ISTANBUL TECHNICAL UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY

FACE RECOGNITION

USING EIGENFACES

M. Sc. THESIS
January, 1996

ILKER ATALAY
Computer Engineer B.Sc.

Date of Submission : 15 January 1996

Date of Approval : 31 January 1996

Examination Committee

Supervisor : Associate Prof. Dr. Muhittin Gokmen
Member : Prof. Dr. Erdal Panayirci
Member : Associate Prof. Dr. Bulent Orencik

Acknowledgments

First of all, I would like to thank my advisor, Ass. Prof. Muhittin Gokmen, who has been very helpful through out the years, starting with his image processing and computer vision lectures, and ending up in the preparation of this thesis.

A great deal of thanks goes to my family, for their endless support and encouragement, even during the most demanding phases of this research.

And finally, I would like to thank YKB Technology., where I have done the beta tests of the proposed face recognition system, together with the preparation of this manuscript.

ILKER ATALAY

Istanbul, January 1996

Table of Contents

List of Figures

List of Tables

Abstract

In this thesis, a research was done to find out the different approaches to the face recognition problem. It has been observed that these different approaches fall into two major categories that are given below:

- **Feature based recognition**, which is based on the extraction of the properties of individual organs located on a face such as eyes, nose and mouth, as well as their relationships with each other. Feature vectors describing the characteristics of face images are evaluated by using deformable templates and active contour models, where excessive geometry and the minimization of energy functions are involved.

- **Principal component analysis,** based on information theory concepts, seek a computational model that best describes a face, by extracting the most relevant information contained in that face. Eigenfaces approach is a principal component analysis method, in which a small set of characteristic pictures are used to describe the variation between face images. Goal is to find out the eigenvectors (eigenfaces) of the covariance matrix of the distribution, spanned by a training set of face images. Later, every face image is represented by a linear combination of these eigenvectors. Evaluation of these eigenvectors are quite difficult for typical image sizes but, an approximation that is suitable for practical purposes is also presented. Recognition is performed by projecting a new image into the subspace spanned by the eigenfaces and then classifying the face by comparing its position in face space with the positions of known individuals.

A face recognition system, based on the eigenfaces approach is proposed. Eigenfaces approach seems to be an adequate method to be used in face recognition due to its simplicity, speed and learning capability. Experimental results are given to demonstrate the viability of the proposed face recognition method.

Özet

Özyüz Yöntemi İle Yüz Tanıma

Bu tezde, yüz tanıma problemi üzerinde araştırma yapılarak, kullanılan temel yöntemler incelenmiştir. Yüz tanımada kullanılan yöntemlerden biri olan, özyüz (eigenface) yöntemi ile bir yüz tanıma sistemi geliştirilerek, sistemin işlevselliği deneysel sonuçlar ile gözlemlenmiştir.

Yüz tanıma günümüzde önemini hızla arttırmakta olan bir uygulama konusu haline gelmektedir. Mevcut yüzlerden elde edilecek bir veri dabanı üzerinde otomatik olarak yapılacak tanımaların, suçluların teşhisi sırasında, emniyet görevlilerinin işini ne kadar kolaylaştıracağı açıktır.

Yüz tanıma problemi için bir bilgisayar modeli kurmak oldukça zordur. Bu zorluk, yüzlerin çok boyutlu, karmaşık ve anlamlı görsel objeler olmalarından kaynaklanmaktadır. Oysa insan beyni, bir an gördüğü bir yüzü yıllar sonra bile yaşlanma, saç uzaması, sakal bırakma gibi değişimlere rağmen, bir kaç saniye içerisinde hatırlayabilmektedir. Insan beynindeki bu üst düzey yetenek, yüz tanıma esnasında bilginin en iyi ve en kısa şekilde kodlandığı gerçeğini ortaya çıkarmaktadır.

Yapılan araştırma sonucu, bilgisayar yardımı ile yapılan yüz tanıma çalışmalarında iki temel yöntem izlendiği tespit edilmiştir. Bu yöntemlerden birincisi, yüz ürezinde yer alan, göz, ağız, burun gibi organların geometrik özelliklerinden elde edilen bilgiler ile bir özellik vektörü oluşturulmasına, tanıma işleminin de, benzer şekilde elde edilmiş vektörlerin karşılıştırılması ile yapılması ilkesine dayanmaktadır. Özellik vektörlerinin elde edilmesi sırasında aşağıda belirtilen üç temel ilişki dikkate alınmaktadır:

- **Birinci derece özellikler**. Bunlar, yüz üzerinde yer alan organların birbirlerinden bağımsız nicel özelliklerinden oluşur. Göz çukurlarının çevre uzunluğu, dudakların uzunlukları bu tür özelliklerdendir.

- **İkinci derece özellikler**. Birinci derece özelliklerin de kullanılarak, organların birbirleri ile ilişkilerinden elde edilen özelliklerdir. Gözlerin ve ağızın, burundan uzaklıkları gibi.

- **Yüksek dereceden özellikler**. Yaşlanma etkileri gibi genellikle nicel olmayan, yüzün bütününü ilgilendiren özelliklerdir. Elde edilmeleri çok zor olduğunda, tanıma işleminde genellikle kullanılmazlar.

Geometrik özelliklerin elde edilmesi sırasında, Yuille tarafından geliştirilmiş olan, esnek göz ve ağız şablonları ile, aktif hat (active contour) yöntemleri kullanılmaktadır. Şablonlar yüz üzerinde gezdirilerek, bazı enerji fonksiyonları minimize edilmeye çalışılır. Enerji fonksiyonu minumum değerini aldığında, göz yada ağız tam olarak saptamış demektir.

Yüz tanımada kullanılan ikinci temel yöntem ise, **temel bileşen analizi** (principal component analysis) adını ile anılmaktadır. Bilgi teorisi temellerinin esas alındığı bu yaklaşımlarda, yüzleri en az bilgi ve en iyi şekilde ifade etme yoluna gidilmektedir. Yüzler ifade edilirken, göz, ağız gibi organlar bağımsız olarak düşünülmemekte, elde edilen kodlamada, onlara ait bazı ayırt edici özellikler de yer alabilmektedir.

Bu tezde geliştirilen yüz tanıma sistemi de, temel bileşen analizi yöntemlerinden biri olan, **özyüz** yöntemine dayanmaktadır. Kullanılan yöntem, yüz tanıma problemine iki boyutlu bir tanıma problemi olarak yaklaşmakta ve yüzler arasındaki farklılıkları en iyi ortaya çıkaran bir yüz uzayının (face space) özvektörlerinin, yani özyüzlerin elde edilerek,diğer yüzlerin, bu özvektörlerin bir lineer kombinasyonu ile ifade edilmesi ilkesine dayanmaktadır.

Özyüzler, eğitim kümesinde yer alan yüzlerin temel bileşenlerini (principal components) oluşturmaktadır. Tanıma olayı yeni bir yüzün, elde edilen bu özyüzler tarafından gerilen yüz uzayına projeksiyonu ile gerçeklenmektedir. Projeksiyon sonucu elde edilen konum, sisteme tanıtılmış yüzlerin bu yüz uzayındaki konumları ile karşılaştırılır. Yeteri kadar yakın olan bireyler varsa, bu, yeni yüz sisteme daha önce tanıtılmış demektir. Aksi halde, istenirse bu yeni yüzün, yüz uzayındaki konumu saklanarak, sistemin yeni yüzü öğrenmesi sağlanabilir.

Geliştirilen yüz tanıma sisteminde özyüz yaklaşımının tercih edilme nedeni; yöntemin yüzün geometrisinden bağımsız olması, gerçeklenmesinin kolay olması, özel donanım kullanılmadan bile gerçek zamanda çalışabilmesi ve sistemin yeni yüzleri tanır hale gelmesinin, yani tanımayı öğrenmesinin diğer yöntemlere göre çok daha kolay ve hızlı olmasıdır.

Genel olarak, gürbüz (robust) bir yüz tanıma sistemi, kendisine daha önce öğretilmiş bir yüzü, aşağıda belirtilen koşullar altında bile doğru tanıyabilmelidir:

- Işık kaynağının şiddetinin ve konumunun değişmesi.
- Kafa pozisyonunun ve büyüklüğünün değişmesi.
- Arka planının (face background) değişmesi.
- Sayısallaştırılmaya ait gürültülerin bulunması.
- Gözlük, sakal, bıyık, maske gibi detayların bulunması.

Tasarlanan Sistem

Kullanılan yönteme matematiksel terimler kullanarak açıklık getirmek istersek, yüzlerin dağılımını temsil eden temel bileşenlerin bulunması amaçlanmaktadır. Bir başka deyişle, yüzlerin dağılımından elde edilecek kovaryans matrisinin özvektörlerinin bulunması hedeflenmektedir. Bu özvektörler, yüzler arasındaki farklılıkları temsil eden bir grup özellik olarak düşünülebilir. Her bir özvektör, az yada çok bir yüze yakınsadığından, bu hayalete benzeyen yüzlerin “özyüz" adı ile anılması uygun görülmüştür. Şekil 2 de örnek özyüzler görülmektedir.

Eğitim kümesinde (training set) yer alan herbir yüz, bu özyüzlerin bir lineer kombinasyonu olarak ifade edilebilir. Özyüzlerin sayısı, eğitim kümesinde yer alan yüzlerin sayısına eşittir. Ancak, özdeğeri en büyük olanlar yani dağılıma en çok katkıda bulunanlar seçilerek bir yakınsama yapılabilir.

Özyüzlerin, yüz tanımada kullanılması, Sirovich ve Kirby ‘nin yüzleri en etkin şekilde temsil etmek amacıyla yaptığı bir çalışmadan esinlenilmiştir. Bu çalışmadan etkilenerek, daha önce M. Turk ve A. Pentland bir yüz tanıma çalışması yapmıştı. Biz de bu tezde, onların izlediği yolu takip ettik.

Tasarlanan yüz tanıma sisteminde, bir eğitim kümesi üzerinden özyüzler bulunarak, her bir yüz için bir özellik vektörü elde edilmektedir. Tanıma işlemi daha sonra, bu özellik vektörlerinin karşılaştırılması haline dönüşmektedir. Yüz tanıma süreci aşağıda belirtildiği gibi gelişmektedir:

- Eldeki yüz resimleri normalize edildikten sonra yüz veri tabanına eklenir.

- Normalize edilen bu resimlerden özyüzlerin oluşturulacağı bir eğitim kümesi seçilir. En yüksek özdeğere sahip olan M adet özyüz, özellik vektörlerinin eldesinde kullanılır. Özyüzler tanıma aşamasında kullanılmak üzere saklanırlar.

- Veri tabanında bulunan herbir yüz için, bu M adet özyüz kullanılarak M boyutlu bir özellik vektörü elde edilir.

- Yeni bir resim sisteme verildiğinde, önce normalize edilir ve daha sonra saklanmış olan M adet özyüz kullanılarak özellik vektörü elde edilir.

- Elde edilen bu özellik vektörü, veri tabanında yer alan yüzlerinkiler ile belli bir eşik (threshold) içerisinde karşılaştırılır. Eşik içine düşen en az bir resim varsa, yüz tanınmış demektir. Aksi halde bulunan bu özellik vektörü ile birlikte, yeni resim veri tabanına eklenerek öğrenme işlemi gerçekleştirilir.

Özyüzlerin Elde Edilmesi

Özyüzler, eğitim kümesinde yer alan resimlerin oluşturduğu dağılıma ait kovaryans matrisinin özvektörleridir. Eğitim kümesinde yer alan resimleri $\Gamma_1, \Gamma_2, .., \Gamma_M$ olarak sembolize edelim. Bu kümede yer alan resimlerin ortalaması $\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n$ olarak elde edilir. Şekil 1 de örnek bir çalışma kümesi ve ona ait ortalama resim görülmektedir. Her bir resimin ortalamadan farkını, $\Phi_j = \Gamma_j - \Psi$ vektörü ile ifade edelim. Elde edilen bu vektörler üzerinden, verinin dağılımını en iyi ifade eden M adet ortonormal u_n vektörü bulunmak istenmektedir. Bu u_n vektörlerinin herbiri aşağıda tanımı verilen λ_k katsayılarını maksimum yapacak şekilde seçilir.

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}(u_k^T \Phi_n)^2 \qquad (1)$$

U_n vektörleri, ortonormallik koşulunu olan (2) yi sağlamalıdırlar.

$$u_l^T u_k = \delta_{lk} = \begin{Bmatrix} 1, l = k \\ 0, l \neq k \end{Bmatrix} \qquad (2)$$

U_k vektörleri ve λ_k sabitleri, aşağıda tanımı verilen kovaryans matrisi C nin sırasıyla, özvektörleri ve özdeğerleridir.

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n \Phi_n^T = AA^T \qquad (3)$$

$$A = [\Phi_1 \Phi_2 \ldots \Phi_M] \qquad (4)$$

Dikkat edilecek olursa, C matrisi, N x N lik resimlerden oluşan bir eğitim kümesi için, $N^2 x N^2$ boyutundadır. Pratikte kullanılan resim boyutları için bu büyüklükte bir matrisin özvektörlerinin hesaplanması oldukça zordur.

Eğitim kümesinde yer alan resimlerin sayısının, resimlerin boyutundan çok daha küçük olduğunu ($M < N^2$) düşünürsek, N^2 den ziyade, M-1 adet anlamlı özvektörün olduğu ortaya çıkar. Bu durumda, diğer özvektörlerin özdeğerleri sıfır olacaktır. M x M lik bir matris özdeğerlerinin elde edilmesi yüz tanıma problemi için yeterli olmaktadır. Bu amaçla, M x M boyutunda olan bir L matrisi $L = A^T A$, her bir elemanı $L_{mn} = \Phi_m^T \Phi_n$ olacak şekilde oluşturulur. Dikkat edilirse, L matrisi köşegene göre simetriktir. Daha sonra, bu L matrisinin v_l ile gösterilen M adet özvektörü elde edilir. Bu v_l özvektörlerinin yardımı ile, gerçek özvektörler için aşağıda belirtilen yakınsama yapılır:

$$u_l = \sum_{k=1}^{M} v_{lk}\Phi_k , \qquad l = 1,...,M \qquad (5)$$

Özyüzler kullanılarak herhangi bir yüz resimine ait özellik vektörü Ω^T aşağıda belirtildiği gibi bulunabilir:

$$w_k = u_k^T(\Gamma - \Psi) \qquad (6)$$

$$\Omega^T = [w_1 w_2 \ldots w_M] \qquad (7)$$

Herhangi bir resim, yüz uzayına yapılacak aşağıda belirtilen projeksiyon ile yaklaşık olarak tekrar elde edilebilir:

$$\Gamma = \Psi + \sum_{i=1}^{M} w_i u_i \qquad (8)$$

Örnek eğitim kümesinden elde edilen özyüzler tarafından gerilen yüz uzayına yapılan, bu tip bir projeksiyon Şekil 3 de görülmektedir.

Deneysel Sonuçlar

Geliştirilen sistemi test etmek amacıyla, 14 değişik kişinin çeşitli pozlarından oluşan toplam 70 adet yüz resimi içeren bir test veri tabanı kurulmuştur. Bu resimlerin bazılarında, ışık kaynağı ve kafaların pozisyonları değiştirilerek çeşitli varyasyonlar sağlanmıştır. Ayrıca gözlük, maske, bıyık gibi detayların, tanıma performansına etkilerini incelemek amacıyla bazı resimlere adı geçen bu detaylar yapay olarak ilave edilmiştir.

Yapılan testlerde, doğru tanıma oranı ile eşik değerinin ters orantılı olduğu ortaya çıkmıştır. Eşik değeri arttırıldığında ıska (miss) oluşma sayısı azalmakta ancak, resimler hatalı tanınabilmektedir. Buna paralel olarak, özyüz adedinin tanıma performansına etkisi incelenmiştir. Doğru tanıma oranının, özyüz sayısı ile doğru orantılı olduğu tespit edilmiştir. Bunun nedeni, özyüzlerin sayısı arttığında, özellik vektörlerindeki bileşenlerin sayısının artmasıdır. Bu ise, resimlerin daha iyi temsil edilmesine neden olmaktadır.

Tanıma performasının en çok kafa pozisyonunun değişiminden etkilendiği gözlenmiştir. Işık kaynağında meydana gelen değişimler sistem üzerinde daha az etkili olmaktadır. Yüzlerde meydana gelen gözlük, maske, bıyık gibi küçük değişimler ise, tanıma performansı etkilememektedir.

İleriye Yönelik Düşünceler

Ileride, aşağıda belirtilen konular üzerinde çalışarak, sistemin daha da iyi bir hale getirilmesi amaçlanmaktadır:

- **Arka plan tespiti.** Yüzün arka planının tespit edilip, çıkartılarak gerçek yüz bilgisi ile çalışılmanın sağlanması.

- **Kamera ve tarayıcı desteği.** Şu anki sistem, yüz resimlerini sadece diskten alabilmektedir. Kamera ve tarayıcı desteği eklenerek, sisteme daha fazla esneklik kazandırılabilinir.

- **Yapay sinir ağı kullanımı.** Mevcut yapıda sinir ağı kullanılmamıştır. Ancak, sinir ağlarının bu tip uygulamalardaki başarısı artmakta olduğundan, sisteme bir sinir ağı yapısı uyarlanabilir.

- **İstemci/Sunucu mimari.** Geliştirilen sistem tek kullanıcıya yöneliktir. Daha etkin bir çalışma için, sistem çok kullanıcılı hale getirilerek, veri tabanının bir sunucu üzerinde oluşturulması düşünülmektedir.

(a) (b)

Şekil 1. (a) Örnek eğitim kümesi. (b) Ortalama resim.

Şekil 2. Örnek eğitim kümesinden elde edilmiş özyüzler.

(a) (b) (c)

Şekil 3. (a) Orijinal resim. (b) Maske takılmış hali. (c) Maskeli resimin yüz uzayına projeksiyonu ile elde edilen yeni resim.

Chapter 1

Introduction

The face is our primary focus of attention in social intercourse, playing a major role in conveying identity and emotion. Although the ability to infer intelligence or character from facial appearance is suspect, the human ability to recognize faces is remarkable. We can recognize thousands of faces learned throughout our lifetime and identify familiar faces at a glance even after years of separation. This skill is quite robust, despite large changes in the visual stimulus due to viewing conditions, expression, aging, and distractions such as glasses, beards or changes in hair style.

Face recognition has become an important issue in many applications such as security systems, credit card verification and criminal identification. For example, the ability to model a particular face and distinguish it from a large number of stored face models would make it possible to vastly improve criminal identification. Even the ability to merely detect faces, as opposed to recognizing them, can be important. Detecting faces in photographs for automating color film development can be very useful, since the effect of many enhancement and noise reduction techniques depends on the image content.

Although it is clear that people are good at face recognition, it is not at all obvious how faces are encoded or decoded by the human brain. Human face recognition has been studied for more than twenty years. Unfortunately developing a computational model of face recognition is quite difficult, because faces are complex, multi-dimensional visual stimuli. Therefore, face recognition is a very high level computer vision task, in which many early vision techniques can be involved.

The first step of human face identification is to extract the relevant features from facial images. Research in the field primarily intends to generate sufficiently reasonable familiarities of human faces so that another human can correctly identify the face. The question naturally arises as to how well facial features can be quantized. If such a quantization if possible then a computer should be capable of recognizing a face given a set of features. Investigations by numerous researchers [1, 2, 3] over the past several years have indicated that certain facial characteristics are used by human beings to identify faces.

There are three major research groups which propose three different approaches to the face recognition problem. The largest group [4, 5, 6] have dealt with facial characteristics which are used by human beings in recognizing individual faces. The second group [7, 8, 9, 10, 11] performs human face identification based on feature vectors extracted from profile silhouettes. The third group [12, 13] uses feature vectors extracted from a frontal view of the face. Although there are three different approaches to the face recognition problem, there are two basic methods from which these three different approaches arise.

The first method is based on the information theory concepts, in other words, on the principal component analysis methods. In this approach, the most relevant information that best describes a face is derived from the entire face image. Based on the Karhunen-Loeve expansion in pattern recognition, M. Kirby and L. Sirovich have shown that [4, 5] any particular face could be economically represented in terms of a best coordinate system that they termed "eigenfaces". These are the eigenfunctions of the averaged covariance of the ensemble of faces. Later, M. Turk and A. Pentland have proposed a face recognition method [14] based on the eigenfaces approach.

The second method is based on extracting feature vectors from the basic parts of a face such as eyes, nose, mouth, and chin. In this method, with the help of deformable templates and extensive mathematics, key information from the basic parts of a face is gathered and then converted into a feature vector. L. Yullie and S. Cohen [15] played a great role in adapting deformable templates to contour extraction of face images.

1.1 Eigenfaces for Recognition

We have focused our research toward developing a sort of unsupervised pattern recognition scheme that does not depend on excessive geometry and computations like deformable templates. Eigenfaces approach seemed to be an adequate method to be used in face recognition due to its simplicity, speed and learning capability.

A previous work based on the eigenfaces approach was done by M. Turk and A. Pentland, in which, faces were first detected and then identified. In this thesis, a face recognition system based on the eigenfaces approach, similar to the one presented by M. Turk and A. Pentland, is proposed.

The scheme is based on an information theory approach that decomposes face images into a small set of characteristic feature images called eigenfaces, which may be thought of as the principal components of the initial training set of face images. Recognition is performed by projecting a new image onto the subspace spanned by the eigenfaces and then classifying the face by comparing its position in the face space with the positions of known individuals.

Actual system is capable of both recognizing known individuals and learning to recognize new face images. The eigenface approach used in this scheme has advantages over other face recognition methods in its speed, simplicity, learning capability and robustness to small changes in the face image.

FACE-PRO, the actual face recognition software based on the eigenfaces approach was developed in C programming language on a personal computer. Although no optimizations were performed for matrix operations, during the tests on a Intel 80486 based personal computer, it was remarkable that the system could build a training set that had 14 members with 7 eigenfaces over a 58 member demo face library by updating all the feature vectors of the library members in around one minute. Once the training set has been built, recognitions were done near real time over this demo face library in less than one second.

1.2 Thesis Organization

This thesis is organized in the following manner: Chapter 2 deals with the basic concepts of pattern and face recognition. Two major approaches to the face recognition problem is given. Chapter 3 is based on the details of the proposed face recognition method and the actual system developed. Chapter 4 is dedicated to FACE-PRO, the face recognition software that was developed to demonstrate the eigenfaces approach. Finally, Chapter 5 gives the results drawn from the research and possible directions for future work.

Chapter 2

Basic Concepts of Pattern and Face Recognition

This chapter is dedicated to basic principals of pattern and face recognition. A brief summary about face recognition history together with two major face recognition approaches is presented for more insight on the subject.

2.1 Basic Concepts of Pattern Recognition

2.1.1 Overview

The need for improved information systems has become more conspicuous, since information is an essential element in decision making, and the world is generating increasing amounts of information in various forms with different degrees of complexity. One of the major problems in the design of modern information systems is automatic pattern recognition.

Recognition is regarded as a basic attribute of human beings, as well as other living organisms. A pattern is the description of an object. A human being is a very sophisticated information system, partly because he possesses a superior pattern recognition capability. According to the nature of the patterns to be recognized, recognition acts can be divided into two major types [16]:

Recognition of concrete items. This may be referred to as sensory recognition, which includes visual and aural pattern recognition. This recognition process involves the identification and classification of spatial and temporal patterns. Examples of spatial patterns are characters, fingerprints, physical objects, and images. Temporal patterns include speech waveforms, time series, electrocardiograms and target signatures.

- **Recognition of abstract items.** On the other hand, an old argument, or a solution to a problem can be recognized. This process involves the recognition of abstract items and can be termed conceptual recognition.

Recognition of concrete patterns by human beings may be considered as a psychophysiological problem which involves a relationship between a person and a physical stimulus. Human recognition is in reality a question of estimating the relative odds that the input data can be associated with one of a set of known statistical populations which depend on our past experience and which form the clues and the a priori information for recognition. Thus, the problem of pattern recognition may be regarded as one of discriminating the input data between populations via the search for features or invariant attributes among members of a population.

2.1.2 Pattern Classes and Patterns

Pattern recognition can be defined as the categorization of input data into identifiable classes via the extraction of significant features or attributes of the data from a background of irrelevant detail.

A **pattern class** is a category determined by some given common attributes or features. The features of a pattern class are the characterizing attributes common to all patterns belonging to that class. Such features are often referred to as intraset features. The features which represent the differences between pattern classes may be referred to as the interset features.

A **pattern** is the description of any member of a category representing a pattern class. For convenience, patterns are usually represented by a vector such as:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ \dots \\ x_n \end{bmatrix} \tag{2.1}$$

where each element x_j, represents a feature of that pattern. It is often useful to think of a pattern vector as a point in an n-dimensional Euclidean space.

2.1.3 Fundamental Problems in Pattern Recognition System Design

The design of an automatic pattern recognition systems generally involves several major problem areas:

- First of all, we have to deal with the representation of input data which can be measured from the objects to be recognized. This is the sensing problem. Each measured quantity describes a characteristic of the pattern or object. Inother words, a pattern vector that describes the input data has to be formed. The pattern vectors contain all the measured information available about the patterns. The set of patterns belonging to the same class corresponds to an ensemble of points scattered within some region of the measurement space. A simple example of this is shown in Figure 2.1 for two pattern classes denoted by w_1 and w_2.

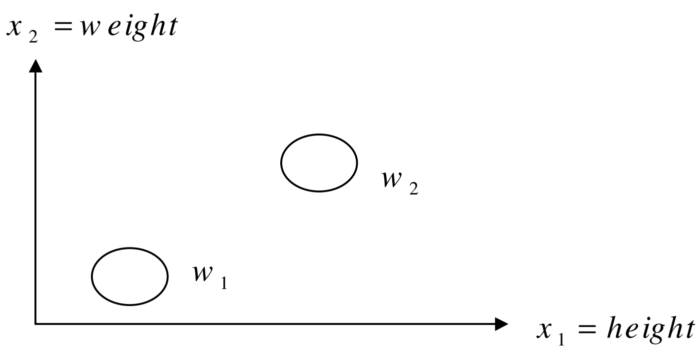

Figure 2.1. Two disjoint pattern classes. Each pattern is characterized by two measurements: height and weight. The pattern vector therefore is in the form of $x = \{x_1, x_2\}^T$.

- The second problem in pattern recognition concerns the extraction of characteristic features or attributes from the received input data and the reduction of the dimensionality of pattern vectors. This is often referred to as the pre-processing and the feature extraction problem. The elements of intraset features which are common to all pattern classes under consideration carry no discriminatory information and can be ignored. If a complete set of discriminatory features for each pattern class can be determined from the measured data, the recognition and classification of patterns will present little difficulty. Automatic recognition may be reduced to a simple matching process or a table look-up scheme. However, in most pattern recognition problems which arise in practice, the determination of a complete set of discriminatory features is extremely difficult, if not impossible.

- The third problem in pattern recognition system design involves the determination of the optimum decision procedures, which are needed in the identification and classification process. After the observed data from patterns to be recognized have been expressed in the form of pattern points or measurement vectors in the pattern space, we want the machine to decide to which pattern class these data belong. Let the system be capable of recognizing M different pattern classes. Then the pattern space can be considered as consisting of M regions, each of which encloses the pattern points of a class. The recognition problem can now be viewed as that of generating the decision boundaries which separate the M pattern classes on the basis of the observed measurement vectors. These decision boundaries are generally determined by decision functions.

2.1.4 Training and Learning

The decision functions can be generated in a variety of ways. When complete a priori knowledge about the patterns to be recognized is available, the decision function may be determined with precision on the basis of this information. When only qualitative knowledge about the patterns is available, reasonable guesses of the forms of the decision functions can be made. In this case the decision boundaries may be far from correct, and it is necessary to design the machine to achieve satisfactory performance through a sequence of adjustments.

The more general situation is that there exists little, if any, a priori knowledge about the patterns to be recognized. Under these circumstances pattern recognizing machines are best designed using a training or learning procedure. Arbitrary decision functions are initially assumed, and through a sequence of iterative training steps these decision functions are made to approach optimum or satisfactory forms.

It is important to keep in mind that learning or training takes place only during the design (or updating) phase of a pattern recognition system. Once acceptable results have been obtained with the training set of patterns, the system is applied to the task of actually performing recognition on samples drawn from the environment in which it is expected to operate. The quality of the recognition performance will be largely determined by how closely the training patterns resemble the actual data with which the system will be confronted during normal operation.

2.1.5 Supervised and Unsupervised Pattern Recognition

In most cases, representative patterns from each class under consideration are available. In these situations, supervised pattern recognition techniques are applicable. In a supervised learning environment, the system is taught to recognize patterns by means of various adaptive schemes. The essentials of this approach are a set of training patterns of known classification and the implementation of an appropriate learning procedure.

In some applications, only a set of training patterns of unknown classification may be available. In these situations, unsupervised pattern recognition techniques are applicable. As mentioned above, supervised pattern recognition is characterized by the fact that the correct classification of every training pattern is known. In the unsupervised case however, one is faced with the problem of actually learning the pattern classes present in the given data. This problem is also known as "learning without a teacher".

2.1.6 Outline of a Typical Pattern Recognition System

In Figure 2.2, functional block diagram of an adaptive pattern recognition system is shown. Although the distinction between optimum decision and pre-processing or feature extraction is not essential, the concept of functional breakdown provides a clear picture for the understanding of the pattern recognition problem.

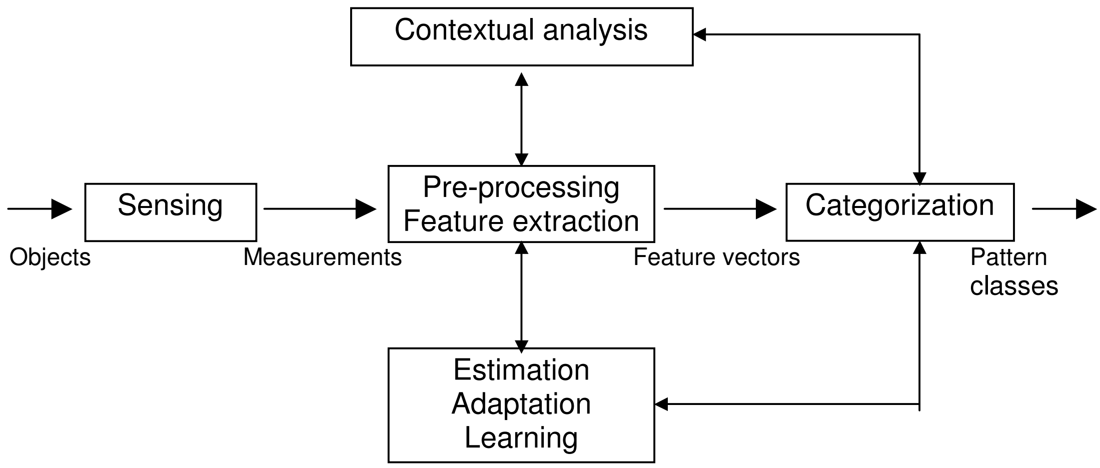

Figure 2.2. Functional block diagram of an adaptive pattern recognition system.

Correct recognition will depend on the amount of discriminating information contained in the measurements and the effective utilization of this information. In some applications, contextual information is indispensable in achieving accurate recognition. For instance, in the recognition of cursive handwritten characters and the classification of fingerprints, contextual information is extremely desirable. When we wish to design a pattern recognition system which is resistant to distortions, flexible under large pattern deviations, and capable of self-adjustment, we are confronted with the adaptation problem.

2.2 Face Recognition

Face recognition is a pattern recognition task performed specifically on faces. It can be described as classifying a face either "known" or "unknown", after comparing it with stored known individuals. It is also desirable to have a system that has the ability of learning to recognize unknown faces.

Computational models of face recognition must address several difficult problems. This difficulty arises from the fact that faces must be represented in a way that best utilizes the available face information to distinguish a particular face from all other faces. Faces pose a particularly difficult problem in this respect because all faces are similar to one another in that they contain the same set of features such as eyes, nose, mouth arranged in roughly the same manner.

2.2.1 Background and Related Work

Much of the work in computer recognition of faces has focused on detecting individual features such as the eyes, nose, mouth, and head outline, and defining a face model by the position, size, and relationships among these features. Such approaches have proven difficult to extend to multiple views and have often been quite fragile, requiring a good initial guess to quide them. Research in human strategies of face recognition, moreover, has shown that individual features and their immediate relationships comprise an insufficient representation to account for the performance of adult human face identification [17]. Nonetheless, this approach to face recognition remains the most popular one in the computer vision literature.

Bledsoe [18, 19] was the first to attempt semi-automated face recognition with a hybrid human-computer system that classified faces on the basis of fiducial marks entered on photographs by hand. Parameters for the classification were normalized distances and ratios among points such as eye corners, mouth corners, nose tip, and chin point. Later work at Bell Labs developed a vector of up to 21 features, and recognized faces using standard pattern classification techniques.

Fischler and Elschlager [20], attempted to measure similar features automatically. They described a linear embedding algorithm that used local feature template matching and a global measure of fit to find and measure facial features. This template matching approach has been continued and improved by the recent work of Yuille and Cohen [15]. Their strategy is based on deformable templates, which are parameterized models of the face and its features in which the parameter values are determined by interactions with the face image.

Connectionist approaches to face identification seek to capture the configurational nature of the task. Kohonen [21] and Kononen and Lehtio [22] describe an associative network with a simple learning algorithm that can recognize face images and recall a face image from an incomplete or noisy version input to the network. Fleming and Cottrell [23] extend these ideas using nonlinear units, training the system by backpropagation.

Others have approached automated face recognition by characterizing a face by a set of geometric parameters and performing pattern recognition based on the parameters. Kanade's [24] face identification system was the first system in which all steps of the recognition process were automated, using a top-down control strategy directed by a generic model of expected feature characteristics. His system calculated a set of facial parameters from a single face image and used a pattern classification technique to match the face from a known set, a purely statistical approach depending primarily on local histogram analysis and absolute gray-scale values.

Recent work by Burt [25] uses a smart sensing approach based on multiresolution template matching. This coarse to fine strategy uses a special purpose computer built to calculate multiresolution pyramid images quickly, and has been demonstrated identifying people in near real time.

2.2.2 Outline of a Typical Face Recognition System

In Figure 2.3, the outline of a typical face recognition system is given. This outline heavily carries the characteristics of a typical pattern recognition system that was presented in Figure 2.2.

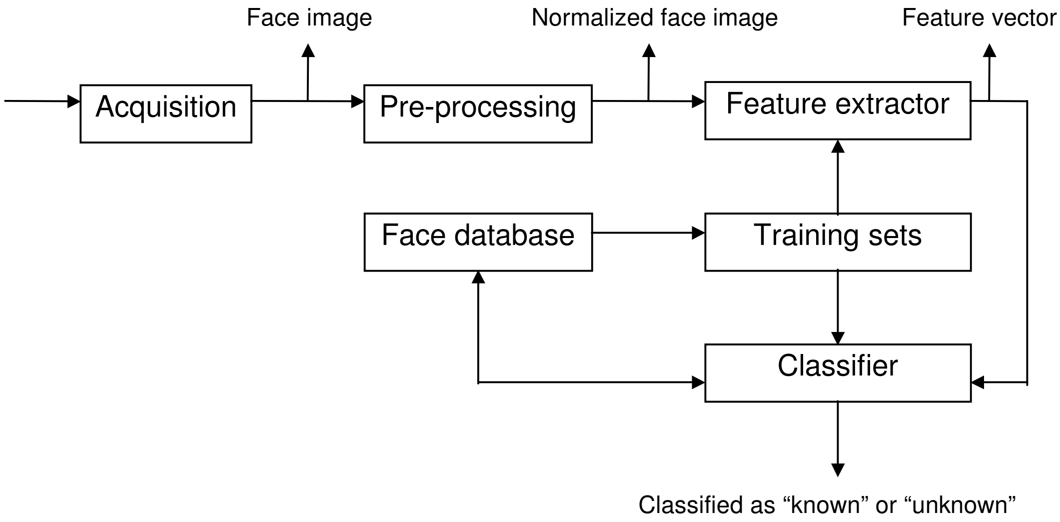

Figure 2.3. Outline of a typical face recognition system.

There are six main functional blocks, whose responsibilities are given below:

- **The acquisition module.** This is the entry point of the face recognition process. It is the module where the face image under consideration is presented to the system. Inother words, the user is asked to present a face image to the face recognition system in this module. An acquisition module can request a face image from several different environments: The face image can be an image file that is located on a magnetic disk, it can be captured by a frame grabber or it can be scanned from paper with the help of a scanner.

- **The pre-processing module.** In this module, by means of early vision techniques, face images are normalized and if desired, they are enhanced to improve the recognition performance of the system. Some or all of the following pre-processing steps may be implemented in a face recognition system:

- **Image size normalization**. It is usually done to change the acquired image size to a default image size such as 128 x 128, on which the face recognition system operates. This is mostly encountered in systems where face images are treated as a whole like the one proposed in this thesis.

- **Histogram equalization.** It is usually done on too dark or too bright images inorder to enhance image quality and to improve face recognition performance. It modifies the dynamic range (contrast range) of the image and as a result, some important facial features become more apparent.

- **Median filtering.** For noisy images especially obtained from a camera or from a frame grabber, median filtering can clean the image without loosing information.

- **High-pass filtering.** Feature extractors that are based on facial outlines, may benefit the results that are obtained from an edge detection scheme. High-pass filtering emphasizes the details of an image such as contours which can dramatically improve edge detection performance.

- **Background removal.** Inorder to deal primarily with facial information itself, face background can be removed. This is especially important for face recognition systems where entire information contained in the image is used. It is obvious that, for background removal, the pre-processing module should be capable of determining the face outline.

- **Translational and rotational normalizations.** In some cases, it is possible to work on a face image in which the head is somehow shifted or rotated. The head plays the key role in the determination of facial features. Especially for face recognition systems that are based on the frontal views of faces, it may be desirable that the pre-processing module determines and if possible, normalizes the shifts and rotations in the head position.

- **Illumination normalization.** Face images taken under different illuminations can degrade recognition performance especially for face recognition systems based on the principal component analysis in which entire face information is used for recognition. A picture can be equivalently viewed as an array of reflectivities r(x). Thus, under a uniform illumination I, the corresponding picture is given by

$$\Phi(x) = Ir(x) \tag{2.2}$$

The normalization comes in imposing a fixed level of illumination I_0 at a reference point x_0 on a picture. The normalized picture is given by

$$\Phi(x) = \frac{I_0 \Phi(x)}{I(x_0)} \tag{2.3}$$

In actual practice, the average of two reference points, such as one under each eye, each consisting of 2 x 2 array of pixels can be used.

- **The feature extraction module.** After performing some pre-processing (if necessary), the normalized face image is presented to the feature extraction module inorder to find the key features that are going to be used for classification. Inother words, this module is responsible for composing a feature vector that is well enough to represent the face image.

- **The classification module.** In this module, with the help of a pattern classifier, extracted features of the face image is compared with the ones stored in a face library (or face database). After doing this comparison, face image is classified as either known or unknown.

- **Training set.** Training sets are used during the "learning phase" of the face recognition process. The feature extraction, and the classification modules adjust their parameters inorder to achieve optimum recognition performance by making use of training sets.

- **Face library or face database.** After being classified as "unknown", face images can be added to a library (or to a database) with their feature vectors for later comparisons. The classification module makes direct use of the face library.

2.2.3 Problems that May Occur During Face Recognition

Due to the dynamic nature of face images, a face recognition system encounters various problems during the recognition process. It is possible to classify a face recognition system as either "robust" or "weak" based on its recognition performances under these circumstances. The objectives of a robust face recognition system is given below:

- **Scale invariance.** The same face can be presented to the system at different scales as shown in Figure 2.4-b. This may happen due to the focal distance between the face and the camera. As this distance gets closer, the face image gets bigger.

- **Shift invariance.** The same face can be presented to the system at different perspectives and orientations as shown in Figure 2.4-c. For instance, face images of the same person could be taken from frontal and profile views. Besides, head orientation may change due to translations and rotations.

- **Illumination invariance.** Face images of the same person can be taken under different illumination conditions such as, the position and the strength of the light source can be modified like the ones shown in Figure 2.4-d.

- **Emotional expression and detail invariance.** Face images of the same person can differ in expressions when smiling or laughing. Also, like the ones shown in Figure 2.4-e, some details such as dark glasses, beards or moustaches can be present.

- **Noise invariance.** A robust face recognition system should be insensitive to noise generated by frame grabbers or cameras. Also, it should function under partially occluded images.

A robust face recognition system should be capable of classifying a face image as "known" under even above conditions, if it has already been stored in the face database.

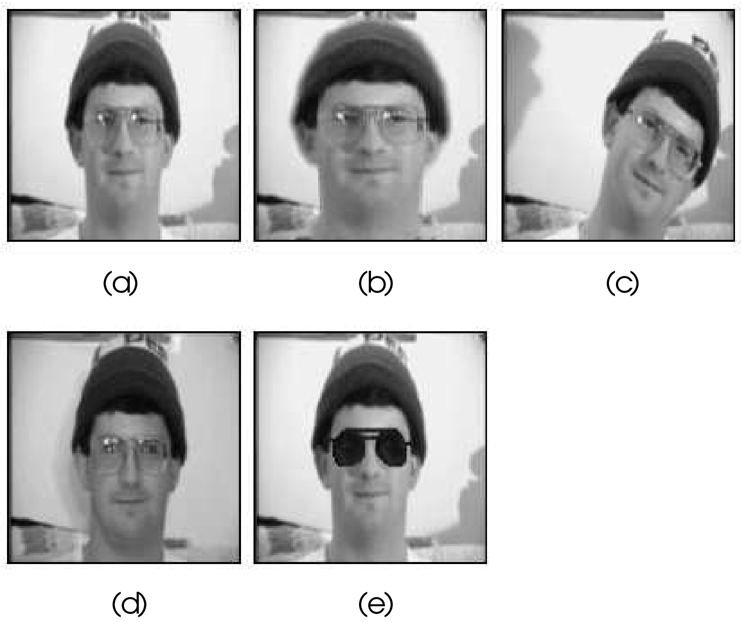

Figure 2.4. (a) Original face image. (b) Scale variance. (c) Orientation variance. (d) Illumination variance. (e) Presence of details.

2.2.4 Feature Based Face Recognition

It was mentioned before that, there were two basic approaches to the face recognition problem: Feature based face recognition and principal component analysis methods. Although feature based face recognition can be divided into two different categories, based on frontal views and profile silhouettes, they share some common properties and we will treat them as a whole. In this section, basic principals of feature based face recognition from frontal views [26] are presented.

2.2.4.1 Introduction

The first step of human face identification is to extract the features from facial images. In the area of feature selection, the question has been addressed in studies of cue salience in which discrete features such as the eyes, mouth, chin and nose have been found important cues for discrimination and recognition of faces.

After knowing what the effective features are for face recognition, some methods should be utilized to get contours of eyes, eyebrows, mouth, nose, and face. For different facial contours, different models should be used to extract them from the original portrait. Because the shapes of eyes and mouth are similar to some geometric figures, they can be extracted in terms of the deformable template model [15]. The other facial features such as eyebrows, nose and face are so variable that they have to be extracted by the active contour model [27, 28]. These two models can be illustrated in the following:

- **Deformable template model.** The deformable templates are specified by a set of parameters which uses a priori knowledge about the expected shape of the features to guide the contour deformation process. The templates are flexible enough to change their size and other parameter values, so as to match themselves to the data. The final values of these parameters can be used to describe the features. This method works well regardless of variations in scale, tilt, and rotations of the head. Variations of the parameters should allow the template to fit any normal instance of the feature. The deformable templates interact with the image in a dynamic manner. An energy function is defined which contains terms attracting the template to salient features such as peaks and valleys in the image intensity, edges and intensity itself. The minima of the energy function corresponds to the best fit with the image. The parameters of the template are then updated by steepest descent.

- **Active contour model (Snake).** The active contour or snake is an energy minimizing spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges. Snakes lock onto nearby edges, localizing them accurately. Because the snake is an energy minimizing spline, energy functions whose local minima comprise the set of alternative solutions to higher level processes should be designed. Selection of an answer from this set is accomplished by the addition of energy terms that push the model toward the desired solution. The result is an active model that falls into the desired solution when placed near it. In the active contour model issues such as the connectivity of the contours and the presence of corners affect the energy function and hence the detailed structure of the locally optimal contour. These issues can be resolved by very high-level computations.

2.2.4.2 Effective Feature Selection

Before mentioning the facial feature extraction procedures, we have the following two considerations:

- The picture-taking environment must be fixed inorder to get a good snapshot.
- Effective features that can be used to identify a face efficiently should be known.

Despite the marked similarity of faces as spatial patterns we are able to differentiate and remember a potentially unlimited number of faces. With sufficient familiarity, the faces of any two persons can be discriminated. The skill depends on the ability to extract invariant structural information from the transient situation of a face, such as changing hairstyles, emotional expression, and facial motion effect.

Features are the basic elements for object recognition. Therefore, to identify a face, we need to know what features are used effectively in the face recognition process. Because the variance of each feature associated with the face recognition process is relatively large, the features are classified into three major types:

- **First-order features values.** Discrete features such as eyes, eyebrows, mouth, chin, and nose, which have been found to be important [2] in face identification and are specified without reference to other facial features, are called first-order features. Important first-order features are given in Table 2.1.
- **Second-order features values.** Another configural set of features which characterize the spatial relationships between the positions of the first-order

features and information about the shape of the face are called second-order features. Important second-order features are given in Table 2.2. Second order features that are related to nose, if nose is noticeable are given in Table 2.3.

- **Higher-order feature values.** There are also higher-level features whose values depend on a complex set of feature values. For instance, age might be a function of hair coverage, hair color, skin tension, presence of wrinkles and age spots, forehead height which changes because of receding hairline, and so on.

Variability such as emotional expression or skin tension exists in the higher-order features and the complexity, which is the function of first-order and second-order features, is very difficult to predict. Permanent information belonging to the higher-order features can not be found simply by using first and second-order features. For a robust face recognition system, features that are invariant to the changes of the picture taking environment should be used. Thus, these features may contain merely first-order and second-order ones. These effective feature values cover almost all the obtainable information from the portrait. They are sufficient for the face recognition process.

The feature values of the second-order are more important than those of the first-order and they are dominant in the feature vector. Before mentioning the facial feature extraction process, it is necessary to deal with two pre-processing steps:

- **Threshold assignment**. Brightness threshold should be known inorder to discriminate the feature and other areas of the face. Generally, different thresholds are used for eyebrows, eyes, mouth, nose, and face according to the brightness of the picture.

- **Rough Contour Estimation Routine (RCER).** The left eyebrow is the first feature that is to be extracted. The first step is to estimate the rough contour of the left eyebrow and find the contour points. Generally, the position of the left eyebrow is about one-fourth of the facial width. Having this a priori information, the coarse position of the left eyebrow can be found and its rough contour can be captured. Once the rough contour of the left eyebrow is established, the rough contours of other facial features such as left eye, right eyebrow, mouth or nose can be estimated by RCER [29]. After the rough contour is obtained, its precise contour will be extracted by the deformable template model or the active contour model.

Table 2.1. First-order features

Measurement	Facial Location
Area, angle	left eyebrow right eyebrow left eye right eye mouth face
Distance	length of left eyebrow length of right eyebrow length of left eye length of right eye length of mouth length of face height of face

Table 2.2. Second-order features

Measurement	Facial Location
Distance	left eyebrow <-> right eyebrow left eye <-> right eye left eyebrow <-> left eye right eyebrow <-> right eye left eyebrow <-> mouth right eyebrow <-> mouth left eye <-> mouth right eye <-> mouth eyebrow <-> side of face eye <-> side of face mouth <-> side of face mouth <-> lower part of face
Angle	left eyebrow - left eye - left eyebrow right eyebrow - right eye - right eyebrow left eye - left eyebrow - left eye right eye - right eyebrow - right eye left eyebrow - mouth - right eyebrow left eye - mouth - right eye left eyebrow - left eye - mouth right eyebrow - right eye - mouth

Table 2.3. Features related to nose, if nose is noticeable.

Measurement	Facial Location
Distance	left nose <-> right nose
	left eyebrow <-> left nose
	right eyebrow <-> right nose
	left eye <-> left nose
	right eye <-> right nose
	left nose <-> mouth
	right nose <-> mouth
Angle	left eyebrow - center of nose - right eyebrow
	left eye - center of nose - right eye
	left nose - mouth - right nose
	left eyebrow - left eye - left nose
	right eyebrow - right eye - right nose

2.2.4.3 Feature Extraction Using the Deformable Templates

After the rough contour is obtained, the next step of face recognition is to find the physical contour of each feature. Conventional edge detectors can not find facial features such as the contours of the eye or mouth accurately from local evidence of edges, because they can not organize local information into a sensible global perception. There is a method to detect the contour of the eye by the deformable template which was originally proposed by Yullie [15]. It is possible to reduce computations at the cost of the precision of the extracted contour.

2.2.4.3.1 Eye Template

The deformable template acts on three representations of the image, as well as on the image itself. The first two representations are the peak and valleys in the image intensity and the third is the place where the image intensity changes quickly. The eye template developed by Yullie et al. consists of the following features:

- A circle of radius r, centered on a point (x_c, y_c), corresponding to the iris. The boundaries of the iris and the whites of the eyes are attracted to edges in the image intensity. The interior of the circle is attracted to valleys, or low values in the image intensity.

- A bounding contour of the eye attracted to edges. This contour is modeled by two parabolic sections representing the upper and lower parts of the boundary. It has a center (x_e, y_e), with 2w, maximum height h_1 of the boundary above the center, maximum height h_2 of the boundary below the center, and an angle of rotation ϕ.

- Two points, corresponding to the centers for the whites of the eyes, which are attracted to peaks in the image intensity.

- Regions between the bounding contour and the iris which also correspond to the whites of the eyes. These will be attracted to large intensity values.

The original eye template can be modified for the sake of simplicity where the accuracy of the extracted contour is not critical. The lack of a circle does not affect the classified results because the feature values are obtained from other information. The upper and lower parabola will be satisfactory for the recognition process. Thus, the total energy function for the eye template can be defined as a combination of the energy functions of edge, white and black points.

The total energy function is defined as

$$E_{total} = E_{edge} + E_{white} + E_{black} \tag{2.6}$$

where E_{edge}, E_{white}, and E_{black} are defined in the following:

- The edge potentials are given by the integral over the curves of the upper and lower parabola divided by their length:

$$\begin{aligned} E_{edge} = &-\frac{w_1}{upper\_length} \int_{upper-bound} \Phi_{edge}(x,y)dS \\ &-\frac{w_2}{lower\_length} \int_{lower-bound} \Phi_{edge}(x,y)dS \end{aligned} \tag{2.7}$$

where upper-bound and lower-bound represent the upper and lower parts of the eye, and Φ_{edge} represents the edge response of the point (x,y).

- The potentials of white and black points are defined as the integral over the area bounded by the upper and lower parabola divided by the area:

$$E_{w,b} = -\frac{1}{Area}\iint_{para-area}(-w_b N_{black}(x,y) + w_w N_{white}(x,y))dA \qquad (2.8)$$

where $N_{black}(x,y)$ and $N_{white}(x,y)$ represent the number of black and white points, and w_b , w_w are weights related with black and white points.

Inorder to be not affected by an improper threshold, the black and white points in Eq.(2.8) are defines as

P(x,y) is a black point if I(x,y) $\leq$ (threshold - tolerance),

P(x,y) is a white point if I(x,y) $\geq$ (threshold + tolerance),

P(x,y) is an unambigious point if I(x,y) is inbetween. (2.9)

where I(x,y) is the image intensity at point (x,y).

By the energy functions defined above, we can calculate the energy in the range of little modulations of 2w, h_1, h_2 and ϕ. When the minimum energy value takes place, the precise contour is extracted.

2.2.4.3.2 Mouth Template

In the whole features of the front view of the face, the role of the mouth is relatively important. The properties of the mouth contour are heavily involved in the face recognition process. The deformable mouth template changes its own shape when it comes across the image areas of edge (which the intensity changes quickly), and white and black points. Generally, features related to middle lips, lower and upper lips are extracted. Because of the effect of brightness in the picture taking period, the middle of the lower lip may not be apparent. RCER can not find the approximate height of the lower lip. Fortunately, the length of the mouth can still be found by RCER. Usually, the height of the lower lip is between one-fourth and one-sixth of the mouth's length.

The mouth contour energy function consists of the edge term E_{edge} and the black term E_{black}. The edge term dominates at the edge area, where as the black term encloses as many black points belonging to the mouth as possible.

$$E_{total} = E_{edge} + E_{black} \tag{2.10}$$

- The edge energy function consists of three parts: middle lip (gap between lips), lower lip and upper lip separated at philtrum. The equation of the middle lip part is

$$\begin{aligned} E_{edge} = &-\frac{w_{lower}}{lower\_length}\int_{lower}\Phi_{edge}(x,y)dS - \frac{w_{left}}{left\_length}\int_{left}\Phi_{edge}(x,y)dS \\ &-\frac{w_{right}}{right\_length}\int_{right}\Phi_{edge}(x,y)dS \end{aligned} \tag{2.11}$$

where *lower* represents the lower boundary of mouth, *left* represents the left part of upper lip, *right* represents the right part of upper lip, and Φ_{edge} (x,y) represents the edge response of point (x,y).

- The black energy function helps the edge energy to enclose black points belonging to the mouth and is defined as:

$$E_{black} = \frac{1}{Area}\int\int_{Lbound}^{Ubound} -w_{black}N_{black}(x,y)dA + \frac{1}{mid\_length}\int_{mid} -w_{mid}N_{black}(x,y)dS \tag{2.12}$$

where *Lbound* represents lower lip, *Ubound* represents upper lip, and *mid* represents number of black points. The black points are defined by Eq. (2.9). The weights w_{black}, w_{mid}, w_{lower}, w_{left} and w_{right} are experimentally determined.

2.2.4.4 Feature Extraction Using the Active Contour

The shapes of eyebrow, nostril and face, unlike eye and mouth, are even more different for different people and their contours can not be captured by using the deformable template. In this case, the active contour model or the "snake" is used. A snake is an energy minimizing spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges. This approach differs from traditional approaches which detect edges and then links them. In the active contour model, image and external forces together with the connectivity of the contours and the presence of corners will affect the energy function and the detailed structure of the locally optimal contour. The energy function of the active contour model [28] is defined as:

$$E_{snake} = \int_0^1 E_{snake}(v(s))ds = \int_0^1 E_{internal}(v(s)) + E_{images}(v(s)) + E_{constraint}(v(s))ds \tag{2.13}$$

where v(s) represents the position of the snake, $E_{internal}$ represents the internal energy of the contour due to bending, E_{images} gives rise to the image forces, and $E_{constraint}$ represents the external energy.

2.2.4.4.1 The Modified Active Contour Model

The original active contour model is user interactive. The advantage of its being user interactive is that the final form of the snake can be influenced by feedback from a higher level process. As the algorithm iterates the energy terms can be adjusted by higher level processes to obtain a local minima that seems most useful to that process. However, there are some problems with minimization procedure. Amini et al [30], pointed out some problems including instability and a tendency for points to bunch up on a strong portion of an edge. They proposed a dynamic programming algorithm for minimizing the energy function. Their approach had the advantage of using points on the discrete grid and is numerically stable, however the convergence is very slow.

It is possible to find a faster algorithm for the active contour [29]. Although this model still has the disadvantage of being unable to guarantee global minima, it can solve the problem of bunching up on a strong portion in the active contour. This problem occurs during the iterative process when contour points will accumulate at certain strong portions of the active contour. Besides, its computation speed is faster and thus, it is more suitable for face recognition. Active contour energy can be redefined [26] as:

$$E_{total} = \int_0^1 (\alpha(s)E_{continuity}(v(s)) + (\beta(s)E_{curvature}(v(s)) + (\delta(x)E_{image}(v(s))ds \qquad (2.14)$$

The definition of v(s) is similar to Eq (2.13) and the following approximations are used:

$$\left|\frac{dv_i}{ds}\right| \approx |v_i - v_{i-1}|^2 \quad \text{and} \quad \left|\frac{d^2 v_i}{ds^2}\right| \approx |v_{i-1} - 2v_i + v_{i+1}|^2 \qquad (2.15)$$

These formulas assume that the points on the contour are spaced at unit intervals and the parameter is the arc length. One effect of this is that unevenly spaced points will have higher curvature. Properties of the forces involved in the modified energy function can be described as follows:

- **Continuity force.** The first derivative $|v_i - v_{i-1}|^2$ causes the curve to shrink. It is actually minimizing the distance between points. It also contributes to the problem of points bunching up on strong portions of the contour. It was decided that a term which encouraged even spacing of the points would satisfy the original goal of first order continuity without the effect of shrinking. Here, this term uses the difference between the average distance points, ***d***, and this distance between the two points under consideration, $\left|d - |v_i - v_{i-1}|\right|$. Thus points having a distance near the average will have the minimum value. At the end of each iteration a new value of *d* is computed.

- **Curvature force.** Since the formulation of the continuity term causes the points to be relatively evenly spaced, $|v_{i-1} - 2v_i + v_{i+1}|^2$ gives a reasonable and quick estimate of the curvature. This term, like the continuity term, is normalized by dividing the largest value in the neighbourhood, giving a number from 0 to 1.

- **Image force.** E_{images} is the image force which is defined by the following operations:

    1. We have eight image energy measurements (Mag), for eight-neighbours.
    2. To normalize the image energy measurements, we select the minimum (Min), and maximum (Max) terms from those eight measurements, and then do the calculation, (Min - Mag)/(Max - Min) to obtain the image force.

At the end of each iteration, the curvature is determined at each point on the new contour. If the value is larger than the threshold, β, is set to 0 for the next iteration. The greedy algorithm [29] is applied for fast convergence. The energy function is computed for the current location of vi and each of its neighbours. The neighbour having the smallest value is chosen as the new position of v_i. The greedy algorithm can be applied to extract the features of eyebrow, nostril, and face. Inorder to prevent the snake failing at inaccurate local minima, contour estimation is done via RCER before the snake iterates. RCER uses a priori knowledge to find a rough contour as a starting (initial) contour for the snake.

2.2.4.4.2 Boundary Extraction of a Face

Inorder to demonstrate the use of active contour models on facial contour extraction, energy function associated to the boundary extraction of a face is presented in this section.

Unlike eyebrow extraction, the boundary extraction of a face is more time consuming because the rough contour of a face can not be estimated by RCER. However, the rough contour need to be approximated as accurately as possible. The energy function associated with boundary extraction of a face is defined as

$$E_{face} = \sum_{i=1}^{n} (\alpha_i E_{continuity} + \beta_i E_{curvature} + \delta_i E_{images}) \tag{2.16}$$

where $E_{continuity}$ and $E_{curvature}$ are defined in *section 2.2.4.4.1* and E_{images} is defined as

$$E_{images} = w_{lines} E_{lines} + w_{edge} E_{edge}. \tag{2.17}$$

The goal of E_{lines} is to attract more white points and less dark ones inside the active contour, where as the goal of E_{edge} is to attract more edge points on the active contour boundary.

E_{lines} and E_{edge} are defined as (2.18)

$$E_{lines} = \begin{cases} -Mag_{white} & \text{, if } \beta(x,y) = 0 \text{ and } I(x,y) \geq \text{threshold} + \text{tolerance} \\ Mag_{dark} & \text{, if } \beta(x,y) = 0 \text{ and } I(x,y) < \text{threshold} - \text{tolerance} \\ -Mag_{white} & \text{, if } \beta(x,y) = 1 \text{ and } I(x,y) \geq \text{threshold} + \text{tolerance} \\ Mag_{dark} & \text{, if } \beta(x,y) = 1 \text{ and } I(x,y) < \text{threshold} - \text{tolerance} \\ -Mag_{contour} & \text{, if } \beta(x,y) = 255 \text{ and } I(x,y) \geq \text{threshold} + \text{tolerance} \\ Mag_{contour} & \text{, if } \beta(x,y) = 255 \text{ and } I(x,y) < \text{threshold} - \text{tolerance} \\ 0 & \text{, otherwise} \end{cases}$$

$$E_{edge} = \left| \begin{array}{l} I(x,y) - \frac{1}{8}(I(x+1,y+1) + I(x+1,y) + I(x+1,y-1) + I(x,y+1) \\ \qquad + I(x,y-1) + I(x-1,y+1) + I(x-1,y) + I(x-1,y-1)) \end{array} \right|$$

where *I(x,y)* represents the intensity value of point x,y), *threshold* is the same as face threshold, *tolerance* is experimentally determined.

To extract the boundary of a face without a beard, the snake iterates and moves toward the chin because E_{face} decreases constantly. The convergent process of the snake is based on the greedy algorithm. When the iterative process stops, iterations can be re-started inorder to find its next convergent place. If they are similar, the position is accurate.

2.2.5 Face Recognition Based on Principal Component Analysis

In section 2.2.4, we have reviewed a face recognition method based on feature extraction. By using extensive geometry, it is possible to find the contours of the eye, eyebrow, nose, mouth, and even the face itself.

Principal component analysis for face recognition is based on the information theory approach. Here, the relevant information in a face image is extracted and encoded as efficiently as possible. Recognition is performed on a face database that consists of models encoded similarly.

In mathematical terms, the principal components of the distribution of faces or the eigenvectors of the covariance matrix of the set of face images, treating an image as a point (vector) in a very high dimensional face space is sought. In Chapter 3, a principal component analysis method will be presented in more detail.

Chapter 3

Face Recognition Using Eigenfaces

This chapter is dedicated to the explanation of a principal component analysis method that is used in face recognition, on which the proposed face recognition system is based.

3.1 Introduction

Much of the previous work on automated face recognition has ignored the issue of just what aspects of the face stimulus are important for face recognition. This suggests the use of an information theory approach of coding and decoding of face images, emphasizing the significant local and global features. Such features may or may not be directly related to our intuitive notion of face features such as the eyes, nose, lips, and hair.

In the language of information theory, the relevant information in a face image is extracted, encoded as efficiently as possible, and then compared with a database of models encoded similarly. A simple approach to extracting the information contained in an image of a face is to somehow capture the variation in a collection of face images, independent of any judgment of features, and use this information to encode and compare individual face images.

In mathematical terms, the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face images, treating an image as point (or vector) in a very high dimensional space is sought.

The eigenvectors are ordered, each one accounting for a different amount of the variation among the face images.

These eigenvectors can be thought of as a set of features that together characterize the variation between face images. Each image location contributes more or less to each eigenvector, so that it is possible to display these eigenvectors as a sort of ghostly face image which is called an "eigenface".

Sample face images and the corresponding eigenfaces are shown in Figure 3.1-a and in Figure 3.2 respectively. Each eigenface deviates from uniform gray where some facial feature differs among the set of training faces. Eigenfaces can be viewed as a sort of map of the variations between faces.

Figure 3.1. (a) Sample, training set face images. (b) Average face image of the training set.

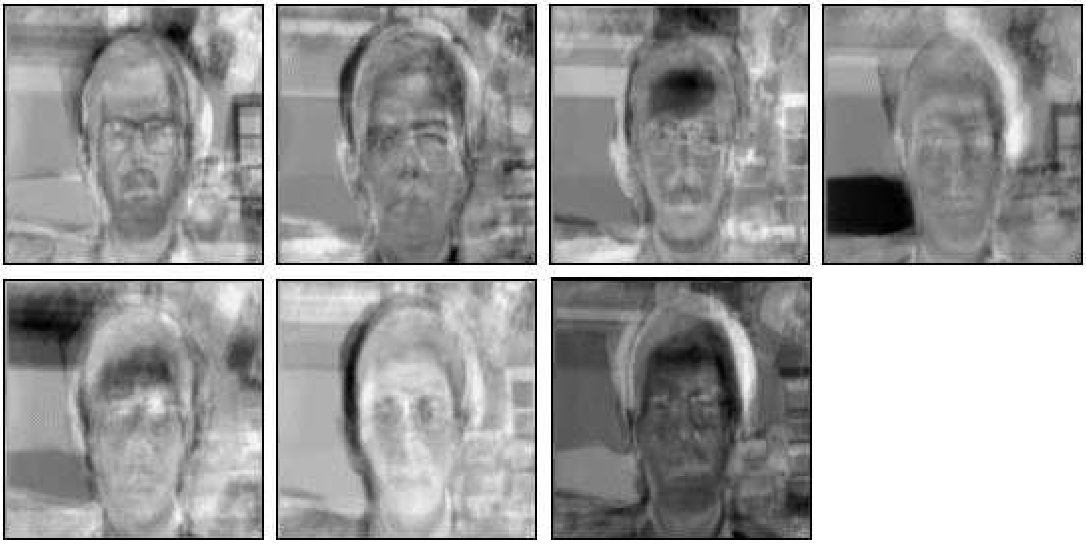

Figure 3.2. 7 eigenfaces with highest eigenvalues, that were calculated from the sample training set, given in Figure 3.1.

Each individual face can be represented exactly in terms of a linear combination of the eigenfaces. Each face can also be approximated using only the "best" eigenfaces, those that have the largest eigenvalues, and which therefore account for the most variance within the set of face images. The best M eigenfaces span an M-dimensional subspace which we call the "face space" of all possible images.

Kirby and Sirovich [4, 5] developed a technique for efficiently representing pictures of faces using principal component analysis. Starting with an ensemble of original face images, they calculated a best coordinate system for image compression, where each coordinate is actually an image that they termed an "eigenpicture". They argued that, at least in principle, any collection of face images can be approximately reconstructed by storing a small collection of weights for each face, and a small set of standard pictures (the eigenpictures). The weights describing each face are found by projecting the face image onto each eigenpicture.

In this thesis, we have followed the method which was proposed by M. Turk and A. Pentland [14] inorder to develop a face recognition system based on the eigenfaces approach. They argued that, if a multitude of face images can be reconstructed by weighted sum of a small collection of characteristic features or eigenpictures, perhaps an efficient way to learn and recognize faces would be to build up the characteristic features by experience over time and recognize particular faces by comparing the feature weights needed to approximately reconstruct them with the weights associated with known individuals. Therefore, each individual is characterized by a small set of feature or eigenpicture weights needed to describe and reconstruct them. This is an extremely compact representation when compared with the images themselves.

3.2 Outline of the Proposed Face Recognition System

The proposed face recognition system passes through three main phases during a face recognition process. Three major functional units are involved in these phases and they are depicted in Figure 3.3. The characteristics of these phases inconjunction with the three functional units are given below:

- **Face library formation phase.** In this phase, the acquisition and the pre-processing of the face images that are going to be added to the face library are performed. Face images are stored in a face library in the system. We call this face database a "face library" because at the moment, it does not have the properties of a relational database. Every action such as training set or eigenface formation is performed on this face library. Face library is initially empty. Inorder to start the face recognition process, this initially empty face library has to be filled with face images. The proposed face recognition system operates on 128 x 128 x 8, HIPS formatted image files. At the moment, scanner or camera support is unavailable. Inorder to perform image size conversions and enhancements on face images, there exists the "pre-processing" module. This module automatically converts every face image to 128 x 128 (if necessary) and based on user request, it can modify the dynamic range of face images (histogram equalization) inorder to improve face recognition performance. Also, we have considered to implement a "background removal" algorithm in the pre-processing module, but due to time limitations, we have left this for future work. After acquisition and pre-processing, face image under consideration is added to the face library. Each face is represented by two entries in the face library: One entry corresponds to the face image itself (for the sake of speed, no data compression is performed on the face image that is stored in the face library) and the other corresponds to the weight vector associated for that face image. Weight vectors of the face library members are empty until a training set is chosen and eigenfaces are formed.

- **Training phase.** After adding face images to the initially empty face library, the system is ready to perform training set and eigenface formations. Those face images that are going to be in the training set are chosen from the entire face library. Because that the face library entries are normalized, no further pre-processing is necessary at this step. After choosing the training set, eigenfaces are formed and stored for later use. Eigenfaces are calculated from the training set, keeping only the M images that correspond to the highest eigenvalues. These M eigenfaces define the M-dimensional "face space". As new faces are experienced, the eigenfaces can be updated or recalculated. The corresponding distribution in the M-dimensional weight space is calculated for each face library member, by projecting its face image onto the "face space" spanned by the eigenfaces. Now the corresponding weight vector of each face library member has been updated which were initially empty. The system is now ready for the recognition process. Once a training set has been chosen, it is not possible to add new members to the face library with the conventional method that is presented in "phase 1" because, the system does not know whether this item already exists in the face library or not. A library search must be performed.

- **Recognition and learning phase.** After choosing a training set and constructing the weight vectors of face library members, now the system is ready to perform the recognition process. User initiates the recognition process by choosing a face image. Based on the user request and the acquired image size, pre-processing steps are applied to normalize this acquired image to face library specifications (if necessary). Once the image is normalized, its weight vector is constructed with the help of the eigenfaces that were already stored during the training phase. After obtaining the weight vector, it is compared with the weight vector of every face library member within a user defined "threshold". If there exists at least one face library member that is similar to the acquired image within that threshold then, the face image is classified as "known". Otherwise, a miss has occurred and the face image is classified as "unknown". After being classified as unknown, this new face image can be added to the face library with its corresponding weight vector for later use (learning to recognize).

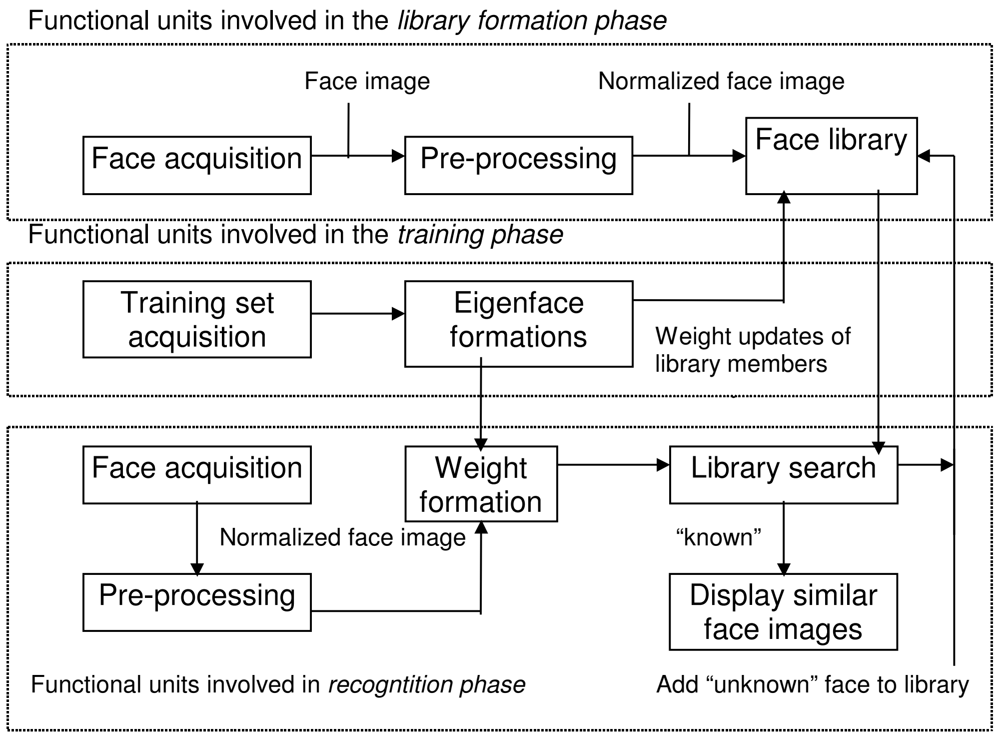

Figure 3.3. Functional block diagram of the proposed face recognition system.

3.3 Calculating Eigenfaces

Let a face image I(x,y) be a two-dimensional N x N array of 8-bit intensity values. An image may also be considered as a vector of dimension N^2, so that a typical image of size 256 x 256 becomes a vector of dimension 65,536, or equivalently a point in 65,536-dimensional space. An ensemble of images, then, maps to a collection of points in this huge space.

Images of faces, being similar in overall configuration, will not be randomly distributed in this huge image space and thus can be described by a relatively low dimensional subspace. The main idea of the principal component analysis (or Karhunen-Loeve expansion) is to find the vectors that best account for the distribution of face images within the entire image space.

These vectors define the subspace of face images, which we call "face space". Each vector is of length N^2, describes an N x N image, and is a linear combination of the original face images. Because these vectors are the eigenvectors of the covariance matrix corresponding to the original face images, and because they are face-like in appearance, we refer to them as "eigenfaces". Some examples of eigenfaces are shown in Figure 3.2.

Definitions:

An N x N matrix A is said to have an eigenvector X, and corresponding eigenvalue λ if

$$AX = \lambda X. \tag{3.1}$$

Evidently, Eq. (3.1) can hold only if

$$\det|A - \lambda I| = 0 \tag{3.2}$$

which, if expanded out, is an Nth degree polynomial in λ whose root are the eigenvalues. This proves that there are always N (not necessarily distinct) eigenvalues. Equal eigenvalues coming from multiple roots are called "degenerate".

A matrix is called *symmetric* if it is equal to its transpose,

$$A = A^T \text{ or } a_{ij} = a_{ji} \tag{3.3}$$

it is termed *orthogonal* if its transpose equals its inverse,

$$A^T A = AA^T = I \tag{3.4}$$

finally, a real matrix is called *normal* if it commutes with is transpose,

$$AA^T = A^T A. \tag{3.5}$$

Theorem: Eigenvalues of a real symmetric matrix are all real. Contrariwise, the eigenvalues of a real nonsymmetric matrix may include real values, but may also include pairs of complex conjugate values. The eigenvalues of a normal matrix with nondegenerate eigenvalues are complete and orthogonal, spanning the N-dimensional vector space.

After giving some insight on the terms that are going to be used in the evaluation of the eigenfaces, we can deal with the actual process of finding these eigenfaces.

Let the training set of face images be $\Gamma_1, \Gamma_2, \ldots, \Gamma_M$ then the average of the set is defined by

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n \, . \tag{3.6}$$

Each face differs from the average by the vector

$$\Phi_i = \Gamma_i - \Psi \, . \tag{3.7}$$

An example training set is shown in Figure 3.1-a, with the average face Ψ shown in Figure 3.1-b.

This set of very large vectors is then subject to principal component analysis, which seeks a set of M orthonormal vectors, u_n, which best describes the distribution of the data. The kth vector, u_k, is chosen such that

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}(u_k^T \Phi_n)^2 \tag{3.8}$$

is a maximum, subject to

$$u_l^T u_k = \delta_{lk} = \begin{Bmatrix} 1, & \text{if } l = k \\ 0, & \text{otherwise} \end{Bmatrix} \tag{3.9}$$

The vectors u_k and scalars λ_k are the eigenvectors and eigenvalues, respectively of the covariance matrix

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n \Phi_n^T = AA^T \, . \tag{3.10}$$

where the matrix $A = \left[\Phi_1 \Phi_2 \ldots \Phi_M\right]$ The covariance matrix C, however is $N^2 x N^2$ real symmetric matrix, and determining the N^2 eigenvectors and eigenvalues is an intractable task for typical image sizes. We need a computationally feasible method to find these eigenvectors.

If the number of data points in the image space is less than the dimension of the space (M < N^2), there will be only M-1, rather than N^2, meaningful eigenvectors. The remaining eigenvectors will have associated eigenvalues of zero. We can solve for the N^2 dimensional eigenvectors in this case by first solving the eigenvectors of an M x M matrix such as solving 16 x 16 matrix rather than a 16,384 x 16,384 matrix and then, taking appropriate linear combinations of the face images Φ_i.

Consider the eigenvectors v_i of $A^T A$ such that

$$A^T A v_i = \mu_i v_i \tag{3.11}$$

Premultiplying both sides by A, we have

$$AA^T A v_i = \mu_i A v_i \tag{3.12}$$

from which we see that Av_i are the eigenvectors of $C = AA^T$.

Following these analysis, we construct the M x M matrix $L = A^T A$, where $L_{mn} = \Phi_m^T \Phi_n$, and find the M eigenvectors, v_l, of L. These vectors determine linear combinations of the M training set face images to form the eigenfaces u_l.

$$u_l = \sum_{k=1}^{M} v_{lk} \Phi_k \ , \qquad l = 1,\ldots,M \tag{3.13}$$

With this analysis, the calculations are greatly reduced, from the order of the number of pixels in the images (N^2) to the order of the number of images in the training set (M). In practice, the training set of face images will be relatively small (M << N^2), and the calculations become quite manageable. The associated eigenvalues allow us to rank the eigenvectors according to their usefulness in characterizing the variation among the images.

The success of this algorithm is based on the evaluation of the eigenvalues and eigenvectors of the real symmetric matrix L that is composed from the training set of images. Root searching in the characteristic equation, Eq. (3.2) is usually a very poor computational method for finding eigenvalues. During the programming phase of the above algorithm, a more efficient method [31] was used inorder to evaluate the eigenvalues and eigenvectors. At first, the real symmetric matrix is reduced to tridiagonal form with the help of the "Householder" algorithm. The Householder algorithm reduces an N x N symmetric matrix A to tridiagonal form by N - 2 orthogonal transformations. Each transformation annihilates the required part of a whole column and whole corresponding row. After that, eigenvalues and eigenvectors are obtained with the help of QR transformations. The basic idea behind the QR algorithm is that any real symmetric matrix can be decomposed in the form A = QR where Q is orthogonal and R is upper triangular. The workload in the QR algorithm is O(N^3) per iteration for a general matrix, which is prohibitive. However, the workload is only O(N) per iteration for a tridiagonal matrix, which makes it extremely efficient.

3.4 Using Eigenfaces to Classify a Face Image

The eigenface images calculated from the eigenvectors of L, span a basis set with which to describe face images. Sirovich and Kirby evaluated a limited version of this framework on an ensemble of M = 115 images of Caucasian males digitized in a controlled manner, and found that 40 eigenfaces were sufficient for a very good description of face images. With M' = 40 eigenfaces, RMS pixel by pixel errors in representing cropped versions of face images were about 2%.

In practice, a smaller M' can be sufficient for identification, since accurate reconstruction of the image is not a requirement. Based on this idea, the proposed face recognition system lets the user specify the number of eigenfaces (M') that is going to be used in the recognition. For maximum accuracy, the number of eigenfaces should be equal to the number of images in the training set. But, it was observed that, for a training set of fourteen face images, seven eigenfaces were enough for a sufficient description of the training set members.

In this framework, identification becomes a pattern recognition task. The eigenfaces span an M' dimensional subspace of the original N^2 image space. The M' significant eigenvectors of the L matrix are chosen as those with the largest associated eigenvalues.

A new face image (Γ) is transformed into its eigenface components (projected onto "face space") by a simple operation,

$$w_k = u_k^T (\Gamma - \Psi) \tag{3.14}$$

for k = 1,...,M'. This describes a set of point by point image multiplications and summations, operations performed at approximately frame rate on current image processing hardware, with a computational complexity of O(N^4).

The weights form a feature vector,

$$\Omega^T = \left[w_1 w_2 \ldots w_{M'} \right] \tag{3.15}$$

that describes the contribution of each eigenface in representing the input face image, treating the eigenfaces as a basis set for face images. The feature vector is then used in a standard pattern recognition algorithm to find which of a number of predefined face classes, if any, best describes the face. The face classes Ω_i can be calculated by averaging the results of the eigenface representation over a small number of face images (as few as one) of each individual. In the proposed face recognition system, face classes contain only one representation of each individual.

Classification is performed by comparing the feature vectors of the face library members with the feature vector of the input face image. This comparison is based on the Euclidean distance between the two members to be smaller than a user defined threshold ε_k. This is given in Eq. (3.16). If the comparison falls within the user defined threshold, then face image is classified as “known”, otherwise it is classified as “unknown” and can be added to face library with its feature vector for later use, thus making the system learning to recognize new face images.

$$\frac{\|\Omega - \Omega_k\|}{\|\Omega_k\|} \le \varepsilon_k \tag{3.16}$$

3.5 Rebuilding a Face Image with Eigenfaces

A face image can be approximately reconstructed (rebuilt) by using its feature vector and the eigenfaces as

$$\Gamma' = \Psi + \Phi_f \tag{3.17}$$

where

$$\Phi_f = \sum_{i=1}^{M'} w_i u_i \tag{3.18}$$

is the projected image.

Eq. (3.17) tells that the face image under consideration is rebuilt just by adding each eigenface with a contribution of w_i Eq. (3.18) to the average of the training set images. The degree of the fit or the "rebuild error ratio" can be expressed by means of the Euclidean distance between the original and the reconstructed face image as given in Eq. (3.19).

$$\text{Rebuild error ratio} = \frac{\|\Gamma' - \Gamma\|}{\|\Gamma\|} \tag{3.19}$$

It has been observed that, rebuild error ratio increases as the training set members differ heavily from each other. This is due to the addition of the average face image. When the members differ from each other (especially in image background) the average face image becomes more messy and this increases the rebuild error ratio.

There are four possibilities for an input image and its pattern vector:

1. Near face space and near a face class,
2. Near face space but not near a known face class,
3. Distant from face space and near a face class,
4. Distant from face space and not near a known face class.

In the first case, an individual is recognized and identified. In the second case, an unknown individual is presented. The last two cases indicate that the image is not a face image. Case three typically shows up as a false classification. It is possible to avoid this false classification in this system by using Eq. (3.18) as

$$\frac{\|\Phi - \Phi_f\|}{\|\Phi_f\|} \leq \phi_k \tag{3.20}$$

where ϕ_k is a user defined threshold for the faceness of the input face images belonging to kth face class..

3.6 Summary of the Eigenface Recognition Procedure

The eigenfaces approach to face recognition can be summarized in the following steps:

- Form a face library that consists of the face images of known individuals.
- Choose a training set that includes a number of images (M) for each person with some variation in expression and in the lighting.
- Calculate the M x M matrix L, find its eigenvectors and eigenvalues, and choose the M' eigenvectors with the highest associated eigenvalues.
- Combine the normalized training set of images according to Eq. (3.13) to produce M' eigenfaces. Store these eigenfaces for later use.

- For each member in the face library, compute and store a feature vector according to Eq. (3.15).

- Choose a threshold ε according to (3.16) that defines the maximum allowable distance from any face class. Optionally choose a threshold ϕ according to Eq. (3.20) that defines the maximum allowable distance from face space.

- For each new face image to be identified, calculate its feature vector according to Eq. (3.15) and compare it with the stored feature vectors of the face library members. If the comparison satisfies the condition given in Eq. (3.16) for at least one member, then classify this face image as "known", otherwise a miss has occured and classify it as "unknown" and add this member to the face library with its feature vector.

3.7 Comparison of the Eigenfaces approach to Feature Based Face Recognition

These two different approaches are compared based on the following aspects of face recognition:

- **Speed and simplicity.** Feature based face recognition involves complex computations such as deformable templates and active contour models. The evaluation of these parameters are very time consuming even on todays computers. Eigenfaces approach is superior in its speed and reasonably simple implementation. In Eq. (3.14), it is seen that the evaluation of a feature vector involves merely additions and multiplications. On a machine that is capable of executing an addition and a multiplication in one clock cycle, this feature evaluation and comparison can be done in real time. During the experiments, although no special hardware (and special software optimizations for matrix operations) was used, the speed of the proposed face recognition system was near real time.

- **Learning capability.** Feature based face recognition systems are generally trained to optimize their parameters in a supervised manner. That is, the system designer presents known individuals to the system and checks system response. In the eigenfaces approach, training is done in an unsupervised manner. User selects a training set that represents the rest of the face images. Eigenfaces are obtained from the training set members and feature vectors are formed.

- **Face background.** Background of the face images are extremely important in the eigenface approach. Eigenfaces and feature vectors are evaluated by image multiplication and additions. As a result of this, entire information contained in the face image is used. If this information changes due to face background, recognition performance can significantly decrease. Inorder to avoid this, a "background removal" algorithm can be implemented in a pre-processing step. Feature based face recognition algorithms can be less sensitive to face background in case, they generally locate face contours inorder to extract facial features.

- **Scale and orientation.** The experiments in section 3.8 show that recognition performance decreases quickly as head size or orientation is misjudged. The head size and orientation in the input image must be close to that of the eigenfaces for the system to work well. Inorder to overcome this problem, multiscale eigenfaces can be used or the head can be removed from the rest of the image, and then scaled or rotated to meet the specifications of the eigenfaces. Again, feature based face recognition algorithms can score better in this comparison because they find facial features by using deformable templates and active contour models that are less sensitive to scale and orientation.

- **Presence of small details.** Feature based face recognition algorithms can suffer when some details are present on the face image such as dark glasses or beards. For a feature based face recognition system, it is quite impossible to extract the features that are related to the eyes when dark glasses is present on the face. Also, active contour models can suffer when a beard is present on the face while locating the face contour. Eigenfaces approach excels in this aspect of face recognition. Small changes in face images such as glasses, beards or moustaches does not cause a decrease in the face recognition performance because the information that is present in the rest of the face image makes it enough to be classified correctly.

3.8 Experimental Results

In this section, experimental results that were obtained from the proposed face recognition system are given.

3.8.1 Database of Face Images

Inorder to test the viability of this approach to face recognition, a sample set of face images were created under known variations of lighting and orientation. Besides, some of these original face images were manually decorated to add facial details such as glasses or beards.

Most of these face images were obtained from the MIT face database that actually contained over 2,500 face images, digitized under controlled conditions. Face images were in 128 x 120 x 8, HIPS format. They contained different appearances of 14 individuals, digitized under two head orientations and two lighting conditions. After manually adding some facial details, a face library with 70 members were constructed.

For convenience, these face images were classified with the name of the individuals, followed by a suffix indicating the properties of the current appearance of that individual. There are five different suffixes (five different appearances) that are given below:

"_0" : The original appearances. There are 12 face images in this category. Although there are 14 different individuals, only 12 of them were successfully retrieved.

"_1" : Light source modification. Light source in "_0" was moved 45 degrees to the left from its initial position There are 14 face images in this category.

"_2" : Head orientation change. Heads of the face images in "_1" were rotated 45 degrees to the left. There are 14 images in this category.

"_3" : Background removal. Inorder to test the effect of face background on recognition performance, background of the face images in "_0" were replaced with a constant color. There are 12 face images in this category.

"_4" : Illumination change. Face images in "_0" were darkened by 10%, inorder to test the effect of the illumination level. There are 12 face images in this category.

Besides, three face images that are given in Figure 3.4, were manually decorated with facial details such as dark glasses, masks and moustaches for two different categories, "_0" and."_1" respectively.

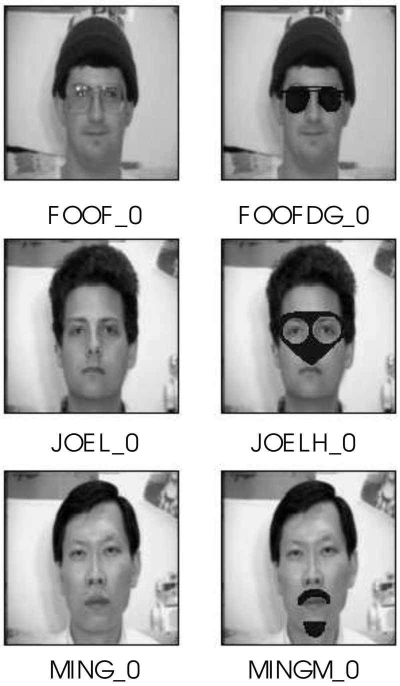

Figure 3.4. Manually decorated face images belonging to "_0" category.

Different face classes with their properties described above, are given in Table 3.1. Members of these face classes with "_1" suffix are shown in Figure 3.5.

Table 3.1. 14 different face classes used during the experiments.

Classes	Different appearances
BILLJ	_0,_1,_2,_3,_4
DAVID	_0,_1,_2,_3,_4
FOOF	_0,_1,_2,_3,_4 + (FOOFDG_0, FOOFDG_1 with dark glasses)
JOEL	_0,_1,_2,_3,_4 + (JOELH_0 , JOELH_1 with mask)
MIKE	_1,_2
MING	_0,_1,_2,_3,_4 + (MINGM_0 , MINGM_1 with moustache)
PASCAL	_0,_1,_2,_3,_4
ROBERT	_1,_2
STAN	_0,_1,_2,_3,_4
STEPHEN	_0,_1,_2,_3,_4
THAD	_0,_1,_2,_3,_4
TREVOR	_0,_1,_2,_3,_4
VMB	_0,_1,_2,_3,_4
WAVE	_0,_1,_2,_3,_4

Figure 3.5. Members of 14 face classes belonging to the "_1" category.

3.8.2 Number of Eigenfaces and Rebuild Error Ratio

Inorder to test the effect of the number of eigenfaces used in the representation of face images, a training set of 12 images from category "_0" was chosen, and each of them was rebuilt by using 5, 7, 9, and 11 eigenfaces respectively. Results are given in Table 3.2. It is clearly seen that, the rebuild error ratio decreases as the number of eigenfaces increases. When the number of eigenfaces increase, the dimension of the face space spanned by these eigenfaces increase, yielding a better representation of the face images (especially face details are represented better). As a result, the rebuild error ratio decreases.

Table 3.2. Eigenface count and rebuild error ratio relation.

Members Rebuilt	Rebuild error ratio (%) for different number of eigenfaces.			
	5	**7**	**9**	**11**
BILLJ_0	14.92	14.06	12.69	12.13
DAVID_0	14.57	13.52	13.32	12.66
FOOF_0	19.76	15.67	15.13	12.29
JOEL_0	17.41	15.65	14.23	13.84
MING_0	16.60	14.45	12.99	12.76
PASCAL_0	12.71	11.86	11.28	10.67
STAN_0	15.89	15.00	14.83	14.81
STEPHEN_0	14.41	12.39	11.08	10.71
THAD_0	15.13	13.40	13.23	13.22
TREVOR_0	14.50	10.27	10.19	9.03
VMB_0	18.75	16.80	12.00	11.69
WAVE_0	16.43	16.05	15.90	15.86
Avg.Error	15.92	14.09	13.07	12.47

3.8.3 Face Background and Rebuild Error Ratio

Face background plays an important role in the eigenfaces approach because entire information in the face image is used, without discarding any part of the image. Without face background, recognition performance should increase because, now the system merely deals with facial information. This is given in Table 3.3 as "face background and rebuild error ratio relation". It is seen that the rebuild error ratio decreases when face background is removed. A training set of

12 images was chosen from category "_3", and each of them was rebuilt by using 5, 7, 9, and 11 eigenfaces respectively.

Table 3.3. Face background and rebuild error ratio relation.

Members Rebuilt	Rebuild error ratio (%) for different number of eigenfaces. 5	7	9	11
BILLJ_3	8.03	6.80	6.73	6.56
DAVID_3	7.78	6.67	5.97	5.86
FOOF_3	9.43	9.02	8.98	8.53
JOEL_3	9.07	8.07	7.75	7.72
MING_3	9.79	9.32	8.77	8.68
PASCAL_3	9.54	8.42	8.10	8.10
STAN_3	7.23	7.21	6.80	6.71
STEPHEN_3	10.92	10.41	10.17	10.16
THAD_3	8.75	8.70	8.57	8.44
TREVOR_3	7.92	7.58	7.39	7.25
VMB_3	12.28	10.66	8.90	8.77
WAVE_3	7.37	7.16	7.14	7.13
Avg.Error	9.01	8.33	7.94	7.82

3.8.4 Training Set and Rebuild Relation

Eigenfaces are obtained from a training set and they span an M-dimensional face space. Eigenfaces represent the most variation among the training set members. Every face library member is later represented by a linear combination of these eigenfaces. Due to this fact, rebuilt face images, more or less, map to training set members. This is illustrated in Figure 3.6. A training set of eight members from category "_0" (7 eigenfaces) was chosen, and four members of category "_0" that were not in the training set, were rebuilt. It is seen that, these four members map to one of the eight members of the training set.

(a)

(b)

(c)

Figure 3.6. (a) Training set face images. (b) Face images that are not in the training set. (c) Rebuilt versions of face images in (b). They more or less map to training set members.

3.8.5 Recognition Performance and Threshold Values

In this experiment, the trade off between threshold value and correct recognition rate (for different number of eigenfaces) was tested. Inorder to achieve this, a face library that contained all members of categories "_0, _1, _2, _3", with six more face images that contained facial details of dark glasses, masks and moustaches was formed. All members of category "_1", were in the training set. Then, every member of category "_1" was searched in the face library (for 5, 7, 9, and 11 eigenfaces) under different threshold values. Recognition results are given in tables 3.4, 3.5, 3.6, and 3.7 for 5, 7, 9, and 11 eigenfaces respectively.

This experiment shows that, number of matches increase when threshold value increases. This can result in misclassifications. In Table 3.4, it is seen that, for a threshold value of 10%, 14 members were correctly classified yielding a correct classification rate of 100%. But, when threshold value was increased to 50%, 5 members were misclassified, yielding a correct classification rate of 64%.

When the number of eigenfaces involved in the recognition process increases, number of misclassifications begin to decrease. This is due to the fact that, components of the feature vectors are formed from eigenfaces. As the number of these components increases, properties of the members (especially the details) are described better. It is seen in Table 3.7 that, 14 members were classified 100% correctly for both of the threshold values 10% and 50%.

Table 3.4. Relationship between recognition performance and threshold values for 5 eigenfaces.

Members searched	Recognitions for different threshold values. 10%	20%	30%	50%
BILLJ_1	BILLJ_1	BILLJ_1	BILLJ_1	BILLJ_1 PASCAL_1
DAVID_1	DAVID_1	DAVID_1	DAVID_1	DAVID_0,_1
FOOF_1	FOOF_1 FOOFDG_1	FOOF_1 FOOFDG_1	FOOF_1 FOOFDG_1	FOOF_1 FOOFDG_1
JOEL_1	JOEL_1	JOEL_0,_1 JOELH_1	JOEL_0,_1 JOELH_1	JOEL_0,_1 JOELH_0,1
MIKE_1	MIKE_1	MIKE_1	MIKE_1	MIKE_1,_2
MING_1	MING_1 MINGM_1	MING_1 MINGM_1	MING_1 MINGM_1	MING_1 MINGM_0,_1
PASCAL_1	PASCAL_1	PASCAL_1	PASCAL_0,_1	PASCAL_0,_1 BILLJ_0,_1
ROBERT_1	ROBERT_1	ROBERT_1	ROBERT_1	ROBERT_1 PASCAL_3 STAN_3 TREVOR_3
STAN_1	STAN_1	STAN_0,_1	STAN_0,_1	STAN_0,_1,_2 WAVE_0,_1,_2
STEPHEN_1	STEPHEN_1	STEPHEN_1	STEPHEN_1	STEPHEN_1
THAD_1	THAD_1	THAD_1	THAD_1	THAD_0,_1
TREVOR_1	TREVOR_1	TREVOR_1	TREVOR_1	TREVOR_1
VMB_1	VMB_1	VMB_1	VMB_1	VMB_1
WAVE_1	WAVE_1	WAVE_0,_1	WAVE_0,_1	WAVE_0,_1,_2 STAN_0,_1,_2

Table 3.5. Relationship between recognition performance and threshold values for 7 eigenfaces.

Members searched	Recognitions for different threshold values. **10%**	**20%**	**30%**	**50%**
BILLJ_1	BILLJ_1	BILLJ_1	BILLJ_1	BILLJ_1 PASCAL_1
DAVID_1	DAVID_1	DAVID_1	DAVID_1	DAVID_0,_1
FOOF_1	FOOF_1 FOOFDG_1	FOOF_1 FOOFDG_1	FOOF_1 FOOFDG_1	FOOF_1 FOOFDG_1
JOEL_1	JOEL_1	JOEL_0,_1 JOELH_1	JOEL_0,_1 JOELH_1	JOEL_0,_1 JOELH_0,1
MIKE_1	MIKE_1	MIKE_1	MIKE_1	MIKE_1
MING_1	MING_1 MINGM_1	MING_1 MINGM_1	MING_1 MINGM_1	MING_1 MINGM_1
PASCAL_1	PASCAL_1	PASCAL_1	PASCAL_0,_1	PASCAL_0,_1 BILLJ_0,_1
ROBERT_1	ROBERT_1	ROBERT_1	ROBERT_1	ROBERT_1 STAN_3
STAN_1	STAN_1	STAN_0,_1	STAN_0,_1	STAN_0,_1,_2
STEPHEN_1	STEPHEN_1	STEPHEN_1	STEPHEN_1	STEPHEN_1
THAD_1	THAD_1	THAD_1	THAD_1	THAD_0,_1
TREVOR_1	TREVOR_1	TREVOR_1	TREVOR_1	TREVOR_1
VMB_1	VMB_1	VMB_1	VMB_1	VMB_1
WAVE_1	WAVE_1	WAVE_0,_1	WAVE_0,_1	WAVE_0,_1,_2

Table 3.6. Relationship between recognition performance and threshold values for 9 eigenfaces.

Members searched	Recognitions for different threshold values. 10%	20%	30%	50%
BILLJ_1	BILLJ_1	BILLJ_1	BILLJ_1	BILLJ_1 PASCAL_1
DAVID_1	DAVID_1	DAVID_1	DAVID_1	DAVID_0,_1
FOOF_1	FOOF_1 FOOFDG_1	FOOF_1 FOOFDG_1	FOOF_1 FOOFDG_1	FOOF_1 FOOFDG_1
JOEL_1	JOEL_1	JOEL_0,_1 JOELH_1	JOEL_0,_1 JOELH_1	JOEL_0,_1 JOELH_0,1
MIKE_1	MIKE_1	MIKE_1	MIKE_1	MIKE_1
MING_1	MING_1 MINGM_1	MING_1 MINGM_1	MING_1 MINGM_1	MING_1 MINGM_1
PASCAL_1	PASCAL_1	PASCAL_1	PASCAL_0,_1	PASCAL_0,_1 BILLJ_0,_1
ROBERT_1	ROBERT_1	ROBERT_1	ROBERT_1	ROBERT_1
STAN_1	STAN_1	STAN_0,_1	STAN_0,_1	STAN_0,_1,_2
STEPHEN_1	STEPHEN_1	STEPHEN_1	STEPHEN_1	STEPHEN_1
THAD_1	THAD_1	THAD_1	THAD_1	THAD_0,_1
TREVOR_1	TREVOR_1	TREVOR_1	TREVOR_1	TREVOR_1
VMB_1	VMB_1	VMB_1	VMB_1	VMB_1
WAVE_1	WAVE_1	WAVE_0,_1	WAVE_0,_1	WAVE_0,_1,_2

Table 3.7. Relationship between recognition performance and threshold values for 11 eigenfaces.

Members searched	Recognitions for different threshold values. 10%	20%	30%	50%
BILLJ_1	BILLJ_1	BILLJ_1	BILLJ_1	BILLJ_1
DAVID_1	DAVID_1	DAVID_1	DAVID_1	DAVID_0,_1
FOOF_1	FOOF_1 FOOFDG_1	FOOF_1 FOOFDG_1	FOOF_1 FOOFDG_1	FOOF_1 FOOFDG_1
JOEL_1	JOEL_1	JOEL_0,_1 JOELH_1	JOEL_0,_1 JOELH_1	JOEL_0,_1 JOELH_0,1
MIKE_1	MIKE_1	MIKE_1	MIKE_1	MIKE_1
MING_1	MING_1 MINGM_1	MING_1 MINGM_1	MING_1 MINGM_1	MING_1 MINGM_1
PASCAL_1	PASCAL_1	PASCAL_1	PASCAL_0,_1	PASCAL_0,_1
ROBERT_1	ROBERT_1	ROBERT_1	ROBERT_1	ROBERT_1
STAN_1	STAN_1	STAN_0,_1	STAN_0,_1	STAN_0,_1,_2
STEPHEN_1	STEPHEN_1	STEPHEN_1	STEPHEN_1	STEPHEN_1
THAD_1	THAD_1	THAD_1	THAD_1	THAD_0,_1
TREVOR_1	TREVOR_1	TREVOR_1	TREVOR_1	TREVOR_1
VMB_1	VMB_1	VMB_1	VMB_1	VMB_1
WAVE_1	WAVE_1	WAVE_0,_1	WAVE_0,_1	WAVE_0,_1,_2

3.8.6 Recognition Performance and Illumination

In this experiment, the effect of illumination level on recognition performance was tested. Inorder to achieve this, a face library that contained all members of categories "_0, _1, _3" was formed. A training set from all members of category "_0" (7 eigenfaces) was chosen. Later, a library search for all members of category "_4" was performed. As you will recall, members of category "_4", were obtained synthetically from the members of category "_0", by subtracting 10% of the original pixel values.

Recognition results are given in Table 3.8. It is seen that every member of category "_4" was classified as "unknown" for a threshold value of 10%. When the threshold value was increased to 45%, every member of category "_4" was classified as "known". Due to the increase in the threshold value, one member was misclassified yielding a correct classification rate of 92%.

Table 3.8. Relationship between recognition performance and illumination level. "?" shows a miss.

Members searched	Recognitions for different threshold values. **10%**	**20%**	**35%**	**45%**
BILLJ_4	?	?	BILLJ_0	BILLJ_0
DAVID_4	?	?	DAVID_0	DAVID_0,_1
FOOF_4	?	?	?	FOOF_0
JOEL_4	?	?	JOEL_0	JOEL_0,_1
MING_4	?	?	MING_0	MING_0
PASCAL_4	?	?	PASCAL_0	PASCAL_0,_1 BILLJ_0
STAN_4	?	STAN_0,_1	STAN_0,_1	STAN_0,_1
STEPHEN_4	?	?	STEPHEN_0	STEPHEN_0
THAD_4	?	?	THAD_0	THAD_0,_1
TREVOR_4	?	?	?	TREVOR_0,_1
VMB_4	?	?	?	VMB_0
WAVE_4	?	WAVE_0	WAVE_0,_1	WAVE_0,_1

3.8.7 Recognition Performance and Light Source Position

In this experiment, the effect of light source position on recognition performance was tested. Inorder to achieve this, a face library that contained all members of categories "_0, _3" was formed. A training set from all members of category "_0" (7 eigenfaces) was chosen. Later, a library search for every member of category "_1" was performed. As you will recall, members of category "_1" were obtained under a light source that was moved 45 degrees to the left, which was at its original position in category "_0".

Recognition results are given in Table 3.9. For a threshold value of 10%, a correct classification rate of 8% was achieved. 92% of the face images were classified as "unknown". When the threshold value was increased to 45%, a correct classification rate of 58% was achieved. 17% of the face images were misclassified and 25% of them were classified as "unknown".

3.8.8 Recognition Performance and Head Orientation

In this experiment, the effect of head orientation on recognition performance was tested. Inorder to achieve this, a face library that contained all members of categories "_0, _1, _3" was formed. Then, a training set from all members of category "_1" (7 eigenfaces) was chosen. Later, a library search was performed for every member of category "_2". As you will recall, members of category "_2" were obtained from the members of category "_1", by rotating their heads 45 degrees to the left.

Recognition results are given in Table 3.10. For a threshold value of 10%, every member of category "_2" was classified as unknown. A correct classification rate of 14% was achieved for a threshold value of 45%. 86% of the members were classified as "unknown" even under this threshold value.

Table 3.9. Relationship between recognition performance and light source position. "?" shows a miss.

Members searched	Recognitions for different threshold values. 10%	20%	35%	45%
BILLJ_1	?	?	?	BILLJ_0 PASCAL_0
DAVID_1	?	DAVID_0	DAVID_0	DAVID_0,_3
FOOF_1	?	?	?	?
JOEL_1	?	?	JOEL_0	JOEL_0 MING_0 THAD_3
MING_1	?	?	MING_0	MING_0
PASCAL_1	?	PASCAL_0	PASCAL_0	PASCAL_0
STAN_1	STAN_0	STAN_0	STAN_0	STAN_0
STEPHEN_1	?	?	?	?
THAD_1	?	?	?	THAD_0
TREVOR_1	?	TREVOR_0	TREVOR_0	TREVOR_0
VMB_1	?	?	?	?
WAVE_1	?	WAVE_0	WAVE_0	WAVE_0

Table 3.10. Relationship between recognition performance and head orientation. "?" shows a miss.

Members searched	Recognitions for different threshold values. 10%	20%	35%	45%
BILLJ_2	?	?	?	?
DAVID_2	?	?	?	?
FOOF_2	?	?	?	?
JOEL_2	?	?	?	?
MIKE_2	?	?	?	?
MING_2	?	?	?	?
PASCAL_2	?	?	?	?
ROBERT_2	?	?	?	?
STAN_2	?	?	STAN_1	STAN_0,_1
STEPHEN_2	?	?	?	?
THAD_2	?	?	?	?
TREVOR_2	?	?	?	?
VMB_2	?	?	?	?
WAVE_2	?	?	?	WAVE_0,_1

3.8.9 Recognition Performance and Presence of Details

In this experiment, the effect of the presence of facial details such as dark glasses, masks and moustaches on recognition performance was tested. Inorder to achieve this, a face library that contained all members of categories "_0, _1, _3" was formed. Then, a training set from all members of category "_0" (7 eigenfaces) was chosen.

Recognition results are given in Table 3.11. For a threshold value of 30%, a correct classification rate of 100% was achieved. In Figure 3.7, the rebuilt versions of these manually decorated face images are seen. Small changes in facial appearance does not cause a major problem to the proposed system.

Table 3.11. Relationship between recognition performance and presence of details. "?" shows a miss.

Members searched	Recognitions for different threshold values. **10%**	**20%**	**30%**
FOOFDG_0	?	FOOF_0	FOOF_0
FOOFDG_1	?	?	FOOF_1
JOELH_0	?	JOEL_0	JOEL_0,_1
JOELH_1	?	JOEL_1	JOEL_0,_1
MINGM_0	MING_0	MING_0	MING_0
MINGM_1	MING_1	MING_1	MING_1

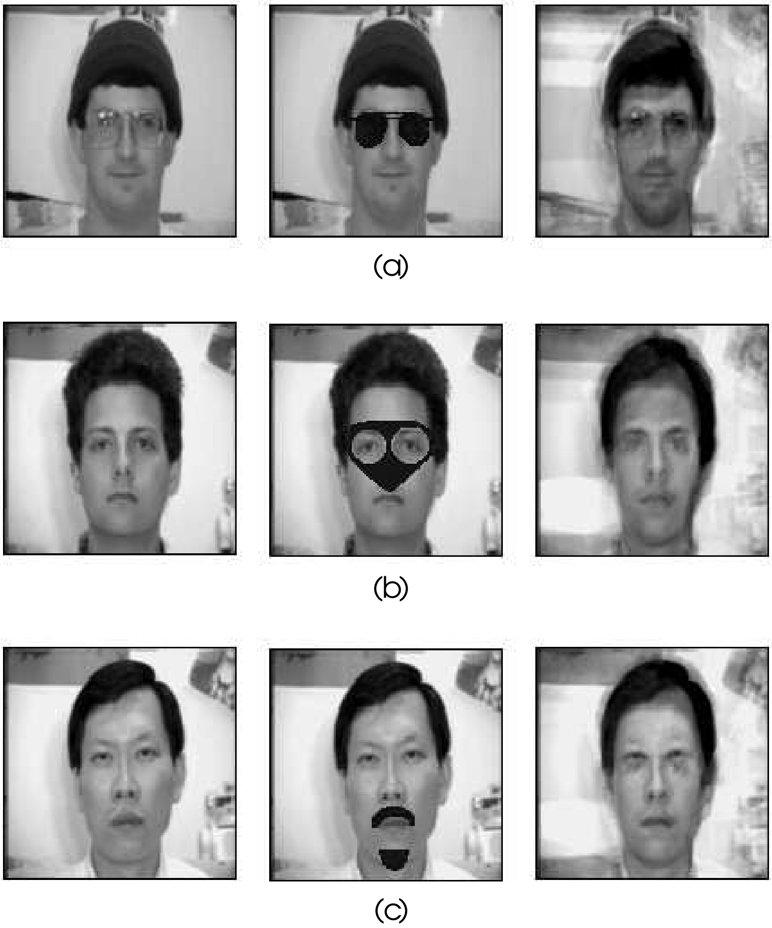

Figure 3.7. Original, manually decorated and rebuilt versions of (a) FOOF_0, (b) JOEL_0 and (c) MING_0.

3.8.10 Recognition Speed

During the software implementation of the eigenfaces approach, no optimizations were performed on matrix operations. Experiments were performed on an INTEL 80486 based personal computer. During these tests, eigenface formations and feature vector updates formed the most time consuming parts of the face recognition process.

After obtaining feature vectors, recognitions were performed in near real time, in less than one second. Sample update times are given in Table 3.12.

Table 3.12. Update speed of the proposed face recognition system over a 58 member face library. Recognitions are done in less than one second.

Eigenface Count	Total time elapsed (seconds) whlle updating all library members.
5	50
7	61
9	73
11	84

CHAPTER 4

FACE-PRO

This chapter is dedicated to the fundamental properties of the face recognition software, that was developed to demonstrate the proposed face recognition approach.

4.1 FACE-PRO GUI and the IM-PRO Connection

Inorder to develop a user friendly interface to the proposed face recognition software - "FACE-PRO", it was suitable to inherit the graphical user interface (GUI) of "IM-PRO", which was developed in one of our a previous studies [32] on image processing. We might have as well chosen to use a more popular software development platform such as "Microsoft WINDOWS" but instead, we have preferred to use a more compact and less hardware dependent environment on which we have gained more experience from our previous studies. Besides, it was an opportunity for us to test the IM-PRO GUI to its limits on which FACE-PRO was built.

4.2 FACE-PRO System Requirements

FACE-PRO is an MS-DOS based application which runs on INTEL 80x86 based personal computers. Minimum hardware and operating system requirements inorder to run FACE-PRO efficiently is given below:

- ***Operating System*** : MS-DOS Ver 4.0,
- ***Base Memory*** : 542 KB,
- ***Expanded Memory*** : 1024 KB,

- ***Video Adapter*** : VESA Compatible, 512 KB SVGA card,
- ***Keyboard*** : AT 101/102 Keys,
- ***Pointing Device*** : Microsoft Mouse Version 6.5 or compatible,
- ***CPU*** : Intel 80386,
- ***Math Coprocessor*** : Intel 80387.

4.3 FACE-PRO Installation

FACE-PRO modules were compressed and stored on two 3.5" HD floppy disks. Inorder to simplify the installation process, install disk #1 contains a shareware installation program that guides the user through installation steps.

Installation starts by simply typing A:\INSTALL on the MS-DOS command prompt. After that, installation program asks for source and destination paths, where FACE-PRO modules are going to be read from and written to respectively.

FACE-PRO package contains four major parts that are given below:

- Executable modules,
- Source modules,
- Sample face files,
- Utilities.

Installation program lets the user to install all or some of these major parts. By default, a full installation is performed. An indicator whether the associated part has been chosen or not, together with the hard disk space it occupies is displayed. It is in users responsibility to provide enough free hard disk space.

4.4 Configuring FACE-PRO

FACE-PRO installation program just decompresses software files under custom defined directories. It does not do any modifications to AUTOEXEC.BAT and CONFIG.SYS system files. Copying merely software files is not enough for FACE-PRO to run properly.

FACE-PRO needs some device drivers to be installed inorder to make use of system resources such as expanded memory or mouse. The device drivers that are needed to run FACE-PRO properly, together with their purposes are given below:

- **An expanded memory manager.** Conventional MS-DOS memory is not enough for FACE-PRO while performing face recognition tasks. FACE-PRO makes use of extended memory via expanded memory. Inorder to use expanded memory, an expanded memory manager such as EMM386 or QEMM386 must be installed before running FACE-PRO. During the beta tests, it has been noticed that FACE-PRO achieved a better throughput while QEMM386 was running.

- **A mouse driver.** Although FACE-PRO GUI can be used by keyboard, it is recommended that a Microsoft Mouse Version 6.5 compatible mouse with its driver is installed. This makes FACE-PRO to be used much more efficiently.

- **A sotware disk cache.** After adding new members to FACE-PRO face library disk access can become a real bottle neck. Minimizing disk accesses significantly improves FACE-PRO performance. The use of a software disk caching utility such as SMARTDRV (with 2 MB reserved memory) is highly recommended. SMARTDRV has a write-behind cache. Any information that is not saved to disk can be lost during a power down. It is recommended that SMARTDRV is installed merely for read operations.

4.5 Trouble Shooting FACE-PRO

After making sure that the minimum system requirements for FACE-PRO, together with its essential device drivers are met, then no problems should occur while running FACE-PRO. But, sometimes things go wrong. Most common problems that may occur while running FACE-PRO are given below.

- **"Error reading start-up configuration file"** message. FACE-PRO looks for some configuration files and run-time modules in the current working directory during start-up.

If the current working directory is not the FACE-PRO start-up directory, then this message is displayed because, the loader is unable to locate the necessary start-up files. FACE-PRO should always be invoked from its start-up directory which was specified as the destination path during installation.

- System checks are performed and the program does not run. By default, FACE-PRO loader first checks the system configuration. If the minimum program requirements are not satisfied then, the loader refuses to start FACE-PRO. After that, the integrity of some vital run-time modules are tested by their file size, CRC, date and time attributes. If any of these modules fail to pass the test, the loader again refuses to start the program. This can happen especially after recompiling FACE-PRO modules. If so, then run UPDATE.EXE utility in the FACE-PRO start-up directory to restore file attributes. FACE-PRO loader marks the offending components with a cross sign - "X", during both of these tests. Although it is not recommended, system checks can be turned off in FACE-PRO.

- **Screen gets messy while FACE-PRO is running.** This is due to an incompatible SVGA card. For the sake of speed, FACE-PRO makes direct screen access via VESA calls. Those SVGA cards, which are not 100% VESA compatible can cause such problems. During the beta tests, FACE-PRO runned smoothly on most of the machines but, especially on those machines that had S3 incorporated SVGA cards, such problems had occured. Try to install the UNIVESA.EXE - shareware VESA driver, that comes with the utilities, which seems to be helpful in overcoming that sort of problems.

4.6 FACE-PRO Executable Modules

This section is dedicated to the executable modules of FACE-PRO with their run-time directory structures.

Installation program creates the following directories that are involved during a face recognition process:

- **\FACEPRO:** This is the FACE-PRO start-up directory. Executable modules with their run-time libraries reside here. There exists a "*read.me*" file in this directory, which gives last minute information about the software package.

- **\FACEPRO\FACELIB:** Face library members are represented by two entries in two different directories. One entry corresponds to the actual uncompressed face data, that resides in this directory. Every face image automatically receives an identification number and the ".FLB" extention when it is added to the face library. Figure 4.1 shows the structure of FACE-PRO face files. FACE-PRO comes with a pre-installed face library that contains 14 members.

- **\FACEPRO\WEIGHTS:** The other entry corresponds to the feature vectors of face images which are also called weight vectors. Weight vectors, together with a brief description of associated face images are stored in this directory with the same identification number as the face data, but with the ".WLB" extention instead. Weight vectors are initially empty. They are updated when a training set is chosen and eigenfaces are calculated. Figure 4.2 shows the structure of FACE-PRO weight files.

- **\FACEPRO\EIGENS:** This directory is initially empty. When a training set is chosen, eigenfaces and the average image of the ensemble are created in this directory. Average image receives the ".AVG" extention, where as eigenfaces receive the ".EIG" extention. The structure of these files are shown in Figure 4.3.

FACE-PRO signature	CRLF
Time stamp	CRLF
16 KB binary, raw face data	

Figure 4.1. Format of a FACE-PRO face file that resides in the FACEPRO\FACELIB directory. 16K raw data corresponds to an image size of 128 x 128 x 8.

FACE-PRO signature	CRLF
Time stamp	CRLF
Entry id	CRLF
Entry information	CRLF
Options	CRLF
Weight vector in binary form	

Figure 4.2. Format of a FACE-PRO weight file that resides in the FACEPRO\WEIGHTS directory. Entry id and entry information corresponds to user defined short and long descriptions of the face image respectively. Options define the pre-processings applied. The components of the weight vector as 4 byte float entries are formed after a training set has been chosen.

FACE-PRO signature	CRLF
Time stamp	CRLF
64 KB binary information	

Figure 4.3. Format of a FACE-PRO eigenface file that resides in the FACEPRO\EIGENS directory. The same format represents the average face image as well. Gray level values of these images are represented by 4 byte float entries, yielding a total of 64K binary information.

FACE-PRO executable modules are comprised of from the following files that reside in the \FACEPRO start-up directory:

- **FACEPRO.EXE:** Executable version of FACE-PRO.
- **FACEPRO.INI:** FACE-PRO ini file. It contains vital information such as start-up directories, number of face library entries that are used during program execution. Format of the FACE-PRO ini file is given in Figure 4-4.
- **SVGA256.BGI:** SVGA device driver that is used for screen access.
- **MENU.ILB:** FACE-PRO menu file.
- **HELP.ILB:** FACE-PRO online help file.
- **CURSOR.ILB:** Mouse cursor definitions.
- **ICON.ILB:** Icon definitions.
- **WALL.ILB:** Wall papers.
- **FACEPRO.PIC:** FACE-PRO start-up picture file.
- **ITULOGO.PIC:** Picture file displayed on exit.

- **PHOTO.PIC:** Picture file of the FACE-PRO programmer.
- **COLOR.PAL:** Color pallette.
- **OLD.PAL:** Previous version of COLOR.PAL.
- **FILES.CHK:** Start-up system check log.
- **FILES.OLD:** Previous version of FILES.CHK.
- **UPDATE.EXE:** Recalculates the information that is used in testing the integrities of system files. Inother words, it brings FILES.CHK up to date.
- ***.CFG:** Various desktop configuration files.

FACE-PRO signature	CRLF
Time stamp	CRLF
Path to start-up directory	CRLF
Path to face files	CRLF
Path to weight files	CRLF
Path to eigenfaces	CRLF
Current pre-processing options	CRLF
Total number of face library members	CRLF
Id of last face library member	CRLF
Training set size	CRLF
Number of eigenfaces	CRLF
User defined hit threshold	CRLF
IDs of the training set members	CRLF

Figure 4.4. Format of the FACE-PRO ini file that resides in the FACEPRO start-up directory.

4.7 FACE-PRO Source Modules

FACE-PRO was written mostly in C (screen update routines and interrupt handlers were written in assembly language for the sake of speed). And then, it was compiled under the huge memory model from the project file "FACEPRO.PRJ" by using BORLAND C/C++ 3.1 and Turbo Assembler 3.1. FACE-PRO source modules with their header files, reside in the **\FACEPRO\SOURCES** directory.

FACE-PRO GUI and the system kernel were inherited from IM-PRO [32], so they are not going to be covered here, once more. Instead, FACE-PRO source modules that were developed purely for the implementation of the proposed face recognition system will be presented briefly:

- **FACEPRO PRJ:** BORLAND C/C++ 3.1 project file where FACEPRO.EXE was compiled from. Contents of the project file are shown in Figure 4.5.

- **HIPS.C:** This module contains routines for reading and writing HIPS formatted image files

- **EIGENS.C:** Routines for the estimation of the eigenvalues and the eigenvectors of an N x N symmetric real matrix reside in this module.

- **PREPROC.C:** Routines for pre-processing operations reside in this module At the moment, only the image size normalization and the histogram equalization algorithms are available. An algorithm for face background removal was intended to be implemented but, due to time limitations it was left for future work.

- **ALGO.C:** Algorithms of the proposed face recognition method were implemented in this module. Routines for the calculation of eigenfaces, update and comparison of weight vectors all reside here.

- **FACEPRO?.C:** Modules that were written for FACE-PRO specific operations whose properties are given below:

  FACEPRO0.C: FACE-PRO start-up routines.
  FACEPRO1.C: File and directory management routines.
  FACEPRO2.C: Face library management routines.
  FACEPRO3.C: Face library management routines.
  FACEPRO4.C: Routines for training set formation.
  FACEPRO5.C: Routines for query management and rebuild operations.
  FACEPRO6.C: Routines for the display of library members and eigenfaces.

- **MAIN.C:** FACE-PRO main menu.

```
TRIP   .OBJ    IMPRO001.C    HIPS    .C
SMALL  .OBJ    IMPRO002.C    EIGENS  .C
CPULOW .ASM    IMPRO003.C    PREPROC .C
VIDLOW .ASM    IMPRO004.C    ALGO    .C
SPEEDY .ASM    IMPRO005.C    FACEPRO0.C
C_ERROR.ASM    IMPRO006.C    FACEPRO1.C
D_LIGHT.ASM    IMPRO007.C    FACEPRO2.C
GLOBALS.C      IMPRO008.C    FACEPRO3.C
CMOS   .C      IMPRO010.C    FACEPRO4.C
MEMORY .C                    FACEPRO5.C
SYSTEM .C                    FACEPRO6.C
TEXT   .C                    MAIN.C
VIDEO  .C
MOUSE  .C
KERNEL .C
```

Figure 4.5. Contents of FACEPRO.PRJ project file.

4.8 FACE-PRO Sample Face Files

FACE-PRO comes with 70 HIPS formatted sample face files that can be used for demonstration purposes. The **\FACEPRO\HIPS** directory contains these face files. These sample face files form 14 different face classes whose properties are given in Table 3.1. There exists a "*read.me*" file in this directory, which also gives the descriptions of these face files.

4.9 FACE-PRO Utilities

Due to the image processing characteristics of the software package, the **\FACEPRO\UTILS** directory contains the following utility programs that can expand the benefits of FACE-PRO:

- **VIEWER.EXE:** A viewer for 8-bit HIPS formatted image files.
- **RAW2HIP.EXE:** Converts RAW formatted 8-bit image files to HIPS formatted ones.
- **HIP2RAW.EXE:** Converts HIPS formatted 8-bit image files to RAW formatted ones.

- **STRETCH.EXE:** Changes the image size of a HIPS formatted 8-bit image file.
- **LIGHT.EXE:** Increases or decreases the gray level values of a HIPS formatted 8-bit image by a given percent.
- **UNIVESA.EXE:** A VESA driver which helps a lot with incompatible SVGA cards.

For more information about these utilities, refer to the documentations in the \FACEPRO\UTILS directory.

4.10 Navigating Through FACE-PRO Menus

This section contains a brief description of FACE-PRO menus. FACE-PRO main menu consists of eight major parts. It is possible to choose a menu item both by clicking the left mouse button or by pressing its hot key combination. Also, a "tool bar" resides at the bottom of the screen which enables the user to choose most frequently used menu entries by clicking on their icons. In Figure 4.6, it is seen that FACE-PRO screen always contains three major areas as; the main menu, the client area and the tool bar. Descriptions of FACE-PRO main menu entries are given below:

- **\ABOUT:** Displays information about the FACE-PRO programmer.
- **\FILES:** It contains three sub-menu entries for operating system functions such as transferring control to an other MS-DOS program or quitting FACE-PRO.
- **\LIBRARY:** Face library operations are performed via the two sub-menu entries of this main menu entry.

  \LIBRARY\Add item: Lets the user to choose and add a HIPS formatted face file located any where on the disks, to the face library. User can specify short and long descriptions of the face file, as well as the pre-processing options.

It is not possible to use this menu entry after a training set has been chosen. Instead, the \QUERY\Search menu should be used inorder to verify whether the specified face image already exists in the face library or not.

\LIBRARY\Tools: By using this menu, it is possible to view or delete face library members. Multiple entries can be chosen by double clicking on the desired entries in the list box. It is not possible to delete library members that are also in the training set for integrity purposes. A sample face library member is seen in Figure 4.7.

- **\TSET:** Training set operations are performed via the three sub-menu entries of this main menu entry.

\TSET\Choose training set. As its name implies, this menu entry is for choosing face library members that are going to be in the training set. Maximum number of training set members as well as the maximum number of eigenfaces are shown at the bottom of the list box. FACE-PRO is quite scaleable so that, these numbers differ according to the available expanded memory. The number of eigenfaces that are going to be calculated from the training set can be chosen by a variety of ways. The maximum number of eigenfaces can not exceed the training set size. After choosing training set members, eigenfaces are calculated and then, weight vectors of the face library members are updated according to these new set of eigenfaces.

\TSET\Display training set. Training set members (if any) are displayed.

\TSET\Display eigenfaces. Eigenfaces that are calculated from the training set, together with the average face image of the ensemble are displayed.

- **\QUERY:** After a training set has been chosen, recognitions are performed via the four sub-menu entries of this main menu entry.

\QUERY\Search. Face library is searched for a given HIPS formatted image file located any where on the disks, under a user defined hit threshold. Acquired image can be pre-processed according to user specifications. If the face image does not exist in the face library, then it can be added to the face library. If it exists, then the matching face library entries are displayed.

\QUERY\Rebuild. Desired face library entries can be approximately rebuilt by using the average and the eigenfaces. Error ratio between the original and the rebuilt image is shown, as well as the images themselves.

\QUERY\Information. Displays information about face recognition statistics such as hit and miss ratios.

\QUERY\Options. User defined hit threshold is specified by using this menu entry. Library searches are performed within this threshold.

- **OPTIONS:** This main menu entry contains various sub-menu entries for desktop configurations and start-up and shut down procedures.
- **HELP:** Main menu entry of the on-line help.
- **EXIT:** Quits FACE-PRO by various ways.

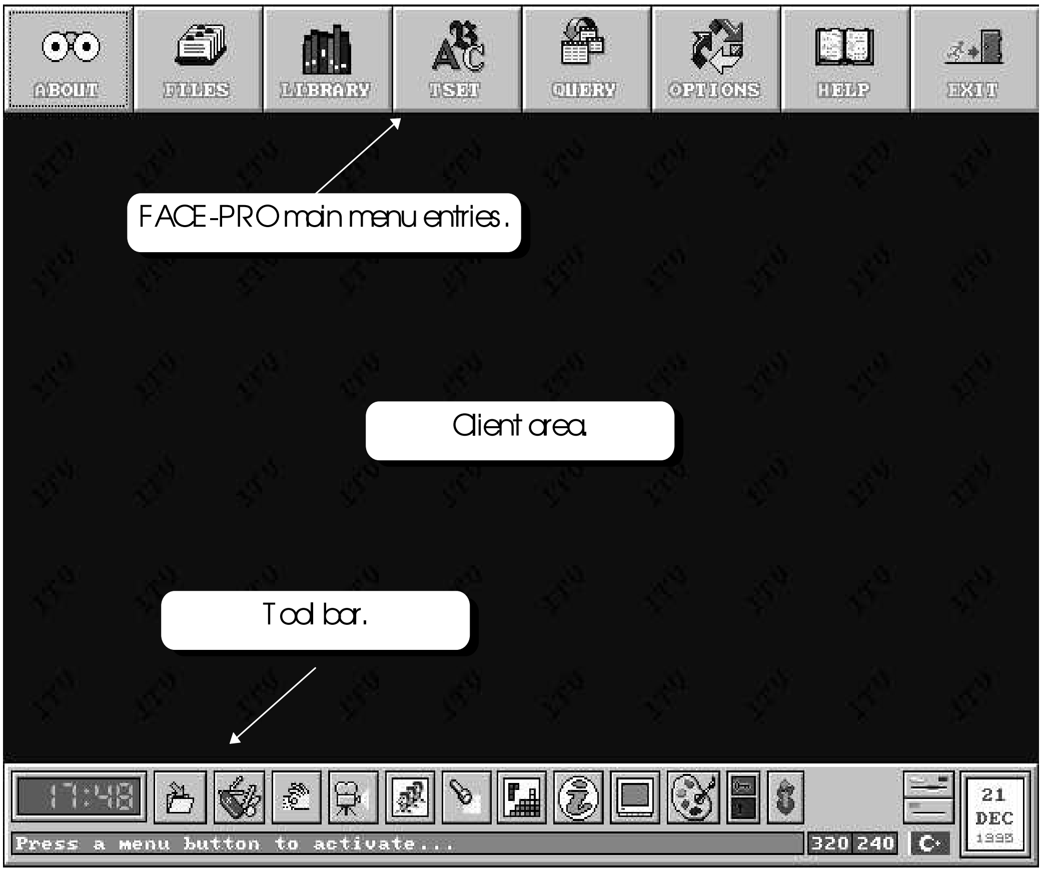

Figure 4.6. FACE-PRO main screen with main menu entries, client area and tool bar.

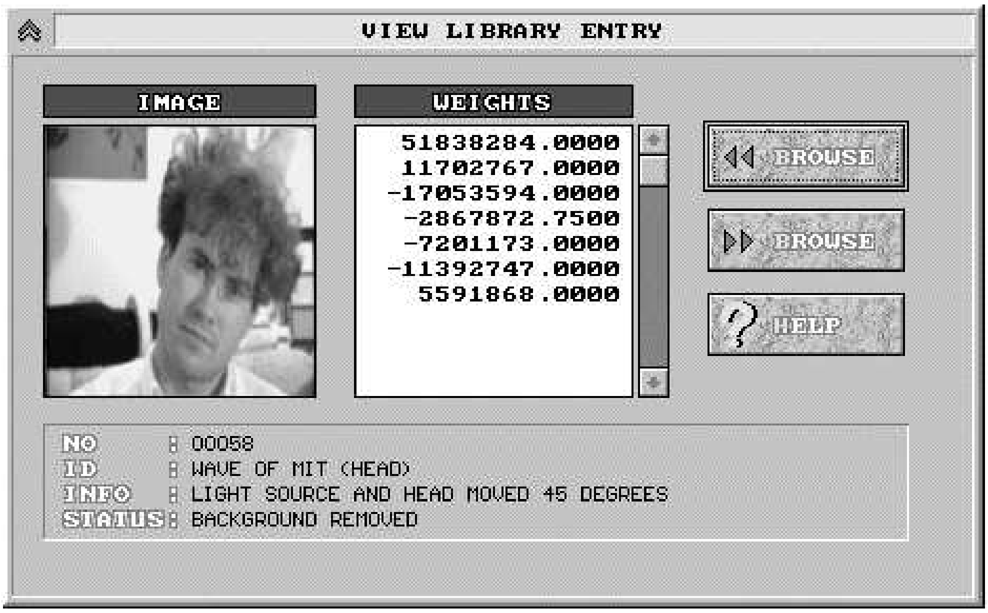

Figure 4.7. A sample face library entry.

CHAPTER 5

Conclusions and Directions For Future Research

This chapter summarizes the research done in this thesis, together with directions for future work.

5.1 Conclusions

In this research, two major approaches to the face recognition problem have been studied and a face recognition system based on the eigenfaces approach was proposed. Major properties of these two approaches are:

- **Feature based face recognition** makes use of the individual properties of the organs that are found on a face such as eyes, mouth and nose as well as their relationships with each other. Most common way of evaluating these features is the use of deformable templates and active contour models. Facial features are located firstly by a rough contour estimation method, and then by minimizing some energy function, exact locations are extracted. The basic characteristic of this approach is its dependency on extensive geometry.

- **Principal component analysis**, approaches to the face recognition problem by means of information theory concepts. The most relevant information that is contained in a face image is extracted. Eigenfaces method is a principal component analysis approach, where the eigenvectors of the covariance matrix of a small set of characteristic pictures are sought. These eigenvectors are called eigenfaces due to their resemblence of face images. Recognition is performed by assigning weight vectors to face images, according to their contributions to the face space spanned by the eigenfaces. This approach excels in its speed, simplicity and learning capability.

A robust face recognition system should be insensitive to

- Changes in illumination,
- Changes in head orientation and scale,
- Presence of facial details such as glasses, beards,
- Face background.

Experimental results have shown that, the proposed face recognition method was very sensitive to face background and head orientations. Changes in the illumination did not cause a major problem to the system. Besides, presence of small details such as dark glasses or masks were too far from being a real challenge to the system. There exists a trade off between the correct recognition rate and the threshold value. As the threshold value increases, number of misses begin to decrease, possibly resulting in misclassifications. On the contrary, when the number of eigenfaces involved in the recognition process increases, misclassification rate begins to decrease, possibly resulting in misses.

5.2 Directions for Future Work

Some topics that may help extend the work in this thesis are:

- **Implementation of a background removal algorithm.** Inorder to minimize the effects of face background and head orientation on the recognition performance, background of face images should be removed and heads should be normalized both in scale and orientation.

- **Recognition from multiple views.** The current recognition system has been designed for frontal views of face images. A neural network architecture (may be together with a feature based approach) can be implemented in which the orientation of the face is first determined, and then the most suitable recognition method is selected.

- **Scanner and camera support.** The current recognition system acquires face images only from face files located on magnetic mediums. Camera and scanner support should be implemented for greater flexibility.

- **Migration to client/server architecture.** Actual system was designed for single users, running under MS-DOS. But, in practice a face database should be located on a file server that is independent of the clients. Clients should perform queries on feature vectors stored on the file server, reducing storage requirements and gaining more flexibility.

References

[1] Goldstein, A. J., Harmon, L. D., and Lesk, A. B., Identification of human faces", Proc. IEEE 59, pp. 748-760, (1971).

[2] Haig, N. K., "How faces differ - a new comparative technique", Perception 14, pp. 601-615, (1985).

[3] Rhodes, G., "Looking at faces: First-order and second order features as determinants of facial appearance", Perception 17, pp. 43-63, (1988).

[4] Kirby, M., and Sirovich, L., "Application of the Karhunen-Loeve procedure for the characterization of human faces", IEEE PAMI, Vol. 12, pp. 103-108, (1990).

[5] Sirovich, L., and Kirby, M., "Low-dimensional procedure for the characterization of human faces", J. Opt. Soc. Am. A, 4, 3, pp. 519-524, (1987).

[6] Terzopoulos, D., and Waters, K., "Analysis of facial images using physical and anatomical models", Proc. 3rd Int. Conf. on Computer Vision, pp. 727-732, (1990).

[7] Manjunath, B. S., Chellappa, R., and Malsburg, C., "A feature based approach to face recognition", Trans. of IEEE, pp. 373-378, (1992).

[8] Harmon, L. D., and Hunt, W. F., "Automatic recognition of human face profiles", Computer Graphics and Image Processing, Vol. 6, pp. 135-156, (1977).

[9] Harmon, L. D., Khan, M. K., Lasch, R., and Ramig, P. F., "Machine identification of human faces", Pattern Recognition, Vol. 13(2), pp. 97-110, (1981).

[10] Kaufman, G. J., and Breeding, K. J, "The automatic recognition of human faces from profile silhouettes", IEEE Trans. Syst. Man Cybern., Vol. 6, pp. 113-120, (1976).

[11] Wu, C. J., and Huang, J. S., "Human face profile recognition by computer", Pattern Recognition, Vol. 23(3/4), pp. 255-259, (1990).

[12] Kerin, M. A., and Stonham, T. J., "Face recognition using a digital neural network with self-organizing capabilities", Proc. 10th Int. Conf. on Pattern Recognition, pp.738-741, (1990).

[13] Nakamura, O., Mathur, S., and Minami, T., "Identification of human faces based on isodensity maps", Pattern Recognition, Vol. 24(3), pp. 263-272, (1991).

[14] Turk, M., and Pentland, A., "Eigenfaces for recognition", Journal of Cognitive Neuroscience, Vol. 3, pp. 71-86, (1991).

[15] Yuille, A. L., Cohen, D. S., and Hallinan, P. W., "Feature extraction from faces using deformable templates", Proc. of CVPR, (1989).

[16] Gonzalez, R. C., and Tou, J. T., "Pattern recognition principles", Addison-Wesley Publishing Company, (1974).

[17] Carey, S., and Diamond, R., "From piecemeal to configurational representation of faces", Science 195, pp. 312-313, (1977).

[18] Bledsoe, W. W., "The model method in facial recognition", Panoramic Research Inc. Palo Alto, CA, Rep. PRI:15, (August 1966).

[19] Bledsoe, W. W., "Man-machine facial recognition", Panoramic Research Inc. Palo Alto, CA, Rep. PRI:22, (August 1966).

[20] Fischler, M. A., and Elschlager, R. A., "The representation and matching of pictorial structures", IEEE Trans. on Computers, c-22.1, (1973).

[21] Kohonen, T., "Self-organization and associative memory", Berlin: Springer-Verlag, (1989).

[22] Kohonen, T., and Lehtio, P., "Storage and processing of information in distributed associative memory systems", (1981).

[23] Fleming, M., and Cottrell, G., "Categorization of faces using unsupervised feature extraction", Proc. of IJCNN, Vol. 90(2), (1990).

[24] Kanade, T., "Picture processing system by computer complex and recognition of human faces", Dept. of Information Science, Kyoto University, (1973).

[25] Burt, P., "Smart sensing within a Pyramid Vision Machine", Proc. of IEEE, Vol. 76(8), pp. 139-153, (1988).

[26] Ching, C. W., and Huang, L. C., "Human face recognition from a single front view", Int. J. of Pattern Recognition and Artificial Intelligence, Vol. 6(4), pp. 570-593, (1992).

[27] Berger, M. O., and Mohr, R., "Towards autonomy in active contour models", Proc. 10th Int. Conf. on Pattern Recognition, pp. 847-851, (1990).

[28] Kass, M., Witkin, A., and Terzopoulos, D., "Snakes: Active contour models", Int. J. of Computer Vision, pp. 321-331, (1987).

[29] Williams, D. J., and Shah, M., "A fast algorithm for active contours", Proc. 3rd Int. Conf. on Computer Vision, pp. 592-595, (1990).

[30] Amir, A. A., Weymouth, T. E., and Jain, R. C., "Using dynamic programming for solving variational problems in vision", IEEE PAMI, Vol. 12(9), pp. 855-867, (1990).

[31] Press, W. H., Flannery, B. P., Teukolsky, S. A., Vetterling, W. T., "Numerical recipes in C", Cambridge University Press, pp. 353-380, (1988).

[32] Atalay, I., and Ballikaya, F.,"Development of an Image Processing Environment for the 80x86 Platform", B.Sc. Thesis, Istanbul Technical University, (1993).

Biography

Ilker Atalay was born in Istanbul, Turkey, in 1971. He completed his high school education in English High School for Boys (N.A.L.), Istanbul, in 1989, and received his B.Sc. degree in control and computer engineering from Istanbul Technical University in 1993.