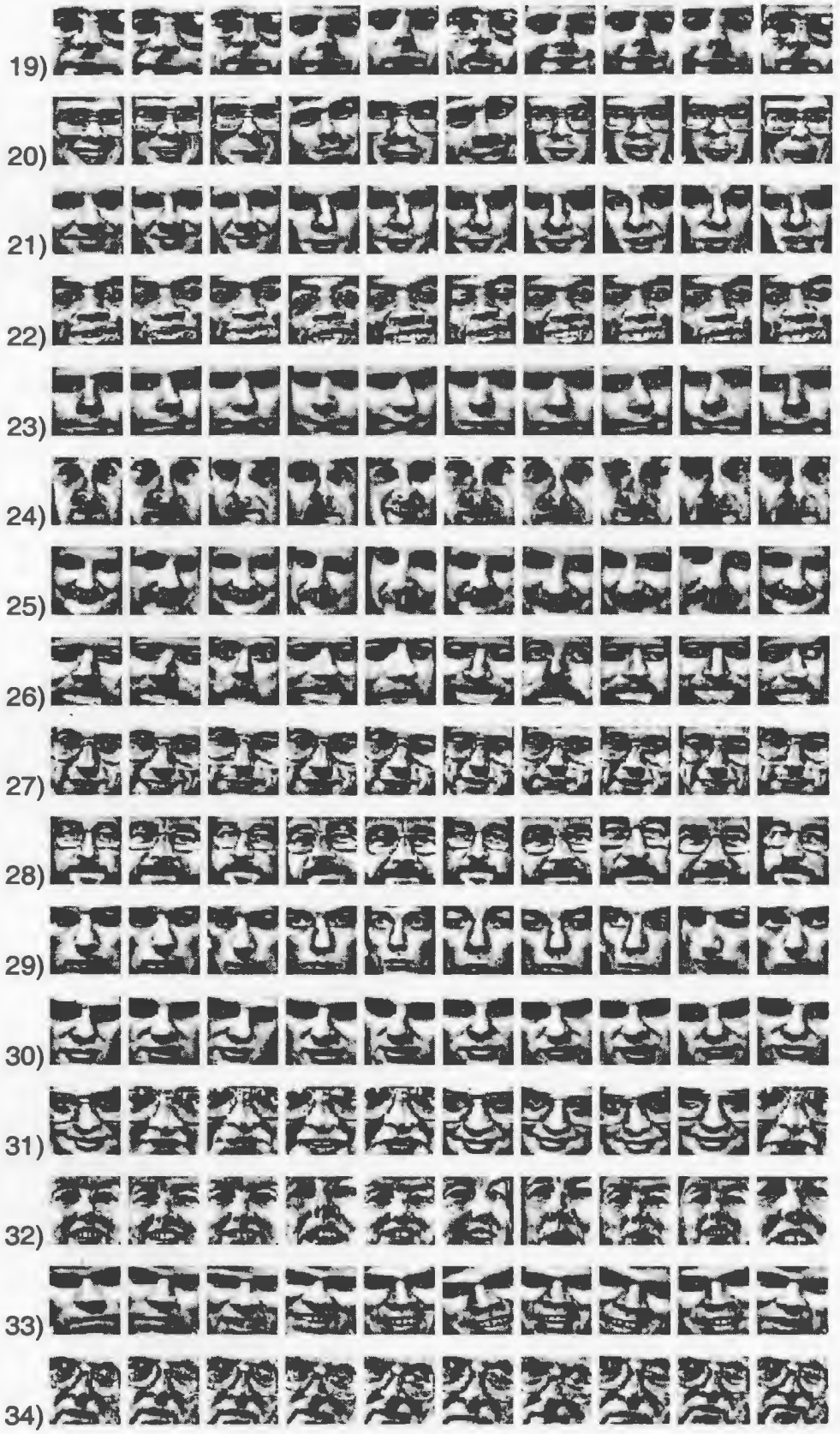
Not much difference is observed in these three curves.

6.4.2.2 Experimental results on automatically extracted face images

Although the face recognition results are pretty good, the faces are extracted according to manually marked facial features. Now we try to extract the face regions automatically. We apply our face detection algorithm described in Section 6.1.6 to the ORL face database and extract the face regions automatically. The extracted images are shown in Figure 6.14. The number to the left of a row means the serial number of a person in the ORL database.




3)
4)
5)
6)
7)
8)
9)
10)
11)
12)
13)
14)
15)
16)
17)
18)


19)
20)
21)
22)
23)
24)
25)
26)
27)
28)
29)
30)
31)
32)
33)
34)

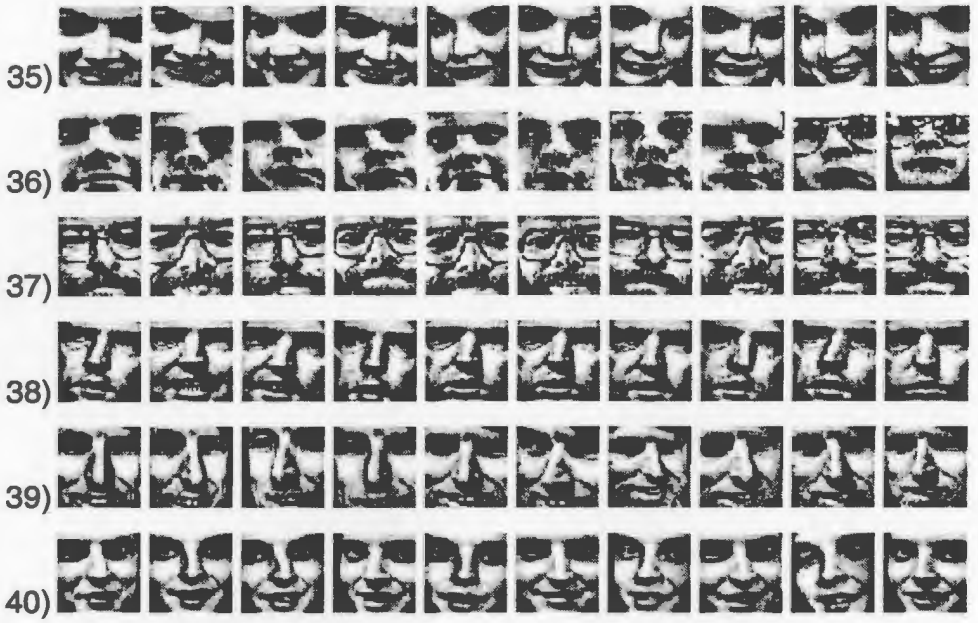


Figure 6.14 Automatically extracted face regions from the ORL face database

Although all the face regions have been successfully extracted, compared with the manually extracted face samples, the scale and in-plane rotation in the automatically extracted face regions are not constant. The located face regions in the row 2, 31, and 34 sometimes contain less upper face because of the bright eyeglasses. The task of detecting and extracting the face region of the same size and orientation is difficult to tackle.

From the 400 extracted images belonging to 40 classes, we do face classification using the nearest-neighbour classifier. The images are represented by the original greyscale, the deformation residue, or the motion vectors.

When the PCA is not performed, the numbers of correct classifications are tabulated in Table 6.10.

Table 6.10 Number of correct best matches out of 400 classifications on automatically extracted face images. No PCA is performed.

Representation	# of correct best matches	Correct rate (%)
Original greyscales	331	82.8
Motion vectors	300	75.0
Deformation residue	334	83.5
Normalised motion vectors and deformation residue	322	80.5

The deformation residue representation outperforms all other three representations. The reason might be that the deformed images have reduced high frequency components and are smoother than the original images. The representations including motion vectors do not work well.

When the PCA in each representation space is performed, the number of correct matches versus the dimensionality of PCA subspace is shown in Figure 6.15.

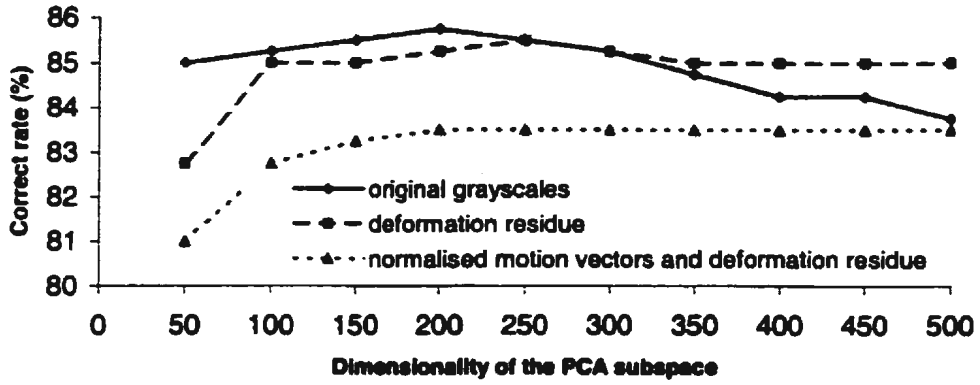


Figure 6.15 Using PCA plus the nearest neighbour classifier, the correct face recognition rate on automatically extracted face images versus the dimensionality of the PCA subspace

The original greyscale representation gives the best result, 343 correct best matches or 85.8% correct rate, when the dimensionality is 200. The deformation residue representation achieves 342 correct matches when the dimensionality is 300. The performances of these two representations are quite similar when the dimensionality is low. However, at high dimensionality, the deformation residue representation is better.

Comparing Figure 6.15 and Table 6.10, we see that PCA improves classification results by around 2%.

Overall, the results based on the automatically extracted images are less satisfying than those based on the manually extracted images. The reason is that the automatically extracted face images are not aligned very well. As shown in Figure 6.14, the scale and orientation of the extracted face region are not consistent among all the images. If this can be improved, we expect better results that are upper-bounded by the results on manually extracted images.

In [Lawrence 1997] an eye detector was used to locate the eyes and then to normalise the face orientation and size. However, we notice that the motion vectors including global motion can also compensate for tilt. Thus we find an alternative way.

For the images shown in the 20th row of Figure 6.14, the deformed images due to the motion vectors including global motion are shown in Figure 6.16.

Figure 6.16 Deformed images due to the motion vectors including global motion

The tilted images have been rotated back to upright position. Thus the images for the same person look more similar.

Based on the automatically extracted face images, the number of correct best matches is listed in Table 6.11.

Table 6.11 Number of correct best matches out of 400 classifications on automatically extracted face images. No PCA is performed. The motion vectors include the global motion. The deformation residue is generated by these motion vectors

Representation	# of correct best matches	Correct rate (%)
Deformation residue	357	89.3
Motion vectors	83	20.8

The result using deformation residue representation is much better than that in Table 6.10. This proves that the tilt problem can be overcome by including global motion estimation in motion vectors.

It is not surprising that the motion vector representation worsens the face recognition performance as compared with that listed in Table 6.10. Now the motion vectors include global motion which is usually more significant than the local motion caused by facial structure difference. Therefore the nearest neighbour classifier will match an input image with images with the same orientation as the input image first.

6.5 Summary

The effectiveness of the proposed motion vectors and deformation residue representation is demonstrated through experiments on face detection, expression analysis, pose estimation, and face recognition. Four image representations for comparison are original greyscales, motion vectors, deformation residue, and the combination of motion vectors and deformation residue.

- Face detection

  Using FLD, the combination of motion vector and deformation residue representation performed the best in face/nonface classification. FLD in PCA derived eigenspace outperformed the pure FLD.

  Using ML classifier, the original greyscale representation is superior to the others. However, using ML classifier in PCA subspace, the combination of motion vectors and deformation residue representation performed the best again.

  Using ML classifier and the dominant feature extraction technique, the results are as good as those using ML classifier in PCA subspace. However, fewer features are required.

  The face detection results using FLD in PCA subspace on real-world images are presented.

- Smiling/nonsmiling image classification

  Using FLD, the combination of motion vectors and deformation residue representation generated the best results.

  Using ML classifier, no meaningful classification results were obtained because the distribution of classes was not Gaussian.

- Pose estimation

  Pose estimation experiments are conducted using the nearest neighbour classifier on 4650 images of 5 different poses. The motion vectors and deformation residue representation in the 500-dimensional PCA subspace tends to match the images of same person first. However, the greyscale representation tends to match the images of same pose first.

- Face recognition

  The ORL face database containing 10 images for each of 40 persons was used for face recognition experiments.

  When the face regions were extracted manually, the deformation residue representation achieved 97% correct classification rate, higher than that of the original greyscale representation. In PCA subspace, these two representations performed nearly equally.

  When the face regions were extracted automatically using the face detection algorithm, the deformation residue was still the best with 83.5% correct rate. The drop in correct rate was caused by the relatively poor alignment of the extracted face regions. In the 250-dimensional PCA subspace, 85.3% correct rate was obtained with the deformation residue representation. The alignment problem is solved by including global motion in the motion vectors. 89.3% correct rate was achieved with the deformation residue representation which utilised the global motion.

Chapter 7

Conclusions

7.1 Contributions of this Research

All of the experiments in this research have been carried out on still greyscale images. The main contributions of this research include the following.

1) **Systematic comparison of feature spaces and discriminants using pixel measurements only**

Face images, nonface images, and anything-images of size 19×19 pixels are utilised. Face images are of frontal or near-frontal views of faces with considerable variation in pose, expression, lighting condition, etc. Nonface images were collected from natural scenery images at different scales by template matching. Anything-images were obtained by randomly extracting a part of scenery images without template matching.

The comparison of classifiers was performed in four spaces: original greyscale space, face-image-whitened space, anything-image-whitened space, and double-whitened space. A whitened space is a PCA subspace whose variance along each dimension is normalised to unity. The Euclidean distance in the whitened space is equal to the Mahalanobis distance in the original space. The PCA subspace utilising the principal components from face images is called face-image-whitened space. Likewise, the principal components of the anything-image-whitened space come

from anything-images. The double-whitened space is obtained by performing face-image whitening in the anything-image-whitened space.

The performance of linear discriminant classifier, hyperquadratic discriminant classifiers, and nearest neighbour classifiers is compared through face/nonface classification experiments in the aforementioned four spaces. The group of hyperquadratic discriminant classifiers includes the ML classifier based on the Gaussian distribution or the hyperellipsoid distribution, and the ML classifier based on the principal subspace and the complementary subspace. The group of nearest neighbour classifiers include the nearest neighbour classifier, the k- nearest neighbours, and the k / l nearest neighbour classifier. Experimental results show that the hyperquadratic discriminant classifiers performed the best, followed by the nearest neighbour classifiers, then the linear discriminant classifier. Another observation is that the lower-dimensional whitened space provides better discriminating power than the original greyscale space no matter which kind of classifier is used. The best results of 0.09% misclassified test faces and 0.04% misclassified test nonfaces are achieved by using the ML classifier in a 250-dimensional face-image-whitened space.

2) **Application of new feature extraction technique and the invention of classification schemes**

A new feature extraction technique, named *dominant feature extraction*, is applied for the first time to face/nonface classification with encouraging results. The Fisher vector that corresponds to the class separability caused by the mean-difference between two classes is extracted as the first feature. In a subspace orthogonal to the first feature, other features maximising the variance-difference between two classes are extracted. The ML classifier based on the features obtained by using this

technique gave 0.18% misclassified test faces and 0.39% misclassified test nonfaces in a 90-dimensional feature space. The low number of features required is the main benefit of this technique.

The proposed *Repeated FLD* classifier obtains a group of Fisher vectors between two classes through iteratively reducing the training samples, adding new training samples, or rotating coordinate system and removing dimension. This scheme is better than single FLD on face/nonface classification.

The proposed *Moving-centre scheme* takes advantage of the fact that in the face-image-whitened space or double-whitened space, the face images compose a hypersphere. In face/nonface classification based on Euclidean distance, the centre of the face class is modified to find a position where the misclassification rate is the lowest. This scheme achieved better results than the FLD.

In addition, the hyperellipsoid distribution instead of Gaussian distribution was used to model the face class and nonface class distribution. The ML classifier based on it generated good classification results.

3) **Use of motion vectors and deformation residue**

Optical flow is commonly used for face expression analysis in an image sequence. However, we use the motion vectors obtained through optical flow analysis as a kind of representation of a single image.

The process of pulling the pixels of an input image along the motion vectors from the input image to a neutral face template is called *deformation*. At present the mean of a large set of face samples is used as the face template.

For an input image, we thus derive two new representations: the pixel values of the deformation residue, which is the difference between the face template and the deformed input image, and the motion vectors from the deformed input image to the

original input image. The greyscales of an image are pixel measurements, while the motion vectors are non-pixel measurements. The aim of introducing deformation residue is to achieve invariance to local distortions.

A representation space including these two representations combines the shape information with the intensity information. It is shown that the principal components capture the outstanding variations across the training face set.

By comparing the mean images between face image pairs in the conventional greyscale space with those in the new representation space, we demonstrate that the face cluster is more convex in the proposed space.

In order to save computation time, the optical flow calculation is performed on small images. The currently adopted image size is 19×19 or 38×38 pixels.

4) Applications of the proposed representation space on face image processing

We have made a demonstration that a feature space derived using PCA from a large measurement space including motion vectors and deformation-residue pixel values allows better separation of faces and nonfaces and of different expressions on faces than the pixel greyscale measurement space. We then demonstrated our technique in the context of face detection on complex images. It is observed that a larger image (38×38 pixels) is better than smaller image (19×19 pixels) in face/nonface classification if everything else is equal.

We continued this investigation by examining the recovery of pose and identity information in the derived face feature space. In the application of pose estimation, the derived feature space tends to match the images of the same person first. However, the original greyscale representation tends to match the images of same pose first. The face recognition experiments show that the deformation residue

representation slightly outperforms the original greyscale representation no matter whether the face regions are extracted automatically or manually. Performing PCA in the representation space improves the correct classification rate by around 2%. 85.3% correct rate was obtained in the 250-dimensional PCA subspace based on the deformation residue representation on the automatically extracted face regions from the ORL face database. The motion vectors including global motion make the deformed images upright. 89.3% correct rate was achieved with the deformation residue representation which utilised the global motion.

These promising results demonstrate the potential for use in real life applications. The distinctive feature of the proposed representation space is that it can cope successfully with almost all aspects of face image processing.

7.2 Future Research

According to the results and experience obtained in this work, future research can be conducted.

1) Satisfying results have been obtained in face detection experiments using the FLD classifier only. In order to further improve the results, the FLD can be applied first, then a hyperquadratic discriminant classifier applied second. The reason is that the FLD is fast and has great discriminating power. A large number of nonface patterns can be eliminated after this step. On the other hand, the hyperquadratic discriminant classifier is slow but accurate. The remaining undecided test patterns will be classified with high correct rate. Therefore, both the speed and the accuracy requirement can be fulfilled. This idea coincides with that of Weber and Hernández [Weber 1999].

2) From the experimental results on face recognition we can see that if the face images are normalised with the same scale and horizontal orientation, the correct classification rate is 97%; however, with relatively poorly-normalised face images, the classification rate dropped to 85%. We used the motion vectors including global motion to solve the orientation problem. Another possible solution is to include a normalisation step after the face detection step. The normalisation should consist of scaling, rotating, and cutting the image such that the eye centres lie at a predefined position. This in turn requires a facial feature location step.
3) The proposed dominant-feature-extraction technique generates encouraging results on face/nonface classification. It can be extended to other areas, such as face recognition.
4) The image representation based on the motion vectors and deformation residue has proved to be effective in various tasks of face image processing. Although we have compared the performance of various images used as the face template and find out that the mean face is the best choice, there might be other solutions better than the mean face as the face template. The search for the better solutions will be another direction of future research.

Bibliography

Achermann, B., and Bunke, H. (1996). "Combination of Face Classifiers for Person Identification," *Proc. 13th International Conference on Pattern Recognition*, Vienna, Austria, Vol. III, pp. 416-420.

Bässmann, H., and Besslich, P. W. (1995). *Ad Oculos, Digital Image Processing, Student Version 2.0*, International Thomson Publishing.

Bartlett, M.S. (1998). "Face Image Analysis by Unsupervised Learning and Redundancy Reduction," PhD thesis, Univ. of California, San Diego

Bartlett, M.S., Lades, H.M., and Sejnowski, T.J. (1998). "Independent Component Representations for Face Recognition," *Proc. SPIE Symp. Electronic Images: Science and Technology; Human Vision and Electronic Imaging III*, T.Rogowitz and B.Pappas, eds., vol. 3,299, pp. 528-539, San Jose, Calif.

Belhumeur, P.N., Hespanha, J.P., and Kriegman, D.J. (1997). "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711-720.

Bern, Switzerland, *University of Bern Face Database*, ftp://iamftp.unibe.ch/pub/Images/FaceImages/

Bernd Gärtner, Smallest Enclosing Ball Algorithm, http://www.inf.ethz.ch/personal/gaertner/miniball.html

Bichsel, M. (1991). "Strategies of Robust Object Recognition for the Automatic Identification of Human Faces," Ph.D. dissertation, ETHZ, Zurich, Switzerland.

Bichsel, M., and Pentland, A. (1993). "Automatic Interpretation of Human Head Movements," MIT Media Laboratory, Vision and Modelling Group, Technical Report no. 186.

Bruce, V. (1991). *Face Recognition*, The European Journal of Cognitive Psychology, Lawrence Erlbaum Associates Ltd.

Brunelli, R., and Poggio, T. (1993)"Face Recognition: Features versus Templates," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp.1042-1052.

Brunelli, R. (1997). "Estimation of Pose and Illuminant Direction for Face Processing," *Image and Vision Computing*, vol. 15, pp. 741-748.

Carnegie Mellon University's face detection test sets, http://www.ius.cs.cmu.edu/IUS/har1/har/usr0/har/faces/test

Cheney, W., and Kincaid, D. (1994). *Numerical Mathematics and Computing*, Brooks/Cole Publishing Company.

Cheng, L., and Robinson, J. (1998). "MCLGallery: A Framework for Multimedia Communications Research," *Proc. Newfoundland Electrical and Computer Engineering Conference.*

Cohn, J.F., Zlochower, A.J., Lien, J.J., Wu, Y.T., and Kanade, T. (1999). "Automated Face Coding: A Computer-Vision Based method of Facial Expression Analysis," *Psychophysiology*, vol. 35, no.1, pp. 35-43.

Colmenarez, A.J., and Huang, T.S. (1997). "Face Detection with Information-Based Maximum Discrimination," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 782-787.

Craw, I., and Cameron, P. (1992). "Face Recognition by Computer," *Proc. British Machine Vision Conference 1992*, pp. 489-507.

Craw, I., Costen, N., Kato, T., and Akamatsu, S. (1999). "How Should We Represent Faces for Automatic Recognition?" *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 725 - 736.

Daugman, J.G. (1980). "Two-dimensional Spectral Analysis of Cortical Receptive Field Profiles," *Vision Research*, vol. 20, pp. 847-856.

Daugman, J.D. (1988). "Complete Discrete 2D Gabor Transforms by Neural Networks for Image Analysis and Compression," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 36, pp.1,169-1,179.

Desilva, L.C., Aizawa, K., and Hatori, M. (1995). "Detection and tracking of Facial Features by Using Edge Pixel counting and Deformable Circular Template Matching," *IEICE Trans. Inf. & Syst.*, vol. E78-D, no. 9, pp. 1195-1207.

de Vel, O., and Aeberhard, S. (1999). "Line-Based Face Recognition under Varying Pose," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 1,081-1,088.

Donato, G., Bartlett, M.S., Hager, J.C., Ekman, P., and Sejnowski, T.J. (1999). "Classifying Facial Actions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 974-981.

Ekman, P., and Friesen, W.V. (1978). *Facial Action Coding System*, Palo Alto, Calif.: Consulting Psychologists Press, Inc.

Essa, I.A., and Pentland, A.P. (1997). "Coding, Analysis, Interpretation, and Recognition of Facial Expressions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 757-763.

Forchheimer, R., and Kronander, T. (1989). "Image Coding - From Waveforms to Animation," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 27, no. 12, pp. 2008-2021.

Frey, B., Colmenarez, A., and Huang, T. (1998). "Mixture of Local Linear Subspaces for Face Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 32-37.

Fukunaga, K. (1991). *Introduction to Statistical Pattern Recognition*, Academic Press, second edition, pp457-458.

Gee, A., and Cipolla, R. (1994a). "Estimating Gaze from a Single View of a Face," *Proc. 12th Int'l Conf. Pattern Recognition*, vol. 1, pp. 758-760, Los Alamitos, Calif.

Gee, A., and Cipolla, R. (1994b). "Determining the Gaze of Faces in Images," *Image and Vision computing*, vol. 132, no. 10, pp. 639-647.

Gee, A., and Cipolla, R. (1996). "Fast Visual Tracking by Temporal Consensus," *Image and Vision Computing*, vol.14, pp. 105-114.

Horn, B.K.P., and Schunck, B.G. (1981). "Determining Optical Flow," *Artificial Intelligence*, vol. 17, pp. 185-203.

Huang, J.S., and Yang, C.K. (1994). "Discriminant Analysis Based on Hyperellipsoid Distribution," *Journal of Information Science and Engineering*, pp. 71-79(October).

Jain, A.K., and Dubes, R.C. (1988). *Algorithms for Clustering Data*, Englewood Cliffs, N.J.: Prentice Hall.

Jain, A.K., Duin, R.P.W., and Mao, J. (2000). "Statistical Pattern Recognition: A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4-37.

Jones, M., and Poggio, T. (1998). "Hierarchical Morphable Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 820-826.

Kirby, M., and Sirovich, L. (1990). "Application of the Karhunen-Loeve Procedure for the Characterisation of Human Faces, " *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1.

Kohonen, T. (1995). *Self-Organising Maps*. Springer Series in Information Sciences, vol. 30, Berlin.

Krüger, N., Pötzsch, M., and Malsburg, C. (1997). "Determination of Face Position and Pose with a Learned Representation Based on Labelled graphs," *Image and Vision Computing*, vol. 15, pp. 665-673.

Kruizinga, P., and Petkov, N. (1994). "Optical Flow Applied to Person Identification,", *Proceedings of the 1994 EuROSIM conference on Massively Parallel Processing Applications and Development*, Delft, The Netherlands, pp. 871-878.

Lanitis, A., Taylor, C.J., and Cootes, T.F. (1997). "Automatic Interpretation and Coding of Face Images Using Flexible Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 743-756.

Lawrence, S., Giles, C., Tsoi, A., and Back, A. (1997). "Face Recognition: A Convolutional Neural Network Approach," *IEEE Trans. Neural Networks*, vol. 8, pp. 98-113.

Lin, S.H., Kung, S. Y., and Lin, L.J. (1997). "Face Recognition/Detection by Probabilistic Decision-Based Neural network," *IEEE Trans. Neural networks, Special Issue on Artificial Neural Networks and Pattern Recognition*, vol. 8, no. 1, pp.114-131.

Liu, C. and Wechsler, H. (1998). "Probabilistic Reasoning Models for Face Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Santa Barbara, California, USA, pp. 827-832.

Liu, C., and Wechsler, H. (1999). "Face Recognition Using Shape and Texture," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Fort Collins, Colorado, USA.

Loève, M. (1955). *Probability Theory*. Princeton, N.J.: Van Nostrand.

Mase, K. (1991). " Recognition of Facial Expression from Optical Flow," *IEICE Trans. E*, vol. 74, no. 10, pp. 3,474 - 3,483.

Miros: True face of security, http://www.miros.com

Moghaddam, B., and Pentland, A.(1994). "Probabilistic Visual Learning for Object Detection," *Proc. Int'l Conf. Computer Vision*, pp. 786-793, Cambridge, Mass.

Moghaddam, B., Wahid, W., and Pentland, A. (1998). "Beyond Eigenfaces: Probabilistic Matching for Face Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 30-35, Los Alamitos, California.

Nastar, C., Moghaddam, B., and Pentland, A. (1996). "Generalised Image Matching: Statistical Learning of Physically-based Deformations," *In ECCV*, pp. 589-598, Cambridge, UK

Nelson, L. (1998). "Commercialising Robust Face Recognition Capability, Polariod & Quebec Vision Start-Up," *Advanced Imaging*, pp. 72-73.

Olivetti & Oracle Research Laboratory, *The Olivetti & Oracle Research Laboratory Face Database of Faces*, http://www.cam-orl.co.uk/facedatabase.html

Padgett, C., and Cottrell, G. (1998). "A Simple Neural Network Models Categorical Perception of Facial expressions," *Proceedings of the Twentieth Annual Cognitive Science Conference*, Madison, WI, Mahwah: Lawrence Erlbaum.

Penev, P.S., and Atick, J.J. (1996). "Local Feature Analysis: A General Statistical Theory for Object Representation," *Network: Computation in Neural Systems*, vol. 7, no.3, pp. 477-500.

Pentland, A., Moghaddam, B., and Starner, T. (1994). "View-Based and Modular Eigenspaces for Face Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 84-91.

Robinson, L. (1998). "A 'Virtual Human Agent' User Interface from Japan: Vision Meets Graphics," *Advanced Imaging*, pp. 12-15(May).

Rosenblum, M., Yacoob, Y., and Davis, L. (1996). "Human Expression Recognition from Motion Using a Radial Basis Function Network Architecture," *IEEE Trans. Neural Networks*, vol. 7, no. 5, pp. 1,121-1,138.

Rowley, H.A., Baluja, S., and Kanade, T. (1997). "Rotation Invariant Neural Network-Based Face Detection," Technical Report CMU-CS-97-201, Carnegie Mellon Univ.

Rowley, H.A., Baluja, S., and Kanade, T. (1998). "Neural Network-Based Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23-38.

Samaria, F., and Young, S. (1994). "HMM-based Architecture for Face Identification," *Image and Vision Computing*, vol. 12, no. 8.

Schneiderman, H., and Kanade, T. (1998). "Probabilistic Modelling of Local Appearance and Spatial Relationships for Object Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 45-51.

Schunck, B.G. (1989). "Image Flow Segmentation and Estimation by Constraint Line Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1010-1027.

Sirovich, L., and Kirby, M. (1987). "Low-dimensional Procedure for the Characterisation of Human Faces," *Journal of the Optical Society of America A*, vol. 4, no. 3, pp. 519-524.

Strother-Vien, L, (1998). "Mugshot Recognition Meets Witness Composite Sketches in L.A.," *Advanced Imaging*, pp.20 (January), http://www.viisage.com

Sung, K.K., and Poggio, T. (1994). "Example-Based Learning for View-Based Human Face Deteciton," *Proc. 23rd Image Understanding Workshop*, pp.843-850, Menlonay, California, USA.

Sung, K.K., and Poggio, T. (1998). "Example-Based Learning for View-Based Human Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20 no.1, pp. 39-51.

Swets, D.L., and Weng, J. (1996). "Using Discriminant Eigenfeatures for Image Retrieval", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 831-836.

Swets, D.L., and Weng, J. (1999). "Hierarchical Discriminant Analysis for Image Retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 386-401.

Tefas, A., Kotropoulos, C., and Pitas, I. (1998). "Variants of Dynamic Link Architecture Based on Mathematical Morphology for Frontal Face Authentication," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 814-819.

Tou, T.J. and Gonzaleze, R. (1973). *Pattern Recognition Principles*, Addison-Wesley, Massachusetts.

Trunk, G.V. (1979). "A Problem of Dimensionality: A Simple Example," *IEEE Trans. Pattern Recognition and Machine Intelligence*, vol. 1, no. 3, pp. 306-307.

Tsukamoto, A., Lee, C.W., and Tsuji, S. (1994a). "Detection and Pose Estimation of Human Face with Synthesised Image Models," *Proc. 12th Int'l Conf. Pattern Recognition*, vol. 1, pp. 754-757, Los Alamitos, Calif.

Tsukamoto, A., and Lee, C.W. (1994b). "Detection and Pose Estimation of Human Face with Multiple Model Images," *IEICE Trans. Inf. and Syst.*, vol. E77-D, no. 11, pp. 1273-1280.

Turk, M.A., and Pentland, A. (1991). "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, no.1. pp. 71-86.

Vieren, C., Cabestaing, F., and Postaire, J. (1995). "Catching Moving Objects with Snakes for Motion Tracking," *Pattern Recognition Letters*, vol. 16, no. 7, pp. 679-685.

Visionic Corporation, http://www.faceit.com

Weber, F., and Hernández, A. (1999). "Face Location by Template Matching with a Quadratic Discriminant Function," *Proc. Int'l Workshop Recognition Analysis and Tracking of Faces and Gestures in Real-time Systems*, pp. 10-13, Corfu, Greece.

Welsh, W.J. (1991). "Model-Based Coding of Videophone Images," *Electronics & Communication Engineering Journal*, pp. 29-36(February).

Yacoob, Y., and Davis, L. (1996). "Recognising Human Facial Expressions from Long Image Sequences Using Optical Flow," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 636-642.

Yang, G., and Huang, T.S. (1994). "Human Face Detection in a Complex Background," *Pattern Recognition*, vol. 27, no. 1, pp. 53-63.

Zhang, J., Yan, Y., and Lades, M. (1997). "Face Recognition: Eigenface, Elastic Matching, and Neural Nets," *Proc. of the IEEE*, Vol. 85, No. 9, pp. 1,423-1,435.