EFFECTS OF LOSSY COMPRESSION ON FACE RECOGNITION ALGORITHMS

by

Mustafa Kamaşak

B.S. in Electrical and Electronic Engineering,

Boğaziçi University, 1997

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science
in
Electrical and Electronic Engineering

Boğaziçi University

1999

ACKNOWLEDGMENTS

I would like to thank Prof. Bülent Sankur for his guidance and support thoughout this work.

I would like to thank Assoc. Prof. Lale Akarun and Assoc. Prof. Muhittin Gökmen for their helpful discussions ans valuable comments.

I would wish to express my gratitute to Aykut Hocanın, Cem Güvener and other friends in BUSIM Lab. for their contributions to this work.

Finally, I would like to thank my family. Without their endless support, encouragement and motivation, this thesis would be just a dream.

ABSTRACT

Face databases can consist of a few hundreds of face images to thousands, even millions. Because of storage and banwidth limitations, face databases are maintained under compressed domain. One of the related problems is the performance evaluation of traditional face recognition techniques on the compressed face images.

In this thesis, we try to determine the effects of information loss due to the compression on the performance of principal face recognition techniques. Besides, the most robust face recognition technique against compression, the extend to which face images can be compressed without a major performance deterioration and the most appropriate compression technique for face images are determined.

Using the results of face recognition experiments on compressed face images, we conclude that the face images can be compressed to 100:1 with face-specific compression tecniques, 40:1 with SPIHT technique and 20:1 with VQ, JPEG and JPEG-2000 techniques. Most robust face recognition techniwue against compression is "Fisherface" method. The eigenfaces generated from compressed face images at 0.4 bit/pixel rate performed better recognition than eigenfaces generated from non-compressed images for VQ, JPEG and JPEG-2000 techniques.

ÖZET

Yüz imgelerinin içerdiği yüz imgelerinin sayısı gün geçtikçe artmaktadır. Imgelerin saklanması için gereken disk sığası ve imge iletimi için gereken bant genişliği kısıtlamalarından dolayı yüz veritabanları sıkıştırılmış olarak saklanmaya başlanmıştır.

Sıkıştırılmış imgeler üzerinde çalışmanın getirdiği problemlerden biri, sıkıştırmadan kaynaklanan bilgi kaybının temel yüz tanıma yöntemlerinin başarımı üzerindeki etkisidir. Bu etkiyi incelerken, başarımı sıkıştırmadan en az etkilenen yüz tanıma yöntemini, yüz imgelerinin, tanıma başarımı düşmeden, değişik sıkıştırma yöntemleri ile ne kadar sıkıştırılabileceğini ve yüz imgelerini sıkıştırmak için en uygun sıkıştırma yönteminin hangisi olduğunu tespit etmeye çalıştık.

Bu çalışma ile, yüz imgelerinin, tanıma başarımı düşmeden, yüzlere özgü sıkıştırma yöntemleriyle 100:1, SPIHT ile 40:1, vektör nicemleme, JPEG ve JPEG-2000 ile 20:1 oranına kadar sıkıştırılabileceği görüldü. Temel yüz tanıma yöntemlerinden "Fisheryüzlerinin" sıkıştırılmış yüz imgeleri üzerinde tanıma başarımı en yüksek yöntem olarak belirlendi. Ayrıca vektör nicemleme, JPEG ve JPEG-2000 ile sıkıştırılmış yüz imgelerinden elde edilen özyuz uzayının özgün yüzlerden elde edilen özyuz uzayındaki tanıma başarımından daha fazla olduğu görüldü.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\mathbf{A}$	State transition probability matrix
$\mathbf{B}$	Observation symbol probability matrix
c	Number of classes
$\mathcal{C}$	Covariance Matrix
$\mathcal{D}$	Dictionary of two dimensional steerable filters
$\mathcal{F}$	Response image at the output of matched filter
$\mathbf{g_i}$	i^{th} basis of projection pursuit filters
m	Dimension of feature vector
n	Dimension of image vector
K	Total number of samples in the training set
N_i	Number of samples from class i in the training set
$\mathbf{O_t}$	Observation symbol at time t
$\mathbf{S}$	Scatter (Covariance) matrix
$\mathbf{S_B}$	Inter-class scatter matrix
$\mathbf{S_W}$	Intra-class scatter matrix
$\mathbf{t}$	2D image template
$\mathbf{u}$	Principal Components
$\mathbf{W}$	Transformation Matrix
$\mathbf{W_{fld}}$	Fisher Linear Discrimination Transform matrix
$\mathbf{W_{opt}}$	Optimal Transformation Matrix
$\mathbf{W_{pca}}$	Karhunen-Loeve Transformation matrix
$\mathbf{x}$	Lexiographically oredered image vector
$\mathbf{y}$	Feature Vector
Ψ	Mean of input images
λ_k	k^{th} eigenvalue in decreasing order
ϵ	Threshold value
$\mathbf{\Pi}$	Initial state distribution

LIST OF ACRONYMS

DCT	Discrete Cosine Transform
DFFS	Distance From Face Space
FLD	Fisher's Linear Discriminant
HMM	Hidden Markov Model
IDCT	Inverse Discrete Cosine Transform
IJG	Independent JPEG Group
JPEG	Joint Picture Expert Group
KLT	Karhunen-Loeve Transform
LDA	Linear Discriminant Analysis
MAP	Maximum a Posteriori
MPF	Matching Pursuit Filter
MSE	Mean Square Error
ORL	Oracle Face Database
PCA	Principal Component Analysis
PDF	Probability Density Function
SNR	Signal to Noise Ratio
SPIHT	Set Partitioning in Hierarchical Trees
VQ	Vector Quantization

1. INTRODUCTION

The remarkable human ability of recognizing hundreds of faces attracted the pyhophysicist, neuroscientists and engineers on various aspects of face processing. A number of experiments were made by neuroscientists on uniqueness of faces, whether face recognition is done holistically or by local feature analysis, how infants perceive faces, organization of memory for faces, inability of recognition due to conditions such as prosopagnosia. These experiments contributed to the face recognition algorithms designed by engineers.

The problem of face recognition can be defined as identifying one or more persons in the scene of a given still image or a video sequence using a stored database of faces. In this thesis, we examine the performance of face recognition algorithms on different lossy compression algorithms. We also try to determine the appropriate compression scheme for face images and determine the extend to which face recognition algorithms can work within acceptable errors in recognition. The most robust algorithm against compression is also determined via a number of conducted experiments.

Deprived of robust mathematical algorithms for feature extraction, this task was too daunting for the researchers. The development of powerful feature extraction techniques in the early 1990s increased significantly the research interest on automatic face recognition problem. One can attribute the increasing interest in face recognition to several factors: An increase in emphasis on civilian/commercial research projects, the re-emergence of neural network classifiers with emphasis on real time hardware and the increasing need for surveillance-related applications due to drug trafficking, terrorist activities etc.

1.1. Applications of Automatic Face Recognition

The conventional security technology, such as passwords, magnetic entrance cards etc. is becoming obsolete with the development of powerful computers and other technical equipments used for fraud. Therefore, new security systems based on the human biometrics is needed. Human identification systems based on biometrics other than face have already led to commercial products with very high identification performances. The iris and fingerprints are widely used biometrics for human identification [1, 2]. However, these systems are not always appreciated by users, as they require some close interaction with the machine often perceived as invasive. Moreover, they require the user to stop at the device and be cooperative, which is acceptable for access control to restricted areas, but not for other applications like surveillance. Therefore, new systems are being developed in which face recognition is integrated to traditional biometric identification systems [3].

In [4], a detailed survey of the automatic face recognition research and the application areas of face recognition are described. A general face recognition application must consist of two parts

- Detection and segmentation of faces from still images or video streams
- Recognition of the segmented faces

In some of the applications only face detection is needed. In this task, still images and video streams are examined in order to determine if there exists a face or not. If there is one or more face objects, it must be segmented from its background. This is sometimes very tedious due to complex background, low image quality, faces with various scales and orientations. Typical examples of the face detection applications are automatic human counters in markets, in terminals etc., advanced alarm systems and many others.

Face detection is also the first step in the face recognition task in many cases. Furthermore, many of the face recognition algorithms cannot handle scale, illumination, and orientation variations in the scene. Therefore, they require a preprocessing stage, in which not only a robust face detection algorithm determines the location of the face in the image (if any), and segments it from the rest of the image, but also normalizes the face image to the desired scale, illumination and orientation.

Typical face recognition applications can be listed as follows

- Personal identification for credit card, driver license, passport, etc. The face images in these type of documents are acquired in a controlled environment, ie. there is no complex background and no variations of scale and orientation. However, the potential database size may be huge.
- Man-machine interface. Although one deals with a small or moderate face database, faces can have very large variations in scale, orientation and expressions. It may also requires the understanding of the human face gestures, ie. anger, happiness, disgust etc.
- Identification for ATM machines and restricted access control. Varying databases from moderate to huge size and face images being acquired in uncontrolled environments make this task more sophisticated than the previous ones.
- Crowd surveillance. This is the most difficult in face recognition. Some application instances can be given as, searching for specific individuals in an airport terminal or for terrorists in illegal demonstrations. All faces in a crowd must be segmented and recognized after normalization by the preprocessing stage. The associated face database can be moderate or large.

There are many other specific applications such as recognition from UV images, recognition of witness reconstructed, hand drawn human caricatures, profile face images etc. Finally, the automatic determination of sex and race is another recognition

related application. The number of applications and their variety make automatic face recognition an attracting research field. From 60s many face-recognizing systems have been developed, and are still being developed with an increasing performance in the correct recognition rate and algorithmic robustness.

Some of these applications have high commercial values, since they perform crucial tasks. In fact for effective law-enforcement in some countries, the research efforts and investment in this area have been intensified. A few of these applications are already been implemented and are commercially available.

1.2. Face Recognition Paradigms

1.2.1. Facial Feature Based Methods

Early face recognition systems were semi-automatic. A human operator used to mark the facial features such as eyes, top of the noise, mouth corners etc. Then using the geometrical relationship of these points such as distance, angle etc., the test faces were recognized. Gradually fully automatic face recognition systems have been developed [5, 6, 7, 8]. In these systems, 10-30 feature points are automatically extracted for recognition tasks. These points are chosen among the most fiducial ones, that do not change with expressions and face obstacles like beard, glasses etc, as shown in FIGURE 1.1.

Although the first systems were not very succesfull, current systems give promising results with facial feature based algorithms.

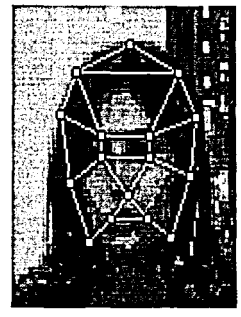

FIGURE 1.1. Facial features selected for recognition

1.2.2. Template Based Methods

The research on face recognition has revealed that the information of a particular face is not only contained necessarily in the coordinates of facial features such as eyes, chin etc. From the view point of information theory, discriminating information is searched not only in the facial features but in the whole face [9].

Template-based methods extract global features of faces and generally represent faces in the lower dimensional feature spaces. To this effect one decorrelates human faces and transforms them to a new space in which their differences from person to person are emphasized.

The most popular algorithms based on templates are correlation, "Eigenfaces" method, "Fisherfaces" method and their variants. However, these algorithms have some problems in common. First of all, they are very sensitive to scale variations and all other changes, which decrease the correlation between templates such as rotation, translation etc. Also since the images are treated as vectors, they cannot capture the local features, and cannot use the information coming from the known topology of human face.

Despite these, they are fast (excluding correlation) enough for real-time applica-

tions, and with proper pre-processing they can be quite robust. With adequate pre-processing, to normalize the input image, these methods can perform with a 90-99% correct recognition rate [10, 11, 12, 13].

1.2.3. Spatial Organization Based Methods

The most common algorithm, which can make use of the spatial organization of faces is Hidden Markov Model. Because of the well-defined topology of faces, the HMM algorithm models the probability of transitions from one of the facial feature to another and the state probabilities.

Each face is thus represented with a different HMM model and any test face can be recognized using the MAP (maximum a posteriori) estimate.

The major drawback of this approach is its computational complexity. A fully connected HMM model is so complex that it cannot be used for practical applications. To reduce this complexity, 2D images can be converted into 1D temporal sequences or 1D spatial sequences. In [14, 15], spatial sequences are preferred. The spatial sequences can be obtained by sampling the image using sliding window. This method converts the image into 1D sequence of windowed data, where each element of the sequence is a vector made of a certain number of samples. It is determined that top-to-bottom line block sampling gives psychologically significant features [15].

1.2.4. Image-Feature Based Methods

In contrast to hand-picked anthropometric features as in Section 1.2, spatial and/or spectral features can be obtained via various interest operators. The most discriminating information can then be searched for by feature selection or extraction methods.

Among such features one can mention DCT coefficients which can be applied on the whole image or face image blocks [16], special wavelet decomposition [17] or Gabor filters [18].

In [17], the features of the face images are extracted using steerable filters [19] which exploits the information in the edges of facial features. According to the problem being addressed, a set of basis are selected. For example, for detection problem the steerable filters that cluster the coefficients of a face object at the output of the filters are chosen.

In [18], the modulus of complex Gabor responses from filters with six orientations and three resolutions are used as features.

1.2.5. Video Based Methods

In surveillance applications, face recognition can be enhanced using video sequences rather than still shots. Video provides a succession of instances of the same face, correlated in time. Some of the advantages of video images for face recognition can be listed as follows:

- Segmentation of moving objects (humans) from a video sequence is easier given their change detection masks
- Multitude of face instances can be fused to improve the recognition rate
- 3D models for faces and/or non-rigid motion analysis can be used
- Speaker recognition can augment the biometric performance

1.2.6. Other Paradigms

To increase the performance and robustness of the face recognition systems, a few face recognition algorithms are cascaded. For example coefficient vectors for different facial features, obtained from a matched filter are discriminated once more to ensure maximum scatter in the new feature space [20, 21], as shown in FIGURE 1.2.

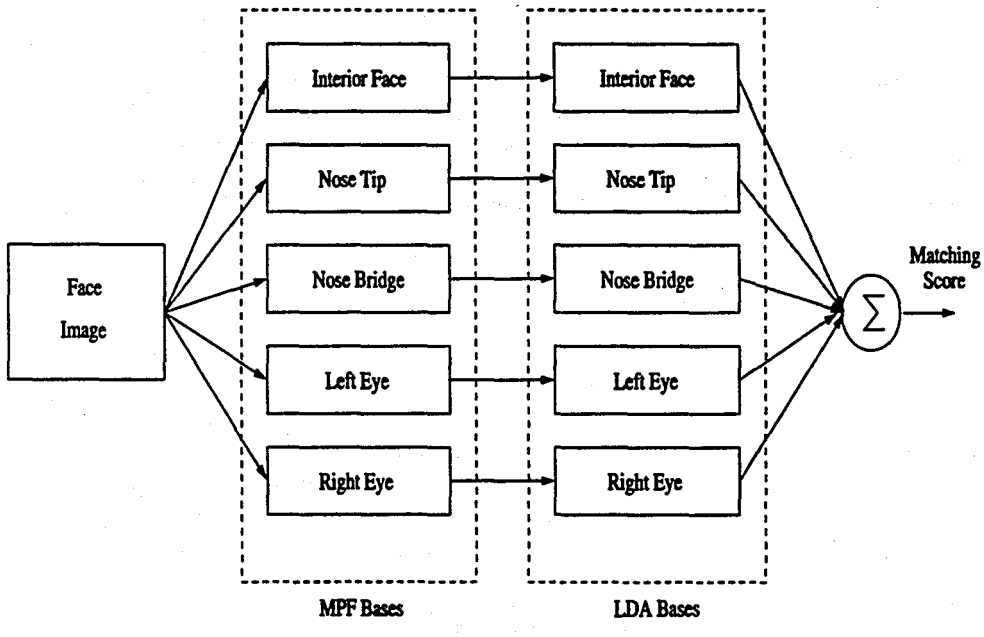

FIGURE 1.2. A novel MPF-LDA based face recognition system [21]

Color information can improve the classification performance and robustness against illumination variations in the scene [2]. Thus approaches have been developed that use hue of the facial complexion.

Neural networks can instrument the construction of templates. They also perform transformation to low dimensional space as in the previous template based methods. Due to the inherent non-linearities, a more effective transformation of the face image and exploitation of nonlinear correlations are expected [2, 20, 22].

Some methods are based on the 3D structure of the face using deformable graphs,

wire meshes etc, which are more complex models [20].

1.3. Scope and Overview of the Thesis

In this thesis, we concentrate on face recognition algorithms and compression schemes. We select the most commonly used template-based methods as face recognition algorithms and prominent compression schemes.

In Chapter 2, the theory and implementation aspects of face recognition algorithms considered in this thesis are explained with a critical review of their weakness and strengths. Thus we detail the "Eigenface" and "Fisherface" methods, Hidden Markov Models, and Projection Pursuit Filters.

In Chapter 3, the compression algorithms used in this work, the reasons for selecting them are discussed. Their compression principles and resulting artifacts at low bit rates are described.

In Chapter 4, the results of our experiments regarding the effects of compression on face recognition algorithms are described.

In Chapter 5, we interpret and try to generalize these results. We have some concluding remarks along with suggestions for future work on this area.

In this thesis, we address the following problems and try to determine:

- The effects of compression schemes on the performance of face recognition algorithms.

- The most robust face recognition algorithm against the face image deformations caused by lossy image compression schemes.
- The entent to which face images can be compressed without a major deterioration in the recognition performance
- The compression scheme that is the most appropriate for the compression of face images, which preserves the facial features and discriminating information in the face images.

Overall, we examined on the one hand three template-based recognition algorithms, on the other hand four compression schemes in order to have an idea about the effects of compression on classification problem.

2. FACE RECOGNITION TECHNIQUES

2.1. Challenges of Face Recognition

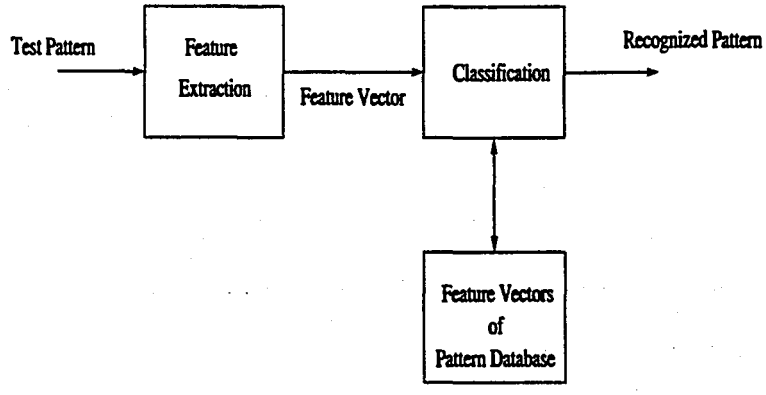

FIGURE 2.1. A general pattern recognition scheme

Face recognition is a very specific type of pattern recognition problem. The very large dimensions of face images and the specific topology of faces, which are highly correlated make this problem at the same time much harder and interesting as compared to usual pattern recognition problems.

Face Variability: In the face recognition problem, a feature extraction method is needed, which is capable of handling very high dimensional vectors and robust enough to counter the variations in the human face, ie., facial expressions, facial accessories (glasses, beard etc.), lighting, pose, scale etc. variations.

In most of the face recognition applications human faces are acquired in an uncontrolled manner. Therefore the incoming faces are generally very noisy and have many obstacles for recognition such as,

- Complex Background

- Resolution (scale) of image
- Illumination conditions of the scene
- Orientation of the face

To cope with these obstacles, a robust preprocessing stage must be included to the system. This stage must perform

- Detection and localization of the faces in the scene
- Normalize its scale
- Compensate for differing lighting conditions
- Adjust face orientation and gaze
- Align facial features

To accomplish these task many methods have been developed. For the detection and localization of faces multi-resolution template matching, use of color and texture information and neural network based methods have been proposed [2].

For the adjustment of head orientation and gaze, geometrical properties of certain points and neural networks are used. After the estimation of face orientation, complex face synthesis methods or other neural network tools are used to obtain a frontal face.

To normalize the illumination variations, some low level image processing algorithms are applied such as histogram equalization. Subtracting the mean of the image compensates the illumination variations, whereas adjusting the image for unit variance may compensate contrast variations between face images.

Another problem is the size of the database. Typical face databases may contain from tens to thousands and even millions of faces. Such a database may require very high capacity for storage and large bandwidth for transmission. For a database consisting of 100,000 faces will require $80kbit/face \times 100,000face = 8Gbit$ and to transmit a face over a $9.6kbit/s$ transmission line will require $\frac{80kbit/face}{9.6kbit/s} = 8.33s$, which is not acceptable for a real-time recognition task.

To overcome these problems face databases must be maintained in compressed format. Concerning the compressed databases, there are two interesting problems to be addressed:

1. To design novel recognition techniques that work directly in the compressed domain

2. To assess the effects of information loss caused by compression on the conventional techniques.

The first problem is a challenging and an increasingly popular problem in the multimedia signal processing. In the face processing context there have been a few attempts [16, 23]. For example in [16], a system is described where the incoming image is fed as input to a neural network structure that outputs of the connected component of the human face and perhaps also including background, shown in FIGURE 2.2. Once such a binary mask is extracted, the human face is isolated and normalized to a given scale, size and illumination and facial features are aligned.

In the recognition stage the distance between block DCT coefficients of the individuals are used. These coefficients are available directly from the bit stream without the need to recur to the IDCT. By allocating more DCT coefficients, the recognition performance is increased.

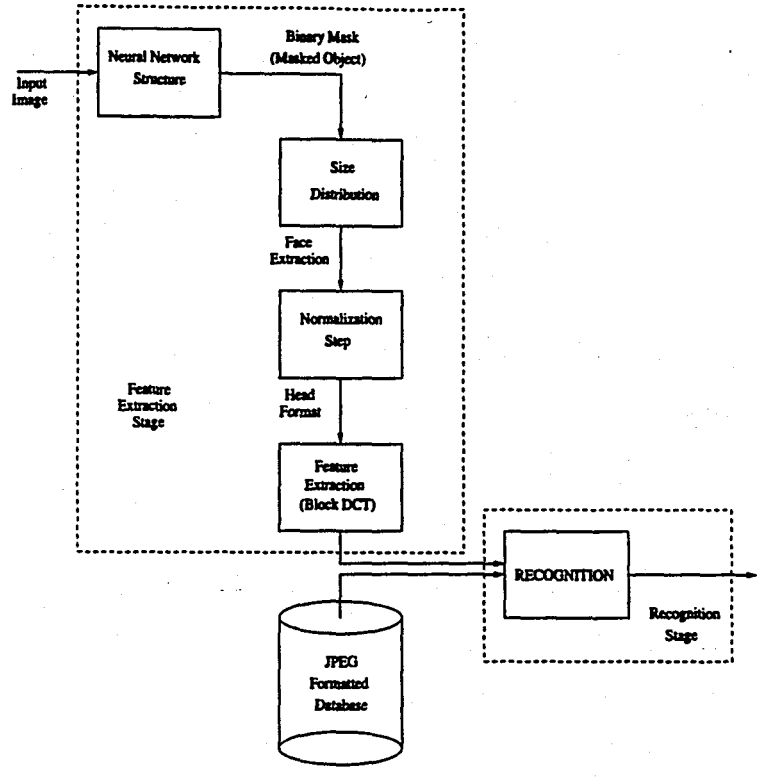

FIGURE 2.2. A recognition algorithm in the compressed domain [16]

In this work, we investigate the second problem. In other words, we try to determine to which rate the faces can be compressed without a major deterioration in recognition performance and which compression algorithm is a better choice for face image compression.

Since we investigate only recognition performance, we choose Oracle Face Database [24], where faces have already been preprocessed. In other words, we bypass any scaling, de-rotation, de-illumination etc. tasks and we concentrate solely on the classification task. In what follows we review the major face recognition techniques considered in this thesis.

2.2. Correlation-based Methods

The simplest method of face recognition is to measure the *correlation* between test images and a set of training images. In [25], Brunelli and Poggio describe a correlation based method for face recognition from frontal views. Their method is based on the matching of templates corresponding to the facial features of relevant significance such as eyes nose and mouth. In order to reduce the complexity of the correlation approach, only the positions of these features are detected and their correlations are examined. The method proposed by Brunelli and Pogio uses a set of templates to detect the eye positions in a new image, by looking for the maximum absolute values of the normalized correlation coefficients of these templates at each point in the test image. To cope with scale variations, a set of five eye templates at different scales was used. However, this method is also computationally expensive. To overcome this problem a hierarchical correlation was used. Once the eyes are located, the detection of other features can take the advantage of these previously estimated positions.

Feature Extraction: Thus, a score matrix is obtained, which contains the score vectors (one score for each feature) of each face in the database.

Classification: The similarity scores of different features can be integrated to obtain a global score. The cumulative score can be computed in several ways:

- Choose the score of the most similar feature
- Sum the feature scores
- Sum the feature scores, using constant weights
- Sum the scores using person-dependent weights.

After the cumulative matching scores are computed, a test face is assigned to the face class for which this score is maximized.

Recognition Performance: The recognition performance reported in [25] using correlation method for frontal faces is higher than 96%. But the face database used for this work was not indicated.

This method is very sensitive to scale, rotation of the image and orientation of the face in the image. The autocorrelation of a face changes very rapidly with scale and rotations in the image plane and rotations in the axis perpendicular to image plane as shown in FIGURE 2.3, FIGURE 2.4 and FIGURE 2.5, given in [25]. In these figures, $D(I)$ denotes the illumination normalized face images.

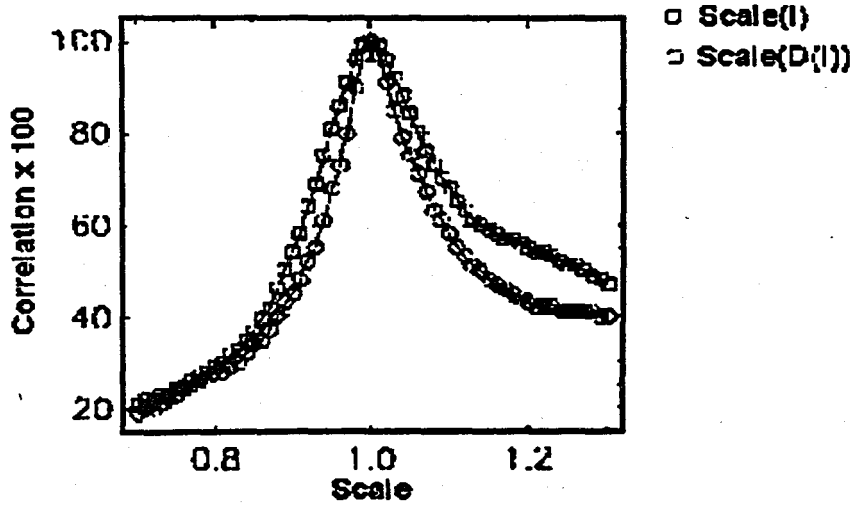

FIGURE 2.3. Change of autocorrelation with scale variations of a face image

The correlation method requires a robust feature detection algorithm to cope with variations in scale, illumination and rotations in image planes and image depth. Besides these the computational complexity of this method $O(Kn^2)$ for full face correlation.

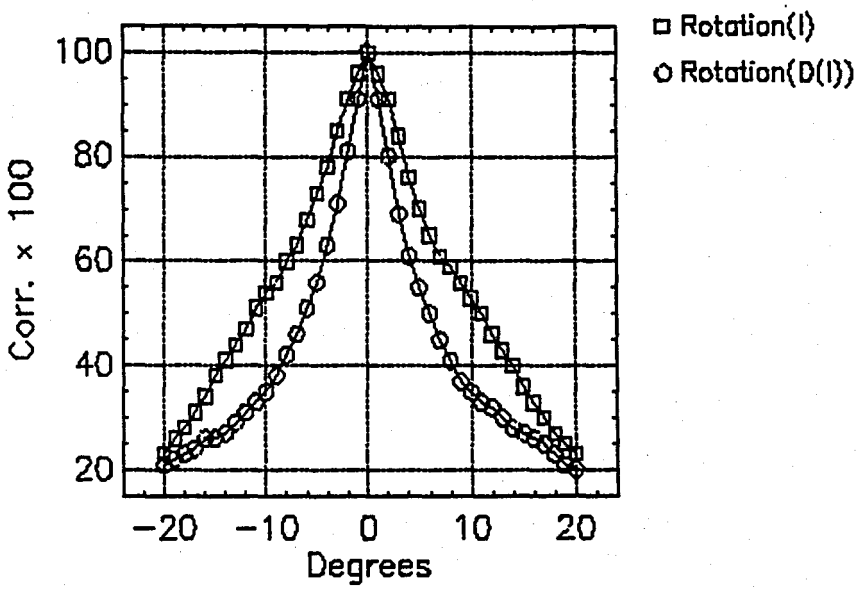

FIGURE 2.4. Change of autocorrelation with rotation of a face image with axis perpendicular to the image plane

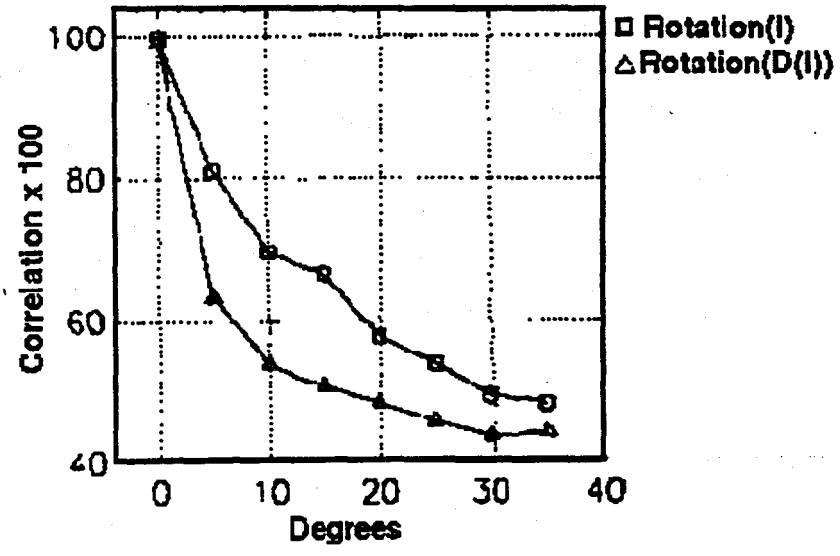

FIGURE 2.5. Change of autocorrelation with rotation of a face image in the image plane

2.3. Eigenface-based Recognition

The "Eigenfaces" method proposed by Turk and Pentland [10] is based on Karhunen-Loeve expansion and is motivated by the earlier work of Sirovitch and Kirby [4] for efficiently representing the picture of faces. The eigenface method presented by Turk and Pentland finds principal components (KL expansion) of the face image distribution, that is the eigenvectors of covariance matrix of the set of face images. These eigenvectors can be thought as a set of coordinates which together characterize the variation between face images.

The faces if represented by the lexicographic ordering of their pixel values, can be thought as points in a huge dimensional space (eg., for an $N \times N$ image $n = N^2$, when one can have $N = 100$, the space dimensions become $n = 10,000$). However, because of the high correlation caused by the similar topology of human faces, all faces are concentrated in a very small portion of this space, as illustrated in FIGURE 2.6. Therefore, we seek a transformation to a new space, where faces are represented in a coordinate space adjusted to this scatter. *"Principal Component Analysis"* is an optimum method which maximizes the projection of the faces along the minimum number of coordinates in the transformed space. Those principal components in the face space, which may appear as ghostly faces, are called "Eigenfaces".

In the eigenface method, the faces are mapped to a new feature space of dimension m, from an n dimensional space, where $m \ll n$, that is

$$\mathbf{x_k} \Longrightarrow \mathbf{y_k} \qquad \mathbf{x_k} \epsilon \mathcal{R}^n \ \ and \ \ \mathbf{y_k} \epsilon \mathcal{R}^m \tag{2.1}$$

In this expression, $\mathbf{x_k}$ denotes the "face image" as a vector consisting of lexicographically ordered n pixels, shown in FIGURE 2.7, while $\mathbf{y_k}$ is the vector of the m eigencoefficients of the same image.

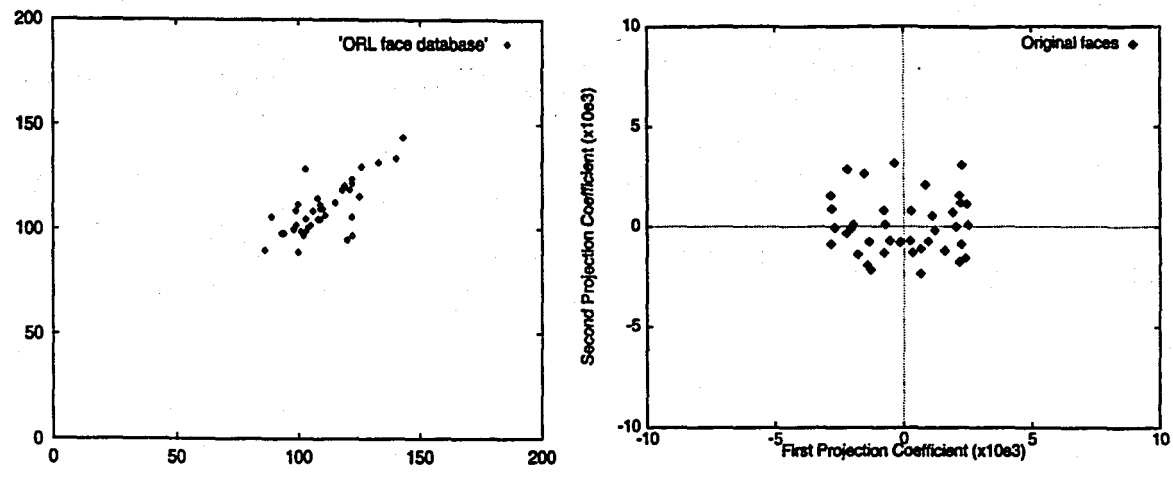

FIGURE 2.6. Scatter of faces in ORL face database before and after PCA

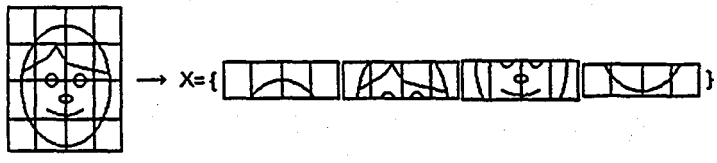

FIGURE 2.7. A lexicographically ordered face image

New feature vectors $\mathbf{y_k}$ can be found by linear transformation:

$$\mathbf{y_k} = \mathbf{W^T x_k} \tag{2.2}$$

where $\mathbf{W} \epsilon \mathcal{R}^{n \times m}$ is a matrix with orthonormal columns. If the scatter matrix $\mathbf{S_T}$ is defined as

$$\mathbf{S_T} = \sum_{k=1}^{K} (\mathbf{x_k} - \mathbf{\Psi})(\mathbf{x_k} - \mathbf{\Psi})^T \tag{2.3}$$

where K is the number of images in the training set and $\Psi \epsilon \mathcal{R}^n$ is the mean of the images in the training set, then one can choose $\mathbf{W_{opt}}$ such that

$$\mathbf{W_{opt}} = arg \max_{\mathbf{W}} |\mathbf{W^T S_T W}| \tag{2.4}$$

Columns of $\mathbf{W_{opt}}$ correspond to the eigenvectors of the scatter matrix, which have the m largest eigenvalues. Then all faces in the database are projected to this new space and their feature vectors are calculated.

Notice however that, maximized scatter contains not only interclass variations, but also intraclass variations, which is unwanted information for classification problems. The variations within a class may result from illumination conditions, facial expressions, poses and accessories. Because of the intraclass variations, classes may not well be clustered in the new feature space.

Implementation: Let a face image $\mathbf{I(x, y)}$ be a two dimensional array of intensity values, or a vector of dimension n after lexicographic ordering. Let the training set of images be $\mathbf{I_1, I_2, ..., I_K}$. The average face image of the set is defined by

$$\Psi = \frac{1}{K} \sum_{i=1}^{K} \mathbf{I_i} \tag{2.5}$$

Each face differs from the the average by the vector $\Phi_i = \mathbf{I_i} - \Psi$. This set of vectors is subjected to the principal component analysis, which seeks a set of orthonormal vectors $\mathbf{u_k}$, $k = 1, ..., m$ and their associated eigenvalues λ_k, which best describe the distribution of data. The vectors $\mathbf{u_k}$ and scalars λ_k are the eigenvectors and eigenvalues of the covariance matrix:

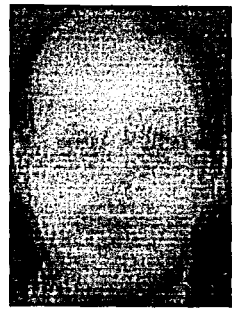

FIGURE 2.8. Mean face

$$C = \frac{1}{K}\sum_{i=1}^{K} \Phi_i \Phi_i^T = AA^T \qquad C \epsilon R^{n \times n} \tag{2.6}$$

where the matrix $A = [\Phi_1, \Phi_2, ..., \Phi_N]$. Finding the eigenvectors of matrix $C_{n \times n}$ is computationally too expensive. However, the eigenvectors of C can be determined by first finding the eigenvectors of a much smaller matrix of size $K \times K$ and taking a linear combination of the resulting vectors [10].

If $\mathcal{L}$ is defined as

$$\mathcal{L} = A^T A \qquad \mathcal{L} \epsilon R^{K \times K} \tag{2.7}$$

and eigenvectors of $\mathcal{L}$ can be found easily by

$$A^T A v_i = \lambda_i v_i \tag{2.8}$$

by multiplying both sides of equation by A, we get

$$AA^T A v_i = \lambda_i A v_i \tag{2.9}$$

$$\mathcal{C}\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{A}\mathbf{v}_i \tag{2.10}$$

So, eigenvectors of the covariance matrix $\mathcal{C}$, can be found using the weighted eigenvectors of $\mathcal{L}$.

$$\mathbf{u}_i = \sum_{k=1}^{K} \mathbf{v}_{ik}\Phi_k \qquad i = 1, ..., K \tag{2.11}$$

The resulting eigenvectors, shown in FIGURE 2.9 resemble ghostly human faces,and they are called "Eigenfaces".

FIGURE 2.9. First eight eigenfaces

Feature Extraction: The space spanned by the covariance matrix $\mathcal{C}$ is called the "Face Space". The eigenvectors of matrix $\mathcal{C}$, which are also called eigenfaces, form a basis set for the face images. A new face image $\mathbf{x}$ is projected onto face space by:

$$y_k = \mathbf{u}_k(\mathbf{x} - \Psi) \qquad k = 1, 2, ..., m \tag{2.12}$$

The projections y_k form the feature vector $\mathbf{y} = [y_1, y_2, ..., y_m]$ which describes the contribution of each eigenface in representing the input image.

Classification: Given a set of face classes and the corresponding feature vectors $\mathbf{y}^q$, the simplest method, for determining which face class the test image belongs to, is to find the face class k that minimizes the Euclidean distance in the feature space:

$$arg\min_k = \|(\mathbf{y} - \mathbf{y}^k)\|^2 \tag{2.13}$$

where $\mathbf{y}$ is the eigenfeature vector of the test image.

If this distance is above a threshold ϵ, then the test face is classified as an unknown face.

In [10], Turk and Pentland used the concept of *Distance From Face Space* metric to decide about the nature of any new pattern. This metric is simply the squared distance between mean adjusted image $\Phi = \mathbf{x} - \Psi$ and its reconstruction from the projection coefficients on the face space $\Phi_\mathbf{f} = \sum_{i=1}^{m} y_i \mathbf{u}_i$.

$$\epsilon^2 = \|\Phi - \Phi_\mathbf{f}\|^2 \tag{2.14}$$

They have described four possibilities for an input image and a pattern vector, shown in FIGURE 2.10. These possibilities are listed in TABLE 2.1. If a projected image is

- Near to a face class and near to the face space, it is classified as the nearest face in the face space,
- Far to a face class but near to the face space, it is a face image but it is not one of the faces in the database
- Near to a face class but far to the face space, it is not a face image

TABLE 2.1. Interpretation of the projection of an unknown image onto the face space

	Face Space	Known Face Class	Result
1	near	near	Recognized as the nearest face
2	near	far	unknown face
3	far	near	not a face
4	far	far	not a face

- Far to a face class and far to the face space, it is not a face image

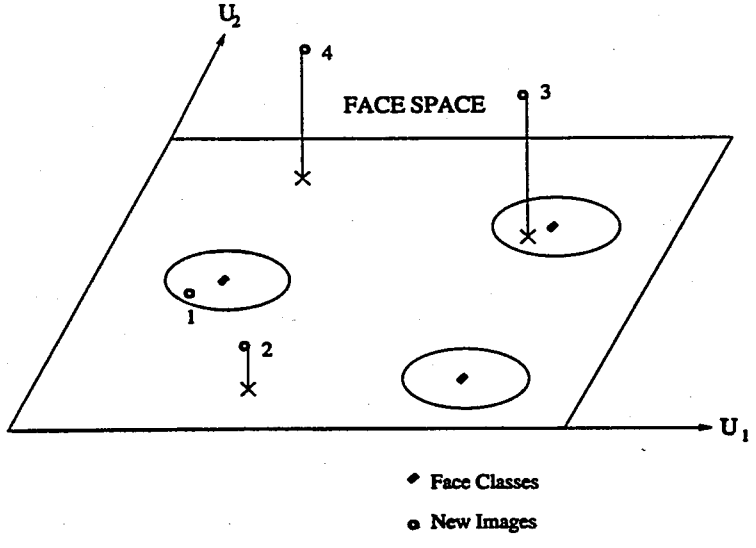

FIGURE 2.10. Projection of an image onto face space

Recognition Performance: The "Eigenface" method was tested on a relatively large database named FERET. Various groups of sixteen images corresponding to sixteen different subjects were selected and used as a training set. The reported recognition performances are 96% correct classification under illumination variations, 85% correct classification over orientation variation and 64% correct classification over size variations [12]. It can be seen that this approach is fairly robust to illumination changes but degrades quickly as the scale changes. This can be described by the low correlation between the different scales of face images.

2.3.1. The View-Based Approach

Based on the eigenface decomposition, Pentland et al. [11] developed a "view-based" eigenspace approach for human face recognition under general viewing conditions. Given N individuals under P different views, shown in FIGURE 2.11, recognition is performed over P separate eigenspaces, each capturing the variation of the individuals in a common view. The "view-based" approach is essentially an extension of the eigenface method, which best describes the selected input image. This is accomplished by calculating the residual description error (distance from face space: DFFS) for each view space. Once the proper view is determined, the image is projected onto appropriate view space and then recognized. The view-based approach is computationally more expensive than the normal eigenface method.

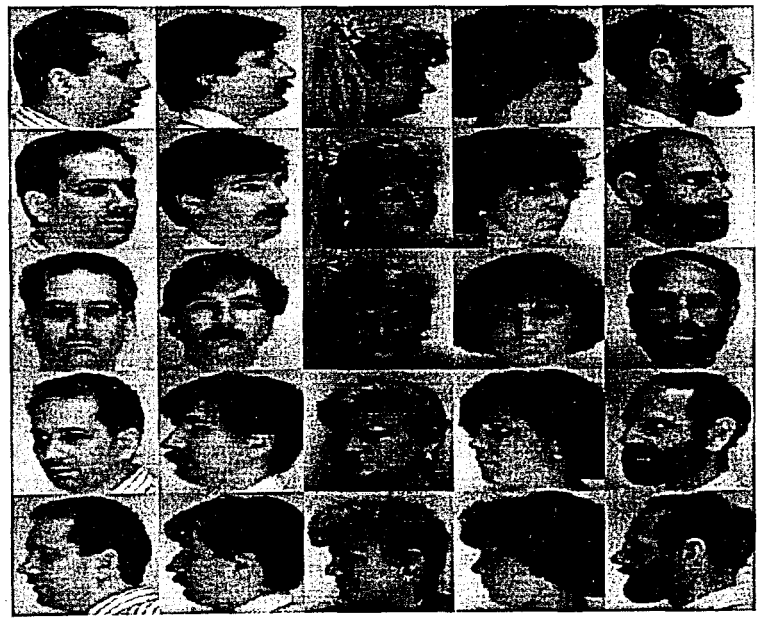

FIGURE 2.11. Multiple views for rotation invariance

Recognition Performance: The recognition performance of the view-based and parametric approaches was evaluated on a database of 189 images, nine views of 21 people [11]. The nine views of each person were tested by training on a subset of the available views of $\pm 90^0$, $\pm 45^0$ and $\pm 0^0$ and testing on the intermediate views $\pm 68^0$, $\pm 23^0$ (interpolation performance). In TABLE 2.2, the performance of this approach is listed.

Column titles indicate the space used for recognition, whereas row titles indicate the nature of test image.

TABLE 2.2. Performance of view-based eigenface method

	Frontal	Half Left	Half Right	Profile Left	Profile Right
Frontal	99	**	**	**	**
Half Left	**	87	38	**	**
Half Right	**	38	82	**	**
Profile Left	**	**	**	70	32
Profile Right	**	**	**	32	68

2.3.2. Recognition Using Eigenfeatures

In [11], Pentland *et. al.* discussed the use of facial features for face recognition. This can be viewed as either a modular or layered representation of the face, where a coarse (low resolution) description of the whole head is augmented by additional (high resolution) details in terms of salient facial features. The eigenface technique was extended to detect facial features. For each one of the facial features, a feature space is built by selecting the most significant eigenfeatures (eigenvectors corresponding to the largest eigenvalues of the features' correlation matrix). In the eigenfeature representation the equivalent "distance from face space" (DFFS) can be effectively used for the detection of facial features under different viewing geometries by using a view-based eigenspace.

Feature Extraction: After the facial features in a test image were extracted, a score of similarity between the detected features and the features corresponding to the model images is computed. The technique used to determine this score is an extension of the eigenface method.

Classification: A simple approach for recognition is to compute a cumulative score in terms of equal contribution by each of the feature scores. Once the cumulative score is determined, the test face is classified so as to minimize this score.

Recognition Performance: The eigenfaces and eigenfeatures was combined and tested on a set of 45 individuals with two views per person corresponding to different facial expressions (neutral vs. smiling). The neutral set of images was used as a training set and the recognition was performed on the smiling set. Since the difference between these particular facial expressions is primarily articulated in the mouth, this feature was discarded for recognition purposes. The recognition results showed that the eigenfeatures alone were sufficient in achieving a recognition rate of 95%, equal to that of the eigenfaces. When a combined representation of eigenfaces and eigenfeatures was tested a recognition rate of 98% was reported, shown in FIGURE 2.12.

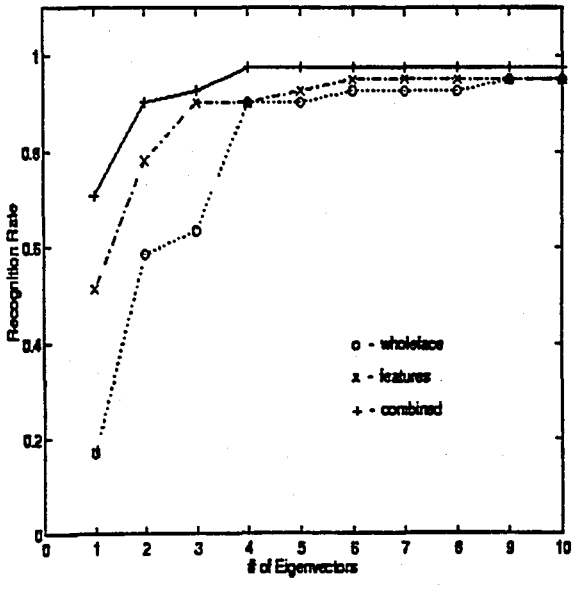

FIGURE 2.12. Performance of the eigenfeature technique [11]

2.4. Fisherfaces

In [13], a new method for reducing the dimensionality of the feature space by using Fisher's Linear Discriminant(FLD) is proposed. The FLD uses the class membership information and develops a set feature vectors in which variations of different faces are emphasized while different instances of the same face due to illumination conditions, facial expressions and orientations are de-emphasized.

2.4.1. Fisher's Linear Discriminant

Given c be the number of classes, let N_i be the number of samples in class i, $i = 1, 2, ..., c$. Obviously K used in Section 2.3. is related as $K = \sum_{i=1}^{c} N_i$. Then the following positive semidefinite scatter matrices are defined as:

$$\mathbf{S_B} = \sum_{i=1}^{c} (\Psi^i - \Psi)(\Psi^i - \Psi)^T \tag{2.15}$$

$$\mathbf{S_W} = \sum_{i=1}^{c} \sum_{i=1}^{c} (\mathbf{x^i} - \Psi^i)(\mathbf{x^i} - \Psi^i)^\mathbf{T} \tag{2.16}$$

where $\mathbf{x^i}$ denotes the i^{th} n dimensional vector, and Ψ^i is the mean of class i:

$$\Psi^i = \frac{1}{N_i} \sum_{i=1}^{N_i} \mathbf{x^i} \tag{2.17}$$

and Ψ is the overall mean of sample vectors:

$$\Psi = \frac{1}{c}\sum_{i=1}^{c}\Psi^i \tag{2.18}$$

$\mathbf{S_W}$ is the *within-class scatter matrix* and represents the-average scatter of sample vector class i; $\mathbf{S_B}$ is the *between-class scatter matrix* and represents the scatter of the mean Ψ^i of class i around the overall mean vector Ψ. If $\mathbf{S_W}$ is non-singular, the *Linear Discriminant Analysis (LDA)* selects a matrix $\mathbf{W_{opt}} \epsilon \mathbf{R}^{n \times m}$ with orthonormal columns which maximizes the ratio of the determinant of the between-class scatter matrix of the projected vector samples to the determinant of the within-class scatter matrix of the projected samples:

$$\mathbf{W_{opt}} = arg\max_{\mathbf{W}} \frac{|\mathbf{W}^T\mathbf{S_B}\mathbf{W}|}{|\mathbf{W}^T\mathbf{S_W}\mathbf{W}|} = [\mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_m}] \tag{2.19}$$

where $[\mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_k}]$ is the set of generalized eigenvectors of $\mathbf{S_B}$ and $\mathbf{S_W}$ corresponding to the set of decreasing eigenvalues of

$$\mathbf{S_B}\mathbf{w_i} = \lambda_i \mathbf{S_W}\mathbf{w_i} \qquad i = 1, 2, ..., m \tag{2.20}$$

Since there are at most $c-1$ nonzero eigenvalues, the upper bound of m is $c-1$.

2.4.2. Face Recognition Using Linear Discriminant Analysis

Let a training set of K images represent c different subjects. Different instances of a person's face (variations in lighting, pose or facial expressions) are defined to be in the same class and faces of different subjects are defined to be from different classes.

Feature Extraction: The scatter matrices S_B and S_W are defined in (2.15) and (2.16) However, the matrix $\mathbf{W_{opt}}$ cannot be found directly from (2.19) because the matrix $\mathbf{S_W}$

is generally singular. This stems from the fact that the rank of $\mathbf{S_W}$ is less than $K-c$, and in general, the number of pixels in each image is much larger than the number of images in the learning set. To overcome this problem, a method is proposed in [13], which was called Fisherfaces method. The problem of $\mathbf{S_W}$ being singular is bypassed by projecting the image set onto a lower dimensional space so that the resulting within-class scatter matrix is nonsingular. This is achieved by using Principal Component Analysis (PCA) to reduce the dimension of the feature space to $K-c$ and then applying the standard linear discriminant defined in (2.19) to reduce the dimension to $c-1$.

$$\mathbf{W_{opt}} = \mathbf{W_{pca}}\mathbf{W_{fld}} \tag{2.21}$$

where

$$\mathbf{W_{pca}} = arg\max_{\mathbf{W}} |\mathbf{W}^T\mathbf{S_T}\mathbf{W}| \tag{2.22}$$

and

$$\mathbf{W_{fld}} = arg\max_{\mathbf{W}} \frac{|\mathbf{W}^T\mathbf{W_{pca}}^T\mathbf{S_B}\mathbf{W_{pca}}\mathbf{W}|}{|\mathbf{W}^T\mathbf{W_{pca}}^T\mathbf{S_W}\mathbf{W_{pca}}\mathbf{W}|} \tag{2.23}$$

where $\mathbf{S_T}$ is the covariance matrix of the set of training images computed in 2.3. The columns of $\mathbf{W_{opt}}$ are orthogonal vectors, which are called Fisherfaces. Unlike the Eigenfaces the Fisherfaces do not correspond to face-like patterns.

Classification: The classification is based on the Euclidean distance between the coefficient vectors in the feature space.

In [26], a weighted distance metric was proposed. The weights for each principal axis is determined based on the reliability of the axis. The weight of axis i, α_i was calculated as

$$\alpha_i = \frac{\lambda_i}{\sum_{j=1}^{m}\lambda_j} \tag{2.24}$$

then, the distance between two feature vector is

$$d(\mathbf{y}, \mathbf{y}') = \sum_{i=1}^{m} \alpha_i (y_i - y_i')^2 \tag{2.25}$$

Recognition Performance: The reported recognition performance of this scheme in [13] is 99.4% under variations in lighting, facial expressions and eye wear (glasses, no-classes), using Yale Database [27]. This database consists of 10 different views from 16 individual, with high variations in lighting and facial expressions. On the same database, the recognition rate reported when using the eigenface method was 80%. The database did not include images with variations in pose or orientation. The training set contained five set of ten face images taken under strong variations in illumination, facial details but no variations in pose.

2.5. Hidden Markov Models

2.5.1. Hidden Markov Models

Hidden Markov Models (HMM) are a set of statistical models used to characterize the temporally or spatially varying properties of a signal. Rabiner [28] provides an extensive and complete tutorial on HMMs. HMM are made of two interrelated processes:

1. An underlying, unobservable Markov chain with finite number of states, a state transition probability matrix and an initial state probability distribution.
2. A set of probability density functions associated to each state.

The elements of a HMM are:

- N, the number of states in the model. If S is the set of states, then $S = S_1, S_2, ..., S_N$. The state of the model at time t is given by $q_t \epsilon S, 1 \leq t \leq T$ where T is the length of the observation sequence.

- M, the number of different observation symbols. If V is the set of all possible observation symbols (also called the *codebook* of the model), then $V = v_1, v_2, ..., v_M$.

- **A**, the state transition probability matrix, ie. $\mathbf{A} = a_{ij}$ where

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i] \qquad 1 \leq i, j \leq N \tag{2.26}$$

with the constraint,

$$0 \leq a_{i,j} \leq 1 \tag{2.27}$$

and,

$$\sum_{j=1}^{N} a_{i,j} = 1 \qquad 1 \leq i \leq N \tag{2.28}$$

- **B**, the observation symbol probability matrix, $\mathbf{B} = b_j(k)$ where

$$b_j(k) = P[\mathbf{O_t} = v_k | q_t = S_j] \qquad 1 \leq k \leq M \tag{2.29}$$

and $\mathbf{O_t}$ is the observation symbol at time t.

- $\mathbf{\Pi}$, the initial state distribution, $\mathbf{\Pi} = \pi_i$ where

$$\pi_i = P[q_1 = S_i] \qquad 1 \leq i \leq N \tag{2.30}$$

Using a shorthand notation, a HMM is defined as:

$$\lambda = (\mathbf{A}, \mathbf{B}, \mathbf{\Pi}) \tag{2.31}$$

The above characterization corresponds to a discrete HMM, where the observations are characterized as discrete symbols chosen from a finite alphabet $V = v_1, v_2, ..., v_M$ In a continuous density HMM, the states are characterized by continuous probability density function (pdf) is a finite mixture of the form:

$$b_i(O) = \sum_{k=1}^{M} c_{ik} N(O, \mu_{ik}, \mathbf{U_{ik}}) \qquad 1 \leq i \leq N \tag{2.32}$$

where c_{ik} is the mixture coefficient for the k^{th} mixture in state i. Without loss of generality $N(O, \mu_{ik}, U_{ik})$ is assumed to be a Gaussian pdf with mean vector μ_{ik} and covariance matrix $\mathbf{U_{ik}}$.

2.5.2. Face Recognition Using HMM

HMMs have been used extensively for speech recognition, where data is one dimensional along the 1D axis. However, the equivalent fully connected two dimensional HMM would lead to a very high computational problem. Attempts have been made to use multi-model representation that lead to pseudo 2D HMM [14]. These model are currently used in character recognition.

Samaria *et. al.* proposed the use of the 1D continuous HMM for face recognition [14]. Assuming that each face is an upright, frontal position, features will occur in a predictable order, i.e. forehead, eyes, nose etc. This ordering suggests the use of a top-bottom model, where only transitions between adjacent states in a top to bottom manner are allowed [14] if the images are taken under small rotations. The states of the model correspond to the significant facial regions, forehead, eyes, nose, mouth and chin. Each of these facial regions is assigned to a state in a left to right 1D continuous HMM. The state structure of the face model and the non-zero transition probabilities, a_{ij} given in 2.26, are shown in FIGURE 2.13.

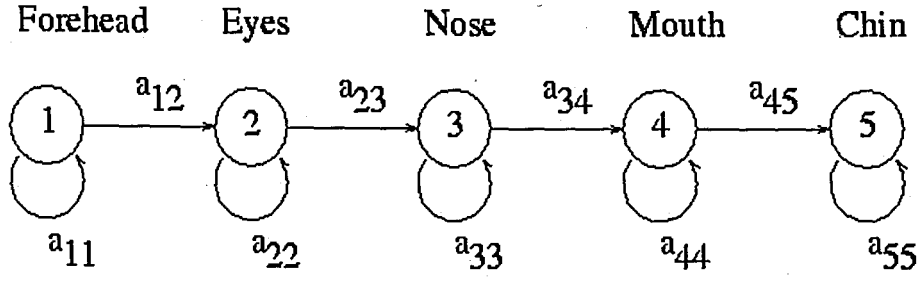

FIGURE 2.13. Left to right HMM for face recognition

The observation sequence **O** is generated from an $X \times Y$ image using an $X \times L$ sampling window with $X \times M$ pixels overlap, shown in FIGURE 2.14. Each observation vector is a block of L lines. There is an M line overlap between successive observations. The overlapping allows the features to be captured in a manner which is independent of vertical position, while a disjoint partitioning of the image could result in the truncation of features occurring across block boundaries. With no overlap and a small height of the sampling window is used, the segmented data may not correspond to significant facial features. However, as the window height increases there is a higher probability of cutting across the features.

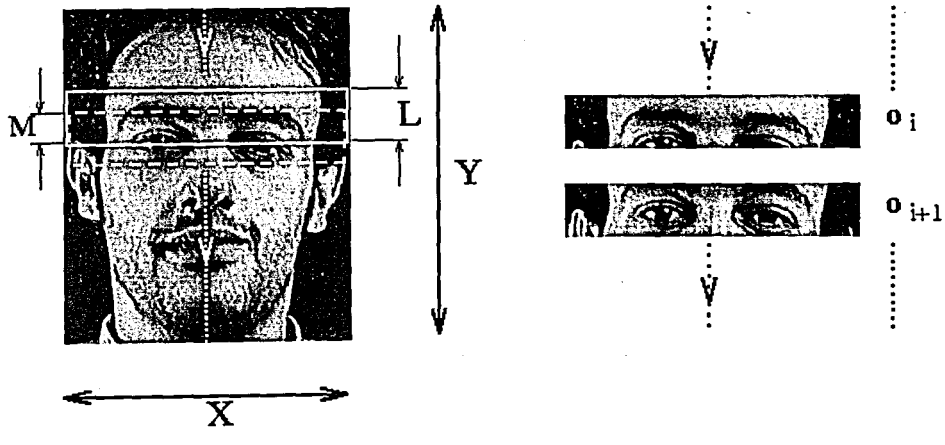

FIGURE 2.14. Image sampling technique for HMM recognition

Training: Each individual in the database is represented by a HMM face model. A set of images representing the same face are used to train each HMM. First, the HMM

$\lambda = (\mathbf{A}, \mathbf{B}, \mathbf{\Pi})$ is initiated. The training data is uniformly segmented from top to bottom in 5 states and the observation vectors, obtained from DCT coefficients, associated with each state are used to obtain initial estimates of the observation probability matrix $\mathbf{B}$. The goal of the training stage is to optimize the parameters $\lambda_i = (\mathbf{A}, \mathbf{B}, \mathbf{\Pi})$ to "best" describe, the observations $\mathbf{O} = o_1, o_2, ..., o_T$ in the sense maximizing $P(\mathbf{O}|\lambda)$. The general HMM training scheme is a variant of K-means iterative procedure for clustering data:

1. The training images are collected for each subject in the database to generate the observation sequence.

2. A common prototype model is constructed with the purpose of specifying the number of states in the HMM and the state transitions allowed.

3. A set of initial parameter values using the training data and the prototype model are computed iteratively. The goal of this stage is to find a good estimate for the observation model probability $\mathbf{B}$. Good initial estimates of the parameters are essential for rapid and proper convergence (to the global maximum of the likelihood function). On the first cycle, the data is uniformly segmented, matched with each model state and the initial model parameters are extracted. On successive cycles, the set of training observation sequences are segmented into states via the Viterbi algorithm. The result of segmenting each of the training sequences is that for each of the N states, a maximum likelihood estimate of the set of observations that occur within each state according to the current model is obtained.

4. Following the Viterbi segmentation, the model parameters are re-estimated using Baum-Welch re-estimation procedure. This procedure adjusts the model parameters so as to maximize the probability of observing the training data, given each corresponding model.

5. The resulting model is then compared to the previous model (by computing a distance score that reflects the statistical similarity of the HMMs). If the model distance score exceeds a threshold, then the old model λ is replaced by new model

$\hat{\lambda}$, and the overall training loop as repeated. If the model distance falls below a threshold, then the model convergence is assumed and the final parameters are saved.

Recognition: Recognition is carried out by matching the test image against each of the trained models. To do this, the image is converted to an observation sequence and then model likelihoods $P(O_{test}|\lambda_i)$ are computed for each λ_i $i = 1, 2, ..., c$. The model with highest likelihood reveals the identity of the unknown face, as shown in FIGURE 2.15.

$$v = arg \max_{1 \leq i \leq c} [P(\mathbf{O_{test}}|\lambda_i)] \tag{2.33}$$

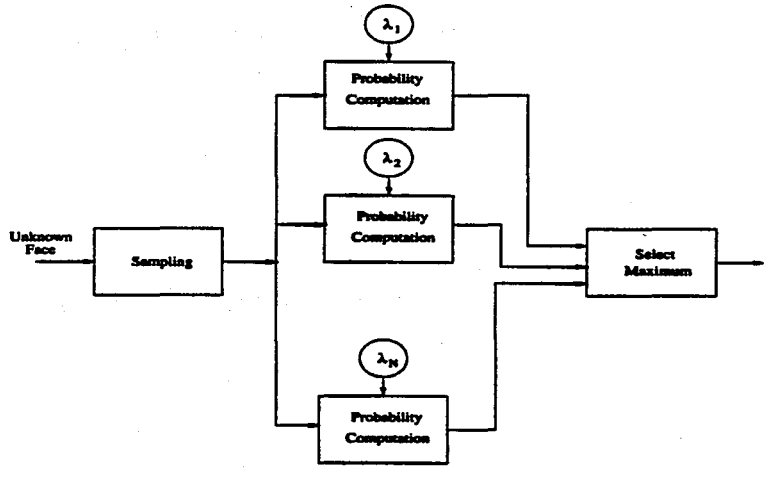

FIGURE 2.15. HMM recognition scheme

Recognition Performance: The recognition performances were tested on a small database of 50 images that were not part of the training dataset of 24 images [15]. The images in the test set contain faces with different facial expressions, facial details (glasses,no-glasses) and variations in lighting. On this database the reported recognition rate was 84%. On the same database the recognition rate obtained by running the eigenface method was 73%.

2.6. Matching Pursuit Filters based Methods

2.6.1. Matching Pursuit Filters

The original pursuit idea of Mallat and Zhang [29] uses a greedy heuristic to iteratively construct a best-adapted decomposition of a function f on $\mathcal{R}$. The algorithm works by choosing at each iteration i the wavelet g in the dictionary $\mathcal{D}$ that has maximal projections onto residue of f. The best-adapted decomposition is selected by the following greedy strategy. Let $R^0 f = f$; then g_i is chosen such that

$$|\langle R^i f, g_i \rangle| = \max_{g \in \mathcal{D}} |\langle R^i f, g_i \rangle| \tag{2.34}$$

where

$$R^{i+1} f = R^i f - \langle R^i f, g_i \rangle g_i \tag{2.35}$$

for $i \geq 1$.

Each wavelet in the expansion is selected by maximizing the right hand term in (2.34) This equation allows for an expansion on a single function, and minimizes the reconstruction error.

In order to extend this algorithm to pattern recognition, the right hand term in 2.34 must be replaced with a cost function C_g, which allows for

- The simultaneous expansion of multiple templates (functions)
- Incorporates knowledge of the pattern recognition problem being addressed

Also a two dimensional wavelet dictionary must be used to extend from f on $\mathcal{R}$ to

$\mathbf{t}$ on $\mathcal{R}^2$.

2.6.2. Matching Pursuit Filters for Detection

In this scheme, a particular image object (eg. face or facial features) is represented as an m-dimensional vector $(y_0, y_1, ..., y_{n-1})$ called a coefficient vector. One computes the coefficient values y_i by projecting the object image onto a basis set $(\mathbf{g_0}, \mathbf{g_1}, ..., \mathbf{g_{m-1}})$, which need not be orthogonal. When the basis is not orthogonal, an iterative projection algorithm can calculate the coefficient vector. The projection algorithm adjusts for the nonorthogonality by using residual images. If t is an image or template, then $\mathbf{R_0 t} = \mathbf{t}$. The coefficient y_i is the projection of the residual image $\mathbf{R^i}$ onto the basis element $\mathbf{g_i}$

$$y_i = \langle \mathbf{R^i t}, \mathbf{g_i} \rangle \tag{2.36}$$

where $\langle ., . \rangle$ is the inner product between two functions. The residual image is updated after each iteration by

$$\mathbf{R^i t} = \mathbf{R^{i-1}} - y_{i-1}\mathbf{g_{i-1}} \tag{2.37}$$

for $i \geq 1$.

After the n^{th} iteration, an image t is decomposed into a sum of residual images:

$$\mathbf{t} = \sum_{i=0}^{n-1} (\mathbf{R^i t} - \mathbf{R^{i+1} t}) + \mathbf{R^n t} \tag{2.38}$$

Rearranging 2.37 and substituting in 2.38 yields

$$\mathbf{t} = \sum_{i=0}^{n-1} y_i \mathbf{g_i} + \mathbf{R^n t} \tag{2.39}$$

and the approximation of the original image after n iterations is

$$\hat{\mathbf{t}} = \sum_{i=0}^{n-1} y_i \mathbf{g_i} \tag{2.40}$$

The approximation need not be accurate, because only enough information is encoded to allow detection or classification.

The goal of the algorithm is to determine whether an observed pattern belongs to a particular class. Hence, there must be a way of measuring the similarity between two patterns. With matching pursuit filters, one compares the coefficient vectors from two patterns, where coefficient vectors are generated using the same basis. The similarity measure between two patterns is the angle between their coefficient vectors, [17]. This measure is invariant to linear changes in the contrast of the image. Furthermore, if the basis is composed of wavelets, then the similarity measure is also invariant to the illumination level in the image [17].

Consider the nose object in images. Ideally all noses would have the same coefficient vector, and all occurrences of this vector would be a nose. Unfortunately this does not occur due to variability of noses. Therefore in [17], all nose instances are clustered and the centroid of the cluster is selected as the basis. The vector in the center of the cluster is called *proto-feature*, and it represents an average feature.

The matching pursuit filter is trained on K different examples of an object. Let $t_1, ..., t_K$ be the K examples of objects, where $\mathbf{t_i}$ represents the i^{th} example of the object.

The objects are aligned in the templates so that the center of the object is origin. The algorithm selects the basis elements from the dictionary $\mathcal{D}$, where $\mathcal{D}$ is composed of 2-D directional wavelets. These wavelets were chosen because they encode information locally at different scales and orientations. The basis elements in the dictionary do not span the space of all possible images. The dictionary excludes high-frequency wavelets to reduce the effect of high-frequency noise. Low-frequency wavelets are also excluded in order to avoid the encoding the information in the background. For face recognition, a dictionary derived from the second partial derivatives of Gaussian densities and their Hilbert transforms due to their directional edge detection ability.

A greedy algorithm is used to select the basis elements. In iteration i, the basis function $\mathbf{g_i}$ is selected. The choice being a function of the residual image $\mathbf{R^i t_l}$ and coefficients α_j^l from previous iterations. Let the coefficient $\alpha_j^l = \langle \mathbf{R^j t_l}, \mathbf{g_i} \rangle$, that is, the j^{th} coefficient for object l. The set of coefficients generated through the i^{th} iteration is denoted by $\Omega_i = U_l(\alpha_0^l, \alpha_1^l, ..., \alpha_i^l)$, $i \geq 0$ and $\Omega_{-1} = \emptyset$.

Each iteration of the basis selection algorithm consists of three steps. In the first step, a new basis function g_i is selected, as in (2.41). In the second step, the coefficient vectors for each object t_l are updated. In the third step, the residual images are updated by $\mathbf{R^{i+1} t_l} = \mathbf{R^i t_l} - \alpha_i^l \mathbf{g_i}$. The i^{th} basis function is selected by the following cost function

$$\mathbf{g_i} = arg \min_{g \epsilon \mathcal{D}} C_g(R^i t_1, ..., R^i t_m, \Omega_{i-1}) \tag{2.41}$$

where C_g measures how well the coefficient vectors cluster when the i^{th} basis is g_l. The function C_g is evaluated for each $g \epsilon \mathcal{D}$, and the g that minimizes C_g is selected as the basis element g_i. In [17], C_g for a given g, the cluster is the mean of $(\alpha_0^l, ..., \alpha_{i-1}^l, \langle R^i t_l, g \rangle)$, $1 \leq l \leq m$. Once the cluster vector is determined, C_g computes the average distance from the coefficient vectors to the cluster vector. This distance is a measure of scatter (variance) of the coefficient vectors about the cluster vector. If the dispersion is small, then

g is a good candidate for g_i; on the other hand, if the dispersion is large, then g is a poor choice.

The algorithm is iterated until n basis elements are selected. The choice of the number of basis elements depends on the performance level desired and is usually determined experimentally. If n is too small, then false-alarm rate is too high; if n is too large, the filter will not generalize to features outside the training set.

The output from the matching pursuit filter design algorithm is an ordered list of n basis elements and a list of n coefficients. If the filter design algorithm generates k proto-features, the matching pursuit filter consists of the basis elements and the k coefficient lists. The location of the basis elements encodes the geometrical structure of the object. The centers of the basis elements are usually not aligned, whenever the feature is larger than the support of the basis element.

A detection filter consists of

- Ordered list of basis elements $(\mathbf{g_1}, \mathbf{g_2}, ..., \mathbf{g_m})$
- Their relative center points $a_1(u_1, u_2), a_2(u_1, u_2), ..., a_m(u_1, u_2)$

A matching pursuit filter detects a feature by scanning the feature detection filter across the image, which results in a response image, $\mathcal{F}$. The response at pixel (u_1, u_2) measures the similarity between the region centered at (u_1, u_2) and the proto-feature. The maxima in the response image above a threshold is reported as feature occurrence.

The algorithm computes the image coefficient vector $\mathbf{y}(u_1, u_2)$ by expanding the image about the pixel (u_1, u_2), and projects the image on the translated basis elements. Let $y_i(u_1, u_2)$ be the response for the shifted position of the i^{th} basis function.

$$y_i(u_1, u_2) = \langle R^i \mathcal{I}, g_i(. + u_1, . + g_2) \rangle \tag{2.42}$$

and

$$\mathbf{y}(u_1, u_2) = (y_1(u_1, u_2), ..., y_m(u_1, u_2)) \tag{2.43}$$

After the image coefficient vectors have been determined, the next step computes the response image. Let $\Omega = (\alpha_1, ..., \alpha_{m-1})$ be the cluster vector that represents the protofeature; then $\mathcal{F}(u_1, u_2) = d_\theta(\Omega, \mathbf{a}(u_1, u_2))$, where d_θ is the cosine of the angle between two vectors, i.e., the response is cosine of the angle between Ω and $\mathbf{a}(u_1, u_2)$. The last step searches this response image $\mathcal{F}(u_1, u_2)$ for feature occurrences, by using some appropriate threshold.

2.6.3. Face Recognition using Matching Pursuit Filters

In the detection problem, the matching pursuit filter design procedure selected a basis in which the coefficient vectors clustered, and only one coefficient vector Ω represented a class of objects. For the purpose of detection, Ω is simply compared to the image coefficient vectors. Obviously, for detection only one coefficient vector suffices. However, for the identification problem, that is in order to distinguish among all the people in the database, there is a different coefficient vector for each individual. Person l is represented by coefficient vector $\Omega^l = \alpha_0^l, ..., \alpha_{n-1}^l$. To measure the similarity between an unknown face and individual l, their coefficient vectors are compared.

A face centered at (u_1, u_2) is identified as person l if the distance between $a(u_1, u_2)$ and Ω^l is minimized.

The algorithm for selecting the i^{th} basis element in the identification case has a different cost function C_g, which is designed to reveal the differences among the features;

$$C_g(\mathbf{R^i t_1}, ..., \mathbf{R^i t_m}, \mathbf{\Omega_{i-1}}) = -\sum_k \max_{l \neq k} d_\theta(k,l) + \lambda \sum_k \|(\alpha_0^k, ..., \alpha_{i-2}^k, \langle R^{i-1} t_k, g \rangle)\| \quad (2.44)$$

selects the i^{th} basis function. The function $d_\theta(k,l)$ equals the cosine of the angle between $(\alpha_0^k, ..., \alpha_{i-2}^k, \langle \mathbf{R^{i-1} t_k}, \mathbf{g} \rangle)$ and $(\alpha_0^l, ..., \alpha_{i-2}^l, \langle \mathbf{R^{i-1} t_l}, \mathbf{g} \rangle)$. Obviously, the coefficient vector $(\alpha_0^k, ..., \alpha_{i-2}^k)$ represents person k after the $i-1$ iteration. If g were selected in iteration i, then $(\alpha_0^k, ..., \alpha_{i-2}^k, \langle R^{i-1} t_k, g \rangle)$ would represent person k after this iteration. The first term in 2.44 forces the coefficient vectors as disparate as possible, while the second term searches for sets of coefficient vectors with the largest average magnitude. The parameter λ sets the relative importance of these two terms. If the second term is not included, the filter becomes too sensitive to patterns in the background. For identification, the output from the matching pursuit filter design algorithm is a list of n basis elements and a coefficient vector for each person in the training set.

For identification, a response image for each person, $\mathcal{F}^k$ must be computed. The estimated identity of the person in the image is $\hat{k}$, which is found by a search for the maximum response over all the $\mathcal{F}^k$ images.

$$\mathcal{F}^{\hat{k}}(\hat{u}_1, \hat{u}_2) = \max_{k,(u_1,u_2)} \mathcal{F}^k(u_1, u_2) \quad (2.45)$$

where $(\hat{u}_1, \hat{u}_2)$ is the estimated center of the face in the image.

Recognition Performance: In [17], a portion from FERET database consisting of 311 individuals were used. 58 of these were used for training 5 facial features; interior of the face, eyes, top of the nose, bridge of the nose. 30 coefficients were generated for each of these objects. 95.4% correct recognition performance was reported for identification of faces, and 95.2% for the location and identification of the faces [17].

2.7. Comparative Summary for Face Recognition Algorithms

Each recognition algorithm has different properties, as given in TABLE 2.3. A commonality between these is that, they are sensitive to head size, lighting conditions, facial expressions, complex background etc. So they all need preprocessing stages before extracting features. However, while some of these algorithms are very sensitive to these distortions such as correlation, and template-matching algorithms, some others of them can handle these imperfections in the face image to some extent.

As it can be seen from TABLE 2.3,

- *correlation* is very sensitive to scale, lighting conditions, facial expressions, rotations. It is also computationally very complex, $O(Kn^2)$. Furthermore, it requires the storage of all faces in their raw formats.
- *Eigenfaces* are very sensitive to scale, but it can handle the rest to some extend. This method is easy to train and easy to update the training set. Also it is fast enough for real-time applications.
- *Fisherfaces* are sensitive to scale, but it is robust against variation in illumination, facial accessories and expressions. It is also fast as "Eigenfaces".
- *HMM* is robust against rotation, facial expressions and accessories. Major drawback of this approach is its computation complexity.
- *Matching Pursuit Filter* is robust against translations of faces in the scene, but sensitive to the rest. It is not complex as HMM but requires higher recognition time than "Eigenfaces" and "Fisherfaces" methods.

Among these algorithm we choose the "Eigenfaces" and "Fisherfaces" methods to see how compression effects their performances. These algorithms are the most com-

TABLE 2.3. Comparison of the sensitivities and complexity of face recognition algorithms

	Scale	Nonuniform Lighting	Rotation	Facial Expressions	Facial Accessories
Correlation	High	High	High	Medium	High
Eigenfaces	High	Medium	High	Medium	Low
View-based	High	Medium	Low	Medium	Low
Feature-based	High	Medium	High	Medium	Low
Fisherfaces	High	Low	High	Low	Low
HMM	Low	Medium	Medium	Low	Low
Matching Pursuit Filters	Low	Low	High	Low	Low

monly used algorithms in the face processing (and also in pattern recognition) problems. They have been experimented on large databases [11, 12, 13] with high performance. So they are easy to train, robust enough against some of the imperfections, not complex, and results can be compared with results of other studies.

As it can be seen from TABLE 2.4, every face recognition algorithm is tested using different sizes of face databases. Among these recognition algorithms, view-based approach is tested on the largest face database [11] consisting more than 3000 individuals with 7562 images. The highest recognition performance is achieved by Fisherface-based method reported as 99.4% on a small face database.

Correlation, HMM and Matching Pursuit Filters are computationally complex methods, whereas eigenface and Fisherface methods are less complex.

TABLE 2.4. Comparison of face recognition methods

Method	Training Set	Testing Set	Recognition Results	Type of Database
Correlation [25]	NA	NA	over 96%	Frontal faces, small variations in illumination, scale
Eigenfaces [10]	16	2500	96%	Lighting Variations
			85%	Orientation Variations
			64%	Variations in Scale
Eigenfaces view-based approach [11]	128	7562	83%	FERET Database
Eigen Features [11]	45	NA	95%	Variations in head orientation and shifting
Fisherfaces [13]	16	16	99.4%	Strong variations in lighting and facial expressions
HMM[14]	24	24	84%	Variations in facial expression
Matching Pursuit Filters [17]	58	253	99.5%	Portion of FERET Database

3. COMPRESSION ALGORITHMS

Although there is a plethora of lossy compression algorithms, we chose three compression algorithms for comparison purposes in this work. The selected compression schemes are "Vector Quantization", JPEG and SPIHT algorithms, which hopefully forms a good representative set for the rest of compression algorithms.

3.1. Vector Quantization

Vector quantization (VQ) is a commonly used compression algorithm in clustering, compression, palette design, detection etc. It can incorporate certain low-level image processing tasks like interpolation and the coding can be coupled with classification [30, 31].

In VQ, initially a codebook, which is needed both for encoding and decoding is generated. To get a codebook, we used 40 faces (one from each individual in the face database) as training data. For training, we used the common block size 4×4, so there were $40 \times (92 \times 112)/(4 \times 4) = 25,760$ blocks available. The codebook is initiated using the splitting method, and then generated using the LBG algorithm [32].

Codebooks with different sizes are generated to enable compression at different rates. The images in the face database are encoded and decoded using these codebooks, as shown in FIGURE 3.1.

Vector quantization causes unpleasant blocking effects and stair-casing of edges in the encoded images, especially at very low bit rates.

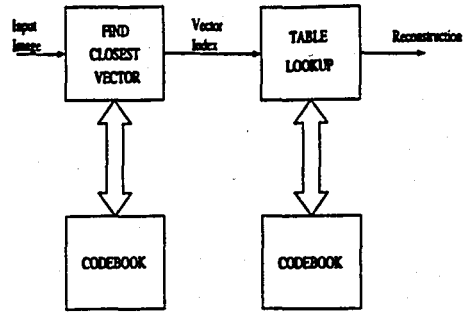

FIGURE 3.1. Overview of vector quantization

3.2. JPEG

We selected the JPEG scheme, because it is one of the most widely known standards for lossy image compression. Despite the increasing number of its close competitors, it is still used as the industry standard. The approach recommended by JPEG is a transform coding approach using the DCT [33].

The input image is divided into 8×8 blocks. These blocks are "level shifted" by 128 (for 8 bit greyscale images) and transformed using forward DCT. If the image size is not a multiple of 8, the last column or row is repeated until the nearest multiple of 8.

The JPEG algorithm uses uniform midtread quantization to quantize the various coefficients. The quantizer step sizes are organized in a table called quantization table. The recommended quantization table is shown in TABLE 3.1

In the quantization table, it is seen that the step size increases from the DC coefficients to higher-frequency coefficients. Because the quantization error is an increasing function of the step size, more quantization error will be introduced in the higher-frequency coefficients than in lower-frequency coefficients. The decision on the relative size of step sizes is based on how errors in these coefficients will be perceived by the

TABLE 3.1. Recommended luminance quantization table for JPEG

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

human visual system. Different coefficients in the transform have widely different perceptual importance. Quantization errors in the DC and lower AC coefficients are more easily detectable than quantization errors in the higher AC coefficients. Therefore larger step sizes for perceptually less important coefficients is used.

Because the quantizers are all midtread quantizers, the quantization process also functions as the thresholding operation. All coefficients with magnitudes less than half the corresponding step size will be set to zero. Because the step size at the tail end of the zigzag scan are larger, this increases the probability of finding a long run of zeros at the end of the scan. The entire run of zeros at the tail end of the scan can be coded with a special code after the last nonzero label, resulting in substantial compression.

Furthermore, this effect also provides us with a method to vary the compression rate. By making the step sizes larger we can reduce the number of nonzero values, and get a lower bit rate.

At the output of the quantizer, the coefficients are entropy coded (generally Huffman or arithmetic coding).

The JPEG scheme has also the blocking problem at the low bit rates as vector quantization scheme.

3.3. JPEG-2000

This novel compression scheme is intended to create a new image coding system that for different types of still images (bi-level, gray-level, color, multi-component) with different characteristics (natural images, scientific, medical, remote sensing imagery, text, rendered graphics, etc.). This coding system also provides low bit-rate operation with rate-distortion and subjective image quality performance superior to existing standards, without sacrificing performance at other points in the rate-distortion spectrum.

Although JPEG-2000 standards are not completely determined, its advantages over current JPEG compression scheme can be listed as:

- *Low bit-rate compression performance*: Current standards, such as standard JPEG, offer excellent rate-distortion performance in the mid and high bit-rates. However, at low bit-rates (e.g., below 0.25 bpp for highly detailed gray-level images) the distortion, especially when judged subjectively, becomes unacceptable.
- *Lossless and lossy compression*: There is no current standard that can provide superior lossless compression and lossy compression in a single codestream.
- *Large images*: Currently, the JPEG image compression algorithm does not allow for images greater then 64K by 64K without tiling.
- *Single decompression architecture*: The current JPEG standard has 44 modes, many of which are application specific and not used by the majority of the JPEG decoders. Greater interchange between applications can be achieved if a single common decompression architecture encompasses these features.

- *Transmission in noisy environments*: The current JPEG standard has provision for restart intervals, but image quality suffers dramatically when bit errors are encountered.
- *Computer generated imagery*: The current standard was optimized for natural imagery and does not perform well on computer generated imagery.
- *Compound documents*: Currently, JPEG is seldom used in the compression of compound documents because of its poor performance when applied to bi-level (text) imagery.

We used "JPEG-2000 Verification Model 4.0" to compress the faces in the database. It is chosen because of its prominent features and the high possibility of becoming the future standard in image compression.

The wavelet-based coding technique used in JPEG-2000, the severe blocking effects at the very low bit rates are prevented.

3.4. Set Partitioning in Hierarchical Trees

SPIHT is a rather new compression scheme, which use embedded and zero three coding techniques [34]. It deserves special attention because of its superior properties, which can be listed as

- *Image quality*: Extensive research has shown that the images obtained with wavelet-based methods yield very good visual quality. At first, it was shown that even simple coding methods produced good results when combined with wavelets. SPIHT belongs to the next generation of wavelet encoders, employing more sophisticated coding. In fact, SPIHT exploits the properties of the wavelet-transformed images to increase its efficiency.

- *Progressive image transmission*: SPIHT was designed for optimal progressive transmission. It achieves this optimality by producing a fully embedded coded file, in manner that at any moment the quality of the displayed image is the best available for the number of bits received up to that moment.

- *Optimized Embedded Coding*: In embedded coding, two encoded streams of size M and N ($M > N$) produced by the encoder from the same data, first N bits of both streams are same. For example, an image will be compressed at 3 different qualities for three remote users, and the minimum encoded streams will be 8 Kb, 30 Kb, and 80 Kb. If a non-embedded encoder like JPEG is used, three different files must be prepared for each image quality. On the other hand, if an embedded encoder is used a single 80 Kb stream is generated and the first 8 Kb of this stream correspond to the encoded image at lowest quality and the first 30 Kb corresponds to encoded image at higher quality.

- *Compression Algorithm*: Improvements in compression techniques result in more complex methods. However, SPIHT can achieve superior results using a simple uniform scalar quantization method.

- *Encoding/Decoding Speed*: A straightforward consequence of the compression simplicity is the greater coding/decoding speed. The SPIHT algorithm requires approximate time amount for encoding and decoding, whereas complex methods tend to have encoding times much more than decoding times.

- *Rate Control*: Image compression schemes do not provide strict control on rate and distortion. For example, with a specified target rate, some of these compression schemes try to generate an approximate rate. The embedded coding property of SPIHT allows exact bit rate control, without any penalty in performance (ie. no bit wasting with padding). The same property also allows exact mean-squared-error (MSE) distortion control.

- *Error Protection*: It is much easier to design error-resilient schemes for SPIHT. The information in SPIHT is sorted according to its importance and the requirement for powerful error correction codes decreases from the beginning to the end of the encoded stream.

Besides embedded coding, zero tree structures are used for efficient representation of the significance map. They are based on the following hypothesis: If a wavelet coefficient at a coarse scale is insignificant with respect to a given threshold T, then all of the coefficients of the same orientation in the same spatial location at finer scales are very likely to be insignificant with respect to T, shown in FIGURE 3.2. A zerotree root is encoded with a special symbol indicating that the whole tree is insignificant [35].

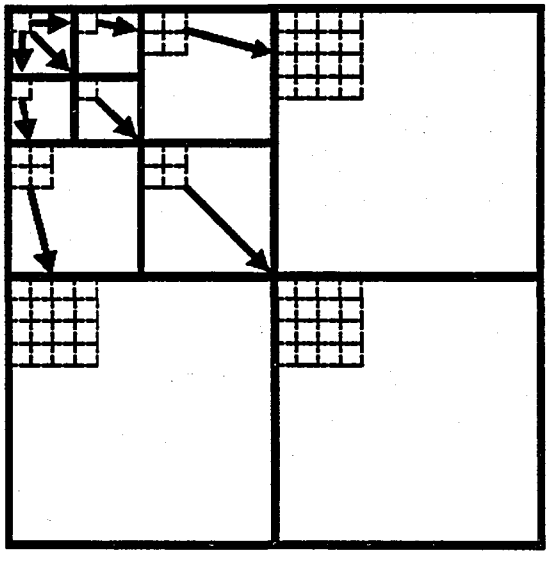

FIGURE 3.2. Zero tree data structures

SPIHT has more efficient significance map coding than EZW because of the set partitioning algorithm (rule for partitioning sets of trees and coefficients shared by encoder and decoder).

We use SPIHT scheme as it is one of the best representatives of the wavelet-based compression schemes.

Although its superior quality properties, SPIHT scheme causes blurring effect in the images at very low bit rate coding.

4. EXPERIMENTS WITH COMPRESSED FACE IMAGES

In order to obtain consistent results, which can be compared with other papers, we select a well-known face database. This database is the *Oracle Face Database* [24], which is commonly used in the face processing related papers in the literature. This database consists a set of images of 40 people with 10 different views, taken between April 1992 and April 1994 at the Olivetti Research Laboratory in Cambridge, UK. The size of images is 92x112, 8 bit grey levels. Some of the face images are taken at different times with varying lighting conditions, facial expressions (open/closed eyes/non-smiling) and facial details (glasses/no-glasses). Images have a dark homogeneous background, and they are in up-right, nearly frontal position. We choose this database because of its desirable face images and the adequate number of views per person. Our primary focus in this work is determining the effects of compression on recognition performance, therefore the following problems were not addressed:

- Detection and segmentation of faces from the image.
- Normalization of head size
- Handling large variations of lighting and head orientations

Therefore, we select the "Oracle Face Database" as the most appropriate database for our problem. We also set a procedure, stated below, in each experiment for the consistency and comparability of the results.

- 15 feature elements are used to represent a face,
- 5 views for each of the 40 face images are used for training,

- The remaining 5 views for each person are used for testing the algorithm. Therefore, the test set and training set are entirely different.

4.1. Correlation

The effects of compression on the autocorrelation of a face image can be seen in FIGURE 4.1. The effects of SPIHT scheme on the autocorrelation of a face image is less than the effects of other schemes. JPEG-2000 and JPEG schemes affect autocorrelation more than SPIHt and less than VQ scheme.

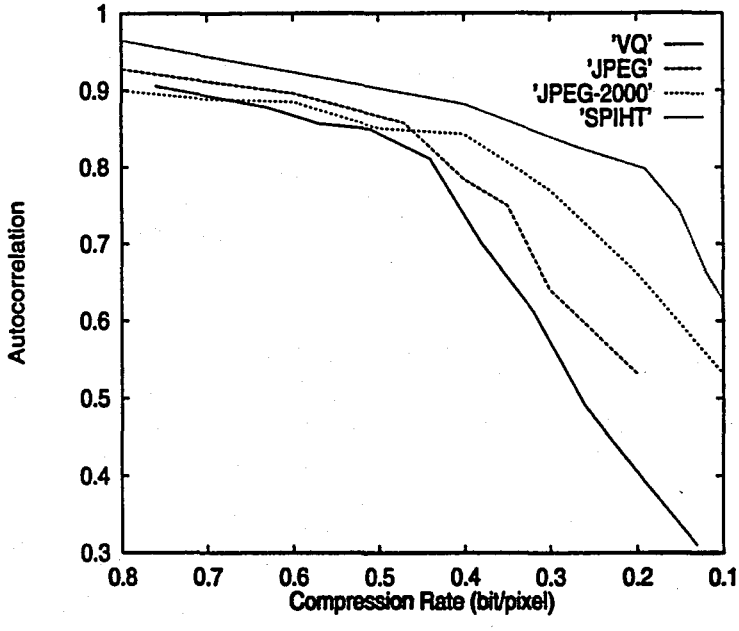

FIGURE 4.1. Effects of compression on the autocorrelation of a face image

The correct recognition performance of face images using correlation of whole faces can be seen in FIGURE 4.2. Face images compressed with all schemes can be recognized within an acceptable recognition rate down to 0.4 bit/pixel. Below this rate, recognition performance of face images compressed with VQ, JPEG and JPEG-2000 schemes degrades. The performance curve obtained from SPIHT scheme is still acceptable. The recognition performance of compressed faces with SPIHT breaks at 0.2 bit/pixel rate.

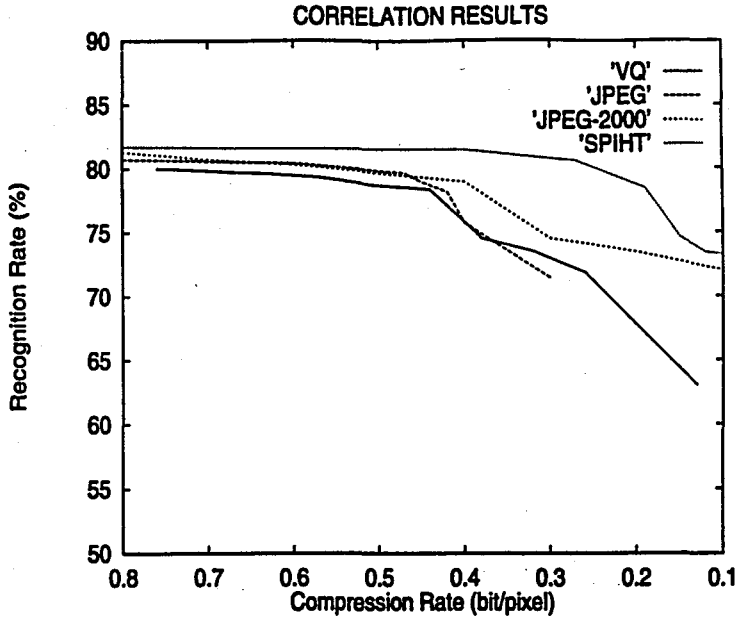

FIGURE 4.2. Effects of compression on recognition rate of correlation method

4.2. Database of Original Images

In the first set of experiments, we generate the training feature space using the original images, as shown in FIGURE 4.3. Test images are compressed, decompressed and their features are extracted. Thus, we employ the classification algorithm using the feature vectors of the compressed face images and the feature space generated by the non-compressed images. We call this the "Uncompressed Training" approach. This scheme is shown in FIGURE 4.4.

It had been shown that using more than 15-20 feature coefficients for "Eigenfaces" method does not improve the system performance by a large amount[25]. This could also be seen from FIGURE 2.12 that after only 4 feature coefficients the performance curve reaches its peak and additional features improves the performance only slightly. FIGURE 4.5 depicts that first 15 principal components are relatively more important than the higher components. Therefore, we use 15 feature coefficients for the representation of the faces for "eigenfaces" method and for other recognition algorithms, we do

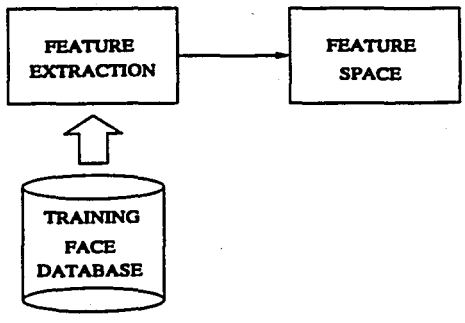

FIGURE 4.3. Generation of the face space

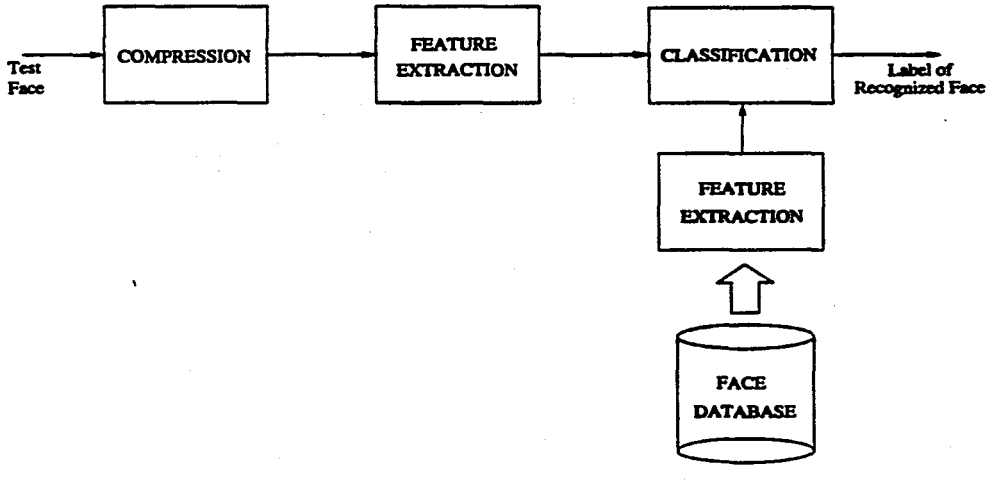

FIGURE 4.4. Scheme overview of "uncompressed training" approach

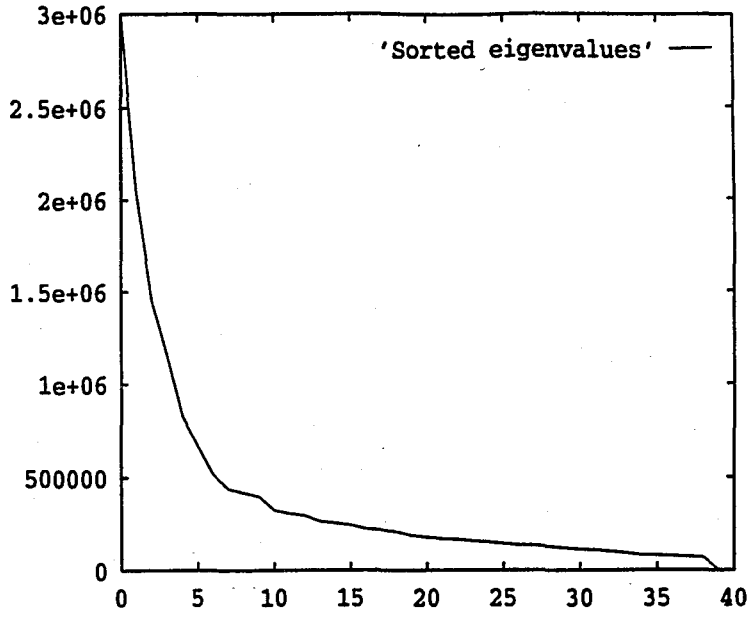

FIGURE 4.5. Sorted eigenvalues corresponding to principal components

not change the number of feature coefficients.

In order to visualize the effects of compression on the "Eigenfaces" method, we plotted the scatter of the face class centers on the first two eigenfeature vector before the compression and after the compression. From FIGURE 4.6 and FIGURE 4.7, we can see that the scatter of the face class centers on the first two components shrinks under the compression using "Vector Quantization" at 0.4 bit/pixel. This implies that, the power to discriminate the faces decreases with loss of information due to compression.

4.3. Compression with Vector Quantization

Using "Vector Quantization" for compression, we see that the face images can be recognized down to 0.4 bit/pixel, a 20:1 compression ratio without any noticeable deterioration. Also it is seen that the performance of "Fisherfaces" method is higher

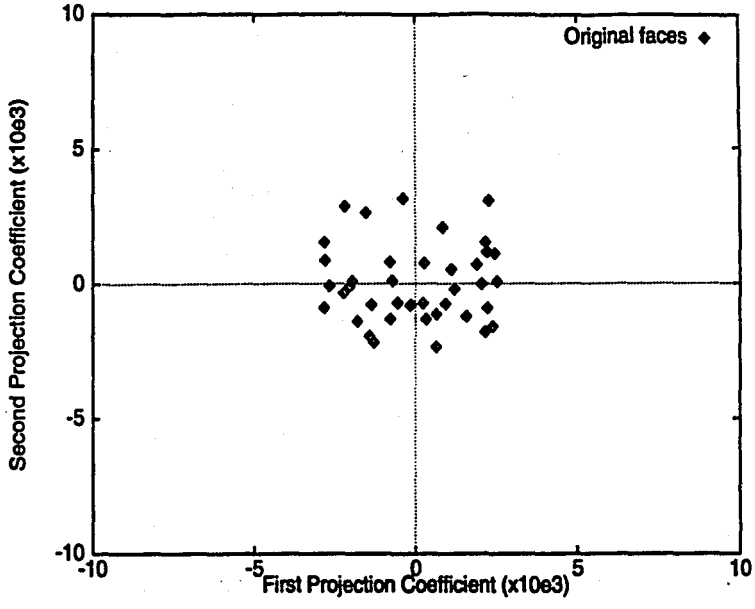

FIGURE 4.6. Scatter of face class centers on first two principal components without compression

than "Eigenfaces" method by about 2-4% all throughout and down to 0.4 bit/pixel rate. This is to be expected, since the "Fisherfaces" method is a class-specific method and optimum for recognition, when more than one view per person is available.

It can be noticed that, after 0.4 bit/pixel rate, a break point occurs in the correct recognition rate in the eigenface method while the decrease in performance of the Fisherface method is simply accelerated. From FIGURE 4.8, it is seen that the breakpoint of "Fisherfaces" method is not as sharp as "Eigenfaces" method. From this figure we also see that the performance of "Fisherfaces" method resembles the SNR curve, whereas the curve of "Eigenfaces" method is parallel to both curves down to 0.4 bit/pixel rate, but abruptly drops down below this rate.

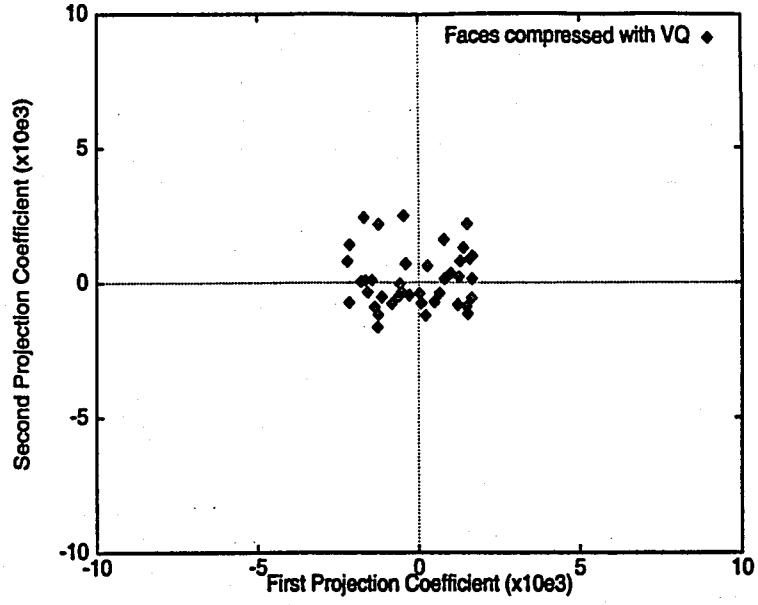

FIGURE 4.7. Scatter of face class centers on first two principal components after compression with vector quantization at 0.4 bit/pixel

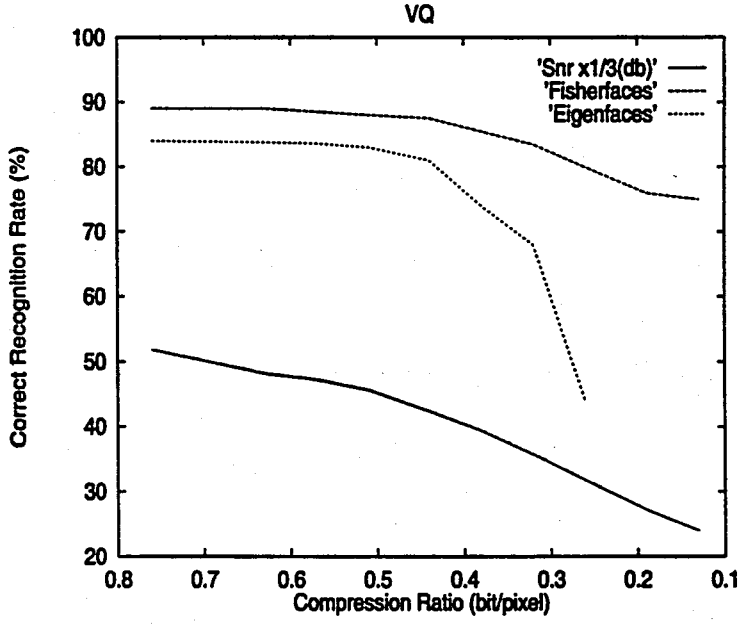

FIGURE 4.8. Effects of vector quantization on "Eigenfaces" and "Fisherfaces" techniques

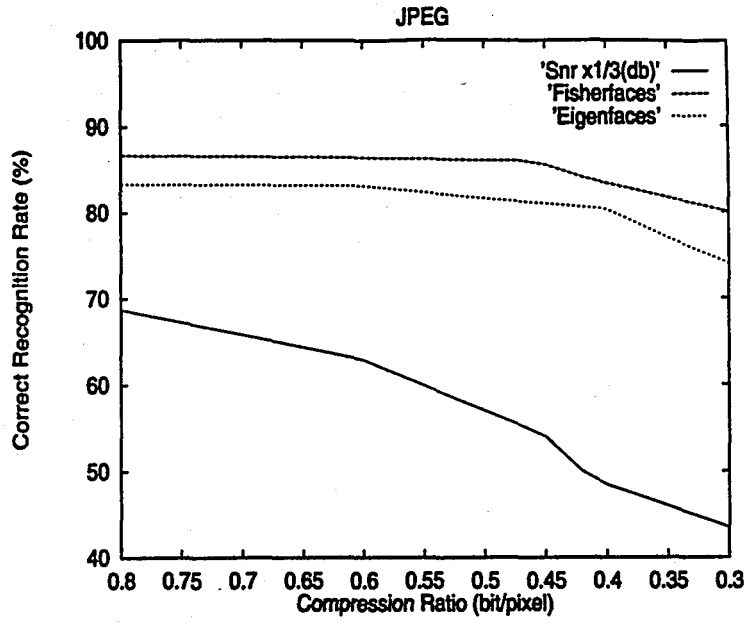

FIGURE 4.9. Effects of JPEG on "Eigenfaces" and "Fisherfaces" techniques

4.4. Compression with JPEG

From FIGURE 4.9, it is seen that the results of the JPEG compression scheme is similar to the VQ scheme. Faces can also be compressed with JPEG down to 0.4 bit/pixel (20:1) with both recognition algorithms, beyond which the drop is accelerated. Again "Fisherfaces" method gives 1-3% better results than "Eigenfaces" method. In contrast to VQ results, the performance curve of "Eigenfaces" algorithm is parallel to that of Fisherfaces algorithm. It can easily be noticed that the recognition rate curves is not parallel to SNR curve at every rate of compression.

After 0.4 bit/pixel rate, the correct recognition rate of both recognition algorithm decrease. Both VQ and JPEG schemes are convenient for compressing the human faces without a performance loss of recognition down to 0.4 bit/pixel. Below this rate, these schemes deteriorate the performance of "Eigenfaces" and "Fisherfaces" methods.

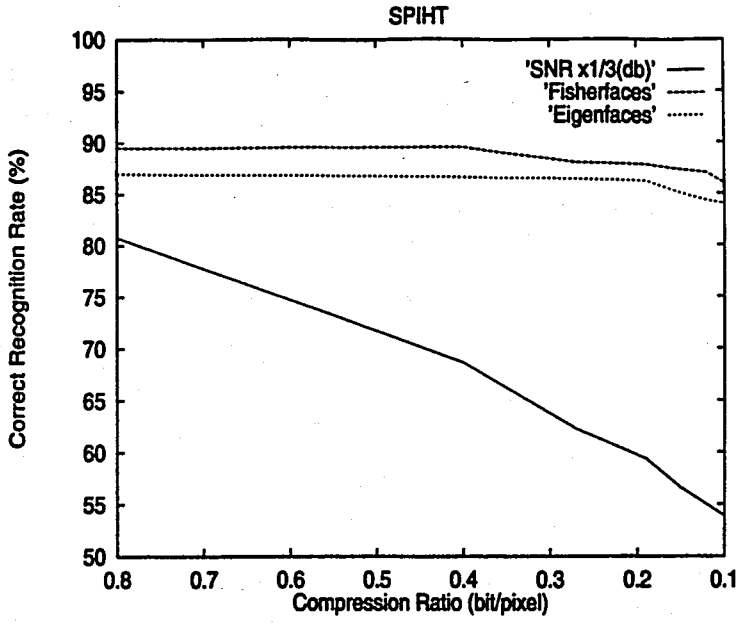

FIGURE 4.10. Effects of wavelet based SPITH compression on "Eigenface" and "Fisherface" techniques

4.5. Compression with SPIHT

From FIGURE 4.10, the SPITH algorithm, which is better than other schemes in terms of SNR, yields also better results in recognition performance as compared to JPEG and VQ. Both "Eigenfaces" and "Fisherfaces" method can recognize the faces without degradation of performance down to 0.2 bit/pixel (40:1). Although the SNR of the image decrease, the recognition performance stays nearly constant down to 0.2 bit/pixel and begins to degrade only after this rate.

4.6. Compression with JPEG-2000

Although more efficient coding methods have been proposed for JPEG-2000, both "Eigenface" and "Fisherface" methods can perform down to 0.4 bit/pixel without a

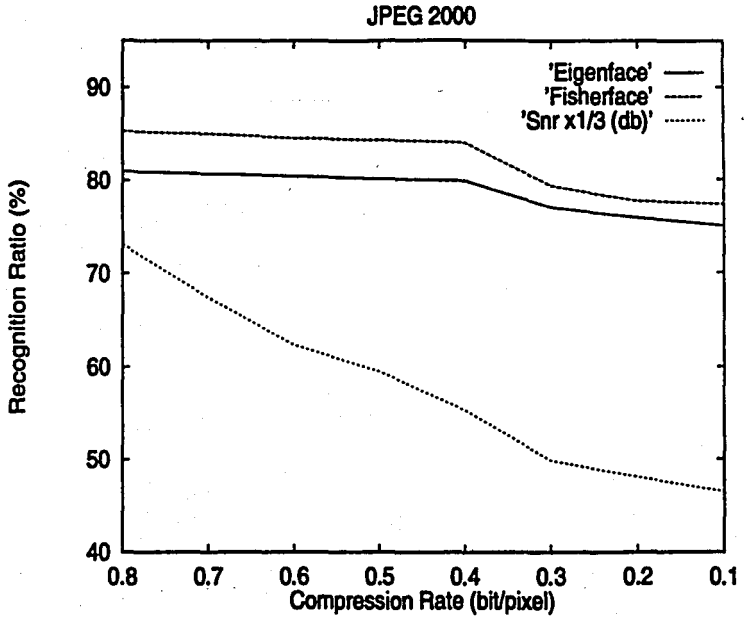

FIGURE 4.11. Effects of JPEG-2000 compression on "Eigenface" and "Fisherface" techniques

performance loss, as shown in FIGURE 4.11. "Fisherface" method is again performed 1-3% better than "Eigenface" method.

4.7. Database of Compressed Images

Generation of the face space using the compressed faces is an alternative approach. We call this the "Compressed Training" approach. In this case as shown in FIGURE 4.12, the face space is generated from the features of compressed training samples at 0.4 bit/pixel and from the features of compressed training samples from different bit rates. Recall that in contrast in FIGURE 4.3 original images were used for the generation of the feature space.

We tried this approach for the "Eigenfaces" method. For Vector Quantization scheme, we see that this approach give 1-2% better performance down to 0.4 bit/pixel.

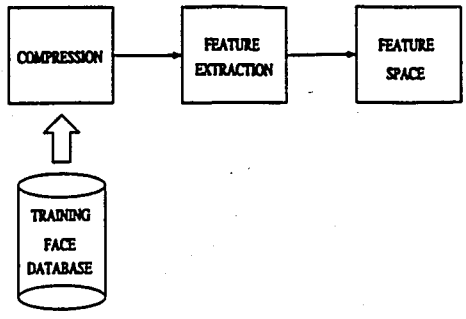

FIGURE 4.12. Generation of the face space from compressed face images

As shown in FIGURE 4.13, below this rate it prevents the abrupt performance drop as compared to the uncompressed training. The eigenfaces generated from compressed face at different bit rates performs better than compressed faces at 0.4 bit/pixel.

"Eigenfaces" generated from the compressed training images also give better results for JPEG scheme. It can be seen from FIGURE 4.14 that "Eigenfaces" generated from original training face images perform 1-2% worse than the "Eigenfaces" generated from the compressed training face images at 0.4 bit/pixel and at different bit rates. Despite the improvement in the recognition performance, this approach do not contribute to the lower bound of compression.

As it can be seen from FIGURE 4.15, eigenfaces generated from compressed face images at 0.4 bit/pixel rate and at different bit rates performs 1-4% better than eigenfaces generated using original face images for JPEG-2000 compression scheme. The breaking point in the performance curve does not change.

For the wavelet based SPIHT scheme, this approach gives worse result than the "Eigenfaces" generated from original face images, as shown in FIGURE 4.16.

Finally, to grasp the effects of compression schemes on the facial features, the "DFFS" metric is used. The "Distance From Face Space" is computed for the images

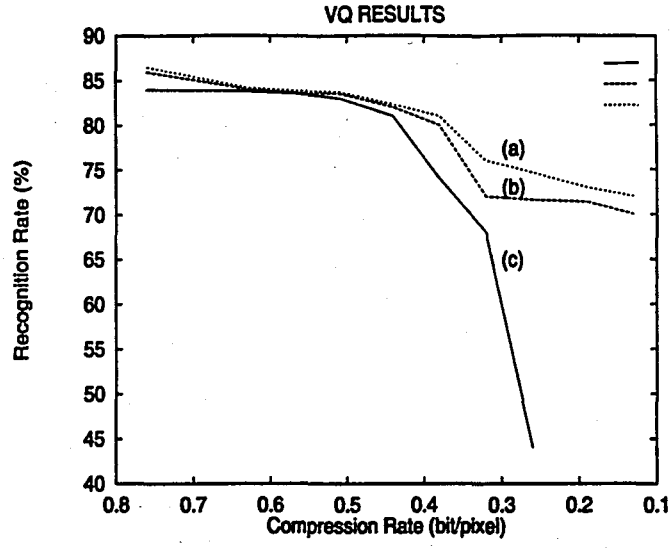

FIGURE 4.13. Comparison of eigenfaces generated from original and compressed faces for VQ. Eigenfaces generated from (a) Compressed face images at different bit rates. (b) Compressed face images at 0.4 bit/pixel. (c) Original face images

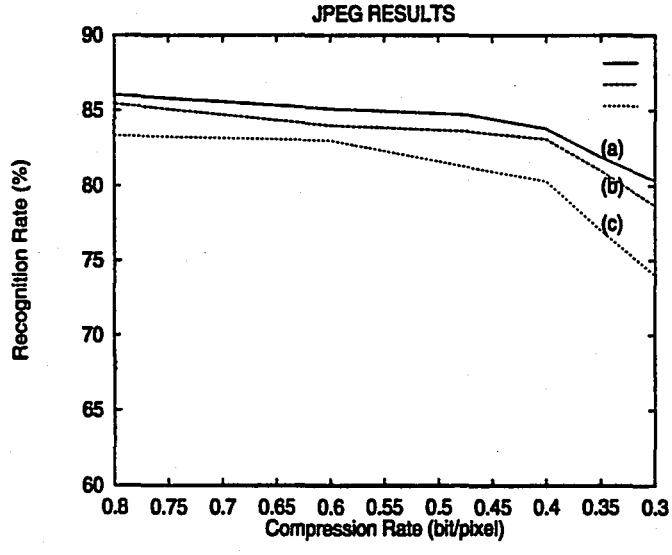

FIGURE 4.14. Comparison of eigenfaces generated from original and compressed faces for JPEG. Eigenfaces generated from (a) Compressed face images at different bit rates. (b) Compressed face images at 0.4 bit/pixel. (c) Original face images

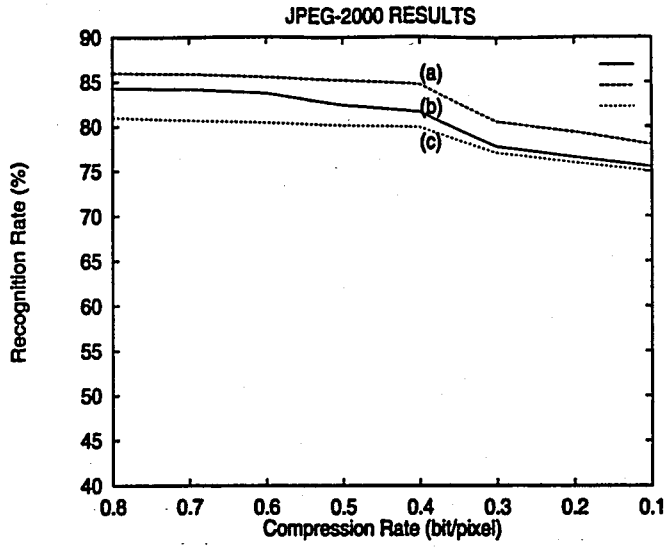

FIGURE 4.15. Comparison of eigenfaces generated from original and compressed faces for JPEG-2000. Eigenfaces generated from (a) Ccmpressed face images at different bit rates. (b) Compressed face images at 0.4 bit/pixel. (c) Original face images

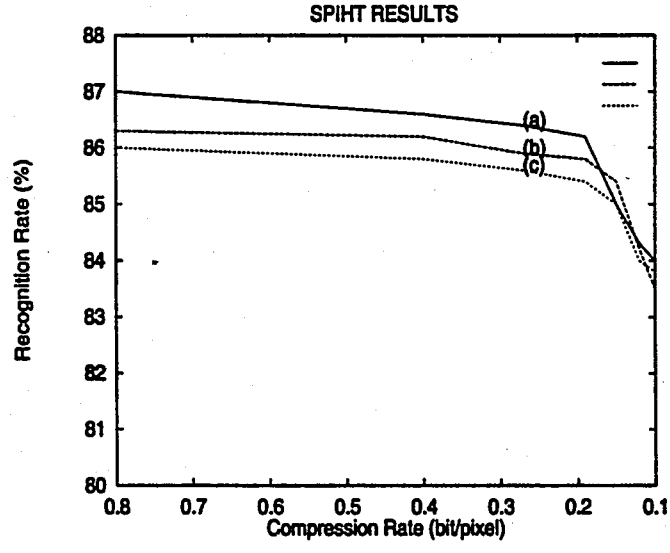

FIGURE 4.16. Comparison of eigenfaces generated from original and compressed faces for SPITH. Eigenfaces generated from (a) Original face images. (b) Compressed face images at different bit rates. (c) Compressed face images at 0.4 bit/pixel.

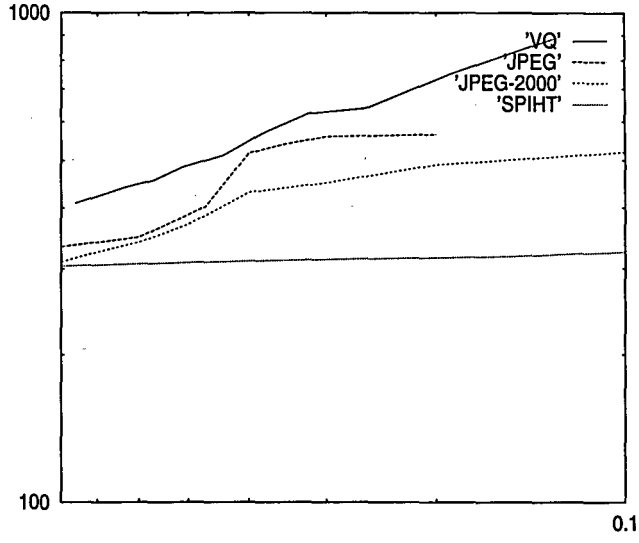

FIGURE 4.17. Distance from face space for compression schemes

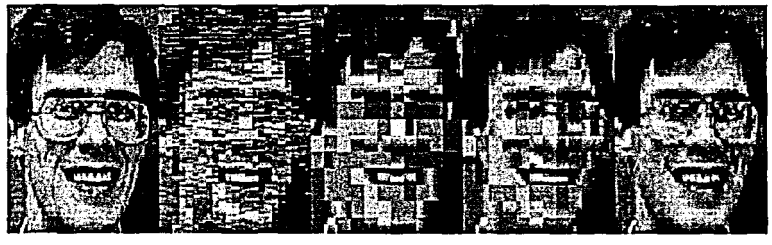

FIGURE 4.18. Original face and faces compressed at 0.4 bit/pixel rate with VQ, JPEG and SPIHT respectively

compressed at various bit rates. Recall that the "DFFS" is a metric, and it shows the faceness of an image. It can be used to measure how well the features of a face are preserved. From FIGURE 4.17, DFFS of the faces compressed with VQ is higher than that of others. DFFS of the faces, compressed with JPEG is less than that of VQ but higher than the one for SPIHT and JPEG-2000 algorithm. This measure shows that SPIHT algorithm preserves the features of a face better than other schemes. From FIGURE 4.18, it can be seen that the features of the faces compressed with VQ, JPEG and JPEG-2000 at 0.4 bit/pixel rate are damaged by the blocking effect, whereas SPIHT can preserve the features better than other schemes.

5. CONCLUSIONS AND FUTURE RESEARCH

With the enlarging human face databases, face recognition in the compressed domain and effects of compression on the performance of existing algorithms are becoming important problems. In this work, we try to determine the effects of well-known compression schemes on widely used face recognition algorithms.

We choose vector quantization, JPEG, JPEG-2000 and wavelet based SPIHT schemes for their popularity and their prominent features. Among these VQ is a widely used low-level image processing tool and JPEG is one of the industrial standard. JPEG-2000 is a novel compression method (which is still being developed) with superior coding properties, and seems to be the future standard for image compression. The SPIHT algorithm is a rather new scheme based on the wavelets with embedded zero-tree coding. It is the closest competitor of the JPEG-2000.

Among a number of face recognition proposed algorithms, we select the most commonly used ones. The correlation, "Eigenface" and "Fisherface" methods are used in this work. These methods are widely used in face recognition tasks and they are quite robust against variations in the head orientation and illumination variations. They are easy to train and has an easy update rule during training.

The segmentation of the face from the image and normalization of the face is not addressed in this work. Therefore, we select "Oracle Face Database" consisting of 40 individuals with 10 different views, which do not contain wide variations in the illumination, head orientation and head scale.

We tried these algorithms on compressed and decompressed faces at different bit rates to determine to what extend the faces can be compressed without a major performance deterioration. We generate the feature space using original (non-compressed)

face images. We see that faces compressed with VQ, JPEG and SPIHT can recognize the faces with a little loss of performance down to 0.4 bit/pixel (20:1) rate. But at lower bit rates than 0.4 bit/pixel, the performance of the face recognition algorithms begin to degrade with VQ and JPEG schemes. Using the SPIHT algorithm faces can be compressed below this rate down to 0.2 bit/pixel (40:1). Below this rate the performance of this method also degrades.

For these compression schemes, "Fisherfaces" method give 2-5% better results than "Eigenfaces" method as we expected. The difference between the performance of these two schemes is lower than the reported difference in the literature. This is due to the near-normalized face database we use. Despite the improved performance of "Fisherfaces" method, it does not change the breaking point rate in the performance curves.

Since VQ and JPEG are block based compression schemes, they destroy the facial features at low bit rates. The blocking effect on the compressed faces can be seen visually. On the other hand, wavelet based compression schemes can preserve the facial features at lower bit rates. Only blurring occurs in the edges. The lower information loss in the SPIHT algorithm enables this scheme to outperform other three schemes.

To generate the feature space from the compressed face is another approach, we tried. Instead of running training algorithms on original images, we use compressed faces at 0.4 bit/pixel rate and at different bit rates as training samples, ie., we generate eigenfaces from the faces compressed with VQ for testing the VQ compressed faces. This approach worked well in VQ and JPEG compression where information loss is high. Even more in VQ scheme, it prevents the abrupt performance drop below 0.2 bit/pixel. At high bit rates, this approach increased the performance 1-2% for VQ and JPEG. In the lower rates, the performance gain for VQ is higher. For SPIHT scheme this approach give 1% worse results than feature space generated by original faces. Below 0.2 bit/pixel the performance difference is even less.

The faces can be compressed down to 0.4 bit/pixel with VQ, JPEG and SPIHT compression schemes with a little loss of performance. Below this bit rate VQ and JPEG cannot preserve the discriminative information of the faces. For lower bit rates (lower than 0.4bit/pixel) it is convenient to use SPIHT algorithm. Face images must not be compressed below 0.2 bit/pixel rates with these compression schemes, which are optimized to minimize the reconstruction error. To compress the faces at much lower bit rates, ie. 0.1-0.01 bit/pixel, a specific compression method for faces must be used. For example, the "Eigenfaces" method can represent a face with 10-30 coefficients without loss of performance, which means 0.1-0.3 bit/pixel.

Finally, we conclude that face images can be compressed to 100:1 ratio using face-specific compression methods, 40:1 using SPIHT method and 20:1 using VQ, JPEG and JPEG-2000 methods, without a major deterioration in recognition performance.

The enlargement of the face database will move the breaking point in the recognition performance curves to a slightly higher bit rate compression.

5.1. Future Research

Compression of the faces is the only way to cope with storage and bandwidth limitations. Therefore, research for the recognition with compressed face images will be active in the following years, with increasing attention. Some of the interesting problems, which may be addressed include;

- Robust recognition algorithms, working on the compressed domain
- The optimization of compression algorithms for pattern recognition problem
- Manipulation of classification algorithms to perform better on compressed patterns

- Development of more powerful feature extraction algorithms, which selects the features that do not change much with compression.

The research on these problems requires establishments of an extensive face database, for the compatibility of research results.

REFERENCES

[1] Daugman, J. G., "High Confidence Visual Recognition of Persons by a Test of Statistical Independence," *IEEE Trans. Pattern Recognition and Machine Intelligence*, Vol. 15, pp. 1148-1161, November 1993.

[2] Bigün, J., G. Chollet, G. Borgefors, *Audio and Video based Biometric Person Authentication*, Springer, 1997.

[3] Hong, L., A. Jain, "Integrating Faces and Fingerprints for Personal Identification," *IEEE Trans. on Pattern Recognition and Machine Intelligence*, Vol. 20, No. 12, pp. 1295-1307, December 1998.

[4] Kirby, M., and L. Sirovich, "Application of Karhunen - Loeve Procedure for the Characterization of Human Faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 12, pp 103-108, 1990.

[5] Kelly, M. D., "Visual Identification of People by Computer," Tech. Rep. AI Proj., Stanford, CA, 1970.

[6] Kanade, T., *Computer Recognition of Human Faces*, Basel and Stuttgart: Birkhauser, 1977.

[7] Hong, Z., "Algebraic Feature Extraction of Image for Recognition," *Pattern Recognition*, Vol. 24, pp. 211-219, 1991.

[8] Manjunath, B. S., R. Chellappa, and C. Malsburg, "A Feature based Approach to Face Recognition," *Proc. IEEE Computer Society Conference on computer Vision and Pattern Recognition*, pp. 373-378, 1992.

[9] Cheng, Y., K. Liu, J. Yang, Y. Zhang, and Y. Gu, "Human Face Recognition Method based on the Statistical Model of Small Sample Size," in *SPIE Prooeedings: Intelligent Robots and Computer Vision X: Alg. and Tech.*, Vol. 1607, pp. 85-95, 1991.

[10] Turk, M., and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, Vol 3, No. 1, pp 71-86, 1991.

[11] Pentland, A., B. Moghaddam, T. Starner, and M. Turk, "View-based and Modular Eigenspaces for Face Recognition," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp 84-91, 1994.

[12] Pentland, A., A. Starner, N. Etcoff, A. Masoiu, O. Oliyide, and M. Turk, "Experiments with Eigenfaces," *Looking at People Workshop*, IJCAI'93, Chamberry, France, August 1993.

[13] Peter, N., J. Hespanha, D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, August 1997.

[14] Samaria, F., F. Fallside, "Face Identification and Feature Extraction Using Hidden Markov Models," *Image Processing: Theory and Applications*, 1993.

[15] Samaria, F., F. Fallside, "Automated Face Identification Using Hidden Markov Models," *Proceedings of the International Conference on Advanced Mechatronics*, 1993.

[16] Tsapatsoulis, N., N. Doulamis, A. Doulamis, and S. Kollias, "Face Extraction from Non-Uniform Background and Recognition in the Compressed Domain," *Proceedings of IEEE Internation Conference on Acoustics, Speech and Signal Processing*, Seattle, pp. 2701-2704, May 1998.

[17] Phillips, J. P., "Matching Pursuit Filters Applied to Face Identification," *IEEE Transactions on Image Processing*, Vol. 7, No. 8, pp. 1150-1164, August 1998.

[18] Duc, B., S. Fischer, and J. Bigün, "Face Authentication with Gabor Information on Deformable Gabors," *IEEE Trans. on Image Processing*, Vol. 8, No. 4, pp. 504-516, April 1999.

[19] Freeman, W. T., E. Adelson, "The Design and Use of Steerable Filters," *IEEE Pattern Analysis and Machine Intelligence*, Vol. 13, No. 9, pp. 891-906, 1991.

[20] Wechsler, H., J. Phillips, V. Bruce, F. F. Soulié, T. S. Huang, *Face Recognition from Theory to Applications*, NATO ASI Series, Series F: Computer and Systems Sciences, Vol. 163, 1998.

[21] Zhao, W., N. Nandhakumar, "Linear Discriminant Analysis of MPF for Face Recognition," *Internation Conference on Acoustics and Signal Processing*, pp. 185-188, 1999.

[22] Kohonen, T., *Self Organization and Associative Memory*, Berlin:Springer, 1998.

[23] Seales, W. B., C. Yuan, W. Hu, M. D. Cutts, "Object Recognition in the Compressed Imagery," Image and Vision Computing, Vol. 16, pp. 337-352, 1998.

[24] Oracle Face Database, http://www.cam-orl.co.uk/facedatabase.html

[25] Brunelli, R., T. Poggio, "Face Recognition: Features versus Templates," *IEEE Transactions on Pattern Analysis ans Machine Intelligence*, Vol 15, No 10, pp 1042-1052, October 1993.

[26] Etemad, K., R. Chellappa, "Discriminant Analysis for Recognition of Human Face Images," *Jounal of Optical Society of America*, Vol. 14, No. 8,pp. 1724-1733, August 1998.

[27] Yale Face Database, http://cvc.yale.edu

[28] Rabiner, L., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Processing," *Proceedings of IEEE*, Vol. 77, pp. 257-286, February 1989.

[29] Mallat, S., and Z. Zhang, "Matching Pursuit with Time-frequency Dictionaries," *IEEE Trans. Signal Processing*, Vol. 41, pp. 3397-3415, 1993.

[30] Rabbani, M., P. W. Jones, *Digital Image Compression Techniques*, SPIE Optical Engineering Press, 1991.

[31] Sayood, K., *Intoduction to Data Compression*, Morgan Kaufman Publishers, 1996.

[32] Linde, Y., A. Buzo, R. M. Gray, " An Algorithm for Vector Quantization Design," *IEEE Trans. on Communications*, Vol. 28, pp. 84-95, January 1980.

[33] Wallace, G.K., "The JPEG Still Compression Standard," *IEEE Trans. on Consumer Electronics ...*

[34] Said, A., W. A. Pearlman, *"A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees"*, IEEE Transactions on Circuits and Systems for Video Technology, Vol 6, pp 243-250, June 1996.

[35] Shapiro, J.M., "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Transactions on Signal Processing*, Vol. 41, No. 12, pp. 3445-3462, December 1993.