ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

FACE RECOGNITION USING EIGENHILLS

M.Sc. Thesis by
Alper YILMAZ, M.Sc.

504970629011

Date of submission : September 1999

Date of defense examination : November 1999

Supervisor (Chairman): Assoc. Prof. Dr. Muhittin GÖKMEN

Members of the Examining Committee: Prof. Dr. Aytül ERÇİL (B.Ü.)

Assoc. Prof. Dr. Işıl CELASUN

NOVEMBER 1999

PREFACE

This study is a result of cooperative work with my adviser Assoc. Prof. Dr. Muhittin Gökmen, I am grateful to him for his assistance on preparing this project. Special thanks to Binnur Kurt, who is a research assistant in *ITU CVIP** Lab. for his efforts on preparing the edge detection phase of the program to test the study's performance.

I am thankful to my wife, Selen, for her kindness, assistance and endurance to me throughout the study. She was the one who obtained the Purdue Face Database of 2 CD's from Internet resources.

The early results of this study was printed in *IEEE National Conference on Signal Processing and Applications*, 1999, Ankara, Turkey and *International Conference on Computer and Information Sciences*, 1999, Kuşadası, Turkey. The whole study is submitted to *Pattern Recognition Journal* in September 1999 and *International Conference on Pattern Recognition'2000.*

September, 1999 Alper YILMAZ

---

* Istanbul Technical University, Computer Vision & Image Processing Laboratory

CONTENTS

ABBREVIATIONS

PCA : Principal component analysis

KLT : Karhunen-Loeve transform

ANN : Artificial neural network

AAM : Auto-associative memory

LAAM : Linear auto-associative memory

dim : Dimension

Pr : Probability

MSD : Mean square distance

GED : Generalized edge detector

RAM : Random access memory

CVC : Computer vision center

IE : Ideal edges

DE : Detected edges

TABLE LIST

FIGURE LIST

SYMBOL LIST

w	: width of image
h	: height of image
o_j	: output of j^{th} neuron, reconstructed face
w_{ij}	: intensity of the connection between the i^{th}, j^{th} neurons (weight)
$\mathbf{W}$	: weight vector
η	: learning constant
t	: iteration index
x_j	: j^{th} component of face image
$\mathbf{X}$	: matrix containing face images
$\mathbf{V}$	: eigenvectors of the matrix $\mathbf{X}*\mathbf{X}^T$
$\mathbf{U}$	: eigenvectors of the matrix $\mathbf{X}^T*\mathbf{X}$
Λ	: diagonal matrix containing the eigenvalues of matrix $\mathbf{X}*\mathbf{X}^T$
Δ	: matrix of singular values ($\Delta = \sqrt{\Lambda}$
I	: identity matrix
$\mathbf{C}$	: covariance matrix
ε	: error
$\mathbf{R}$	: radiance map
$\mathbf{p}, \mathbf{q}$	: gradients components of surface z
r	: radius
T	: number of faces in learning set
M	: number of selected eigenvectors
ψ	: arithmetic average of face images
φ	: caricature face
$\mathbf{A}$	: matrix of caricature faces
$\mathbf{u}$	: eigenfaces
$\in$	: projection error
$\mathbf{I}$	: video frame
$\mathbf{h}, \mathbf{g}$	: filter matrix
$\mathbf{f}$	: function
λ, τ	: hybrid model coefficients

ÖZYÜZLERLE YÜZ TANIMA

Özet

Temel parçacıklar analizine dayalı yüz tanıma algoritmalarının önemli sorunlarından biri aydınlanma değişimleridir. Bu çalışmada, aydınlanma değişimlerinden kaynaklanan sorunlara, gri seviyeli resimlerin kullanılması yerine ayrıt resimlerinin kullanılması ile yeni bir çözün sunulmuştur.

Aydınlanma kaynağının yerinde meydana gelen değişmelere, ayrıtların daha az duyarlı oldukları gösterilmiştir. Fakat ayrıt temelli yaklaşımın kendine özgü sorunları vardır. Poz değişimlerinden veya yüzdeki gölgelerin yerlerinin değişmesinden kaynaklanan farklılaşmalar, ayrıta dayalı yüz tanıma algoritmalarının performansını düşürür. Sunulan yöntemde, ayrıta ve dokuya dayalı yüz tanıma algoritmalarının sorunlarını ortadan kaldıran ve her iki yönteminde avantajlarını kullanan tepeler yaklaşımı geliştirilmiştir. Tepeler, ayrıtların zar ile kaplanması ile elde edilir. Her tepe görüntüsü daha sonra tepeler uzayında elde edilen öztepelerin doğrusal bileşimi olarak ifade edilir.

Öztepeler, özayrıtlar ve özyüzler ile yüz tanıma algoritmalarının aydınlanma farklılıklarına karşı performanslarının karşılaştırması Maya® yazılımı kullanılarak yapılan 14 taklit yüz ve Purdue A&R yüz veri tabanında yer alan 882 gerçek yüz görüntüsü kullanılarak gerçekleştirilmiştir. Deneysel olarak, sunulan yaklaşım olan öztepeler yönteminin diğer yöntemlerden daha iyi sonuç verdiği gösterilmiştir.

FACE RECOGNITION USING EIGENHILLS

Summary

Illumination changes are one of the basic problems of PCA-based face recognition algorithms. In this study, a new approach to overcome problems associated with illumination variation by utilizing the edge images rather than gray level texture images is presented.

It is shown that changes in edge images are limited as compared to changes in shading when illumination source is changed. However, using edges directly creates its own problems. A small displacement of edges due to pose change degrades the performance of the recognition algorithm drastically due to locality of edges. To combine the advantages of algorithms based on shading and edges while overcoming their drawbacks, "hills", which are obtained by covering edges with a membrane, are introduced. Each hill image is described as a combination of most descriptive eigenvectors, called "eigenhills", spanning hill space.

Recognition performances of eigenface, eigenedge and eigenhills methods are compared by considering illumination variation on 14 simulated faces designed using Maya® software and 882 real world faces of Purdue A&R face database. It is shown experimentally that proposed approach has the best recognition performance under these degradations as compared to other approaches.

1. INTRODUCTION

This introductory section deals with different applications (along with their constraints) that could benefit from the face recognition techniques. Next, former studies proposed by the researchers are presented. Finally, the stages, which are performed by any face recognition scheme, are demonstrated.

1.1 Applications and Constraints

The applications of Face Recognition Techniques can be broadly subdivided into commercial and law enforcement applications. A list detailing the applications along with their constraints is shown in table 1.1.

Table 1.1: Applications of face recognition, their advantages and disadvantages.

	Applications	Advantages	Disadvantages
1-a	Credit card, driver's license, passport, and personal identification	Controlled image Controlled segmentation Good quality images	No existing database Large potential database Rare search type
1-b	Mug shots matching	Mixed image quality More than one image available	No existing database Large potential database Rare search type
2	Bank / store security	High value Geographically localized search	Uncontrolled segmentation Low image quality
3	Crowd surveillance	High value Small file size Availability of video images	Uncontrolled segmentation Low image quality Real-time
4	Expert identification	High value Enhancement possible	Low image quality Legal certainty required
5	Witness face reconstruction	Witness search limits	Unknown similarity
6	Electronic mug shots book	Descriptor search limits	Viewer fatigue
7	Electronic line-up	Descriptor search limits	Viewer fatigue

These applications can be classified into two groups: Some applications' inputs are still images, while others are real time dynamic images. Even among these groups, significant differences exist, depending on the specific application. The differences are in terms of image quality, amount of background clutter, and nature, type and amount of input from a human (as in application 4 and 5).

Application 1a-b, 2, and 3 involve matching one face image to another face image. Applications 4-7 involve finding or creating a face image, which is similar to the human recollection of a face. Each of these applications imposes different

requirements on the recognition process. Matching requires that the candidate matching face image be in some set of face images selected by the system. Similarity detection requires, in addition to matching, that images of faces be found which are similar to a recalled face; this requires that the similarity measure used by the recognition system, closely matches the similarity measures used by humans.

This project deals with still images, which could come from a digital camera for security purposes (application 2).

The applications listed above are engineering problems. However, the engineers are not the only who investigate face recognition techniques. In fact, psychophysicists and neuroscientists also conduct research into this field. Their purpose is to understand the visual part of the human brain. One of their valuable techniques, Principal Components Analysis (PCA), also known as the Karhunen-Loeve transform (KLT), is used as the recognition algorithm in this project.

1.2 Background study on face recognition

Much of the work in computer recognition of faces is concentrated on detecting individual features of the faces such as eyes, nose, ears, lip, etc, and developing a relationship among them. However, research in human strategies of face recognition has shown that individual features and their immediate relationships are not sufficient for explaining human performance on handling different visual stimuli in face recognition [1].

By the year 1960's the researchers developed feature-based methods, focused on finding the independent local characteristics such as eye corners, lip corners, place of nose [2],[3]. However, this type of strategy is not only difficult to automate but also dependent on the facial expression changes.

Fischler and Elschlager [4] attempted to measure independent local characteristics of the face automatically. They used an algorithm that used local template matching and measure of features. This approach of template matching was improved by Yuille, Cohen, and Hallinan [5]. Their strategy was based on deformable templates, which were parameterized models of the face.

In the late 1980's connectionist approaches were developed to handle the recognition phenomena. Kohonen and Lahtio [6] described an associative network with a simple learning algorithm that can classify face images. Later, Flemming and Cottrell [7] extended the system by using non-linear units and training the system by

back-propagation. Stonham [8] built a general-purpose pattern recognition system, WISARD, based on neural networks, which had acceptable performance on binary face images.

Some other approaches automated face recognition by characterizing face by a set of geometric parameters [9],[12]. On the other hand, Kanade's face identification system [13] was the first, in which all steps of recognition process has been automated by using a top-down control strategy directed by a generic modal of expected feature characteristics.

After Sivorich and Kirby [14] announced their study on characterization of human faces using *Karhunen-Loeve Expansion* or *Principal Component Analysis*, many researchers attempted to use low dimensional principal component analysis approach for identification of faces. Turk and Pentland [15] developed a near real-time system, based on Sivorich and Kirby's proposal, that can detect and recognize faces using eigenspace decomposition, and named their approach as "eigenfaces".

Nayar, Murase and Nene [16] developed parametric appearance representation of the faces for recognition in a low dimensional sub-space called the eigenspace. Their approach was based on modeling the appearance, rather than shape.

Later, Moghaddam and Pentland [17] used probabilistic approach,

$$P(x \mid \Omega) = \frac{e^{-\frac{1}{2}d(x)}}{(2\pi)^{N/2}|\Sigma|^{1/2}}, \tag{1.1}$$

and Mahalonobis distance,

$$d(x) = (x-\bar{x})^T \Sigma^{-1}(x-\bar{x}), \tag{1.2}$$

(where x: face image, $\bar{x}$:arithmetic mean of learning set and Σ:covariance matrix), for classification of the faces.

However, afore-mentioned approaches on face recognition suffered from illumination changes. There are two approaches to overcome the problem related to illumination,:

1. Determining the illumination free components of a face image and using them directly or after a post processing for recognition purposes,

2. Modeling illumination effect on face structure, and modifying the shading of the face accordingly.

Belhumeur, Hepanha and Kriegman [18] developed an approach using Fischer space instead of eigenspace, they argued that Fischer space was insensitive to illumination. Their method was based on obtaining low dimensional representations of the faces using Fischer's Linear Discriminant, which produced separated classes in the low dimensional space.

Other studies to solve the illumination problem focused on finding illumination free components to identify a face. Pentland and Moghaddam [17] pointed out the advantages of an edge based approach. Recently, Tàkacs and Wechsler [19] proposed an edge based approach to recognize faces using Hausdorff distance,

$$H(A,B)=\max(h(A,B),h(B,A)), \tag{1.3}$$

where $h(A,B)=\max_{a\in A}\min_{b\in B}\|a-b\|$.

However, to overcome the noise in their representation of the edges they modified the *Hausdorff Distance* as follows,

$$h(A,B)=\frac{1}{N_a}\sum_{a\in A}\min_{b\in B}\|a-b\|. \tag{1.4}$$

The study that is proposed in this thesis is based on overcoming illumination by utilizing illumination free components of the face structure. The method uses Principal Component Analysis for face recognition purposes. The approach basically depends on the fact that edge maps do not drastically change for various illumination patterns. The locality problem of edges is solved by covering a membrane, described in regularization theory, over the edge map.

1.3 Three stages in face recognition

A general statement of the problem can be formulated as follows: given still or video images of a scene, identify one or more persons in the scene using a stored database of faces. The solution of the general problem is subdivided into three different stages:

1. *segmentation of scenes from cluttered scenes*,
2. *extraction of features from the face region*, and

3. *decision making.*

The following algorithm usually achieves the first stage, segmentation. An edge map is constructed, then the edges are connected using several heuristics, and the edges are matched into an elliptical shape using a Hough transform. It should be noted that if the input is composed of video images, the segmentation could be achieved using the motion as a cue.

The critical stage is the extraction of the features. There are essentially two types of features: holistic features (where each feature is a characteristic of the whole face) and partial features (hair, nose, mouth eyes, etc.). Partial features techniques make some measurements onto many crucial points of the face, whereas holistic feature technique deals always with the face as a whole. PCA is a holistic feature technique.

At the third stage a decision is taken from the data collected from the previous stage. Three type of decision can be achieved depending on the application:

1. *identification*, in which labels of individuals must be obtained,
2. *recognition* of a person, where it must be decided if the individual has already been seen, and
3. *categorization*, in which the face must be assigned to a certain category.

Since the topic of this report is to develop an efficient algorithm for based on PCA to overcome illumination problem in face recognition the first stage, face segmentation, will not be addressed.

The second section details the theory of PCA. Approach developed by Pentland and Moghaddam will follow the theory of PCA. The theoretical proof for using edges instead of texture information to overcome illumination problem of face recognition is outlined in fourth section. Pure edge based recognition using PCA will be outlined in the fifth section. Than the proposed approach, eigenhills is given in sixth section. Finally, experiments, which are made to test the performance of the proposed scheme, will be given.

2. FACE SPACE AND PRINCIPAL COMPONENTS

In this section, I will give information about the face space we are working in and a theoretical approach of *principal component analysis* to deal with the information for face classification purposes.

2.1 Face as a vector

A face, which is an image, can be viewed as a vector. If the image's width and height are ***w*** and ***h*** pixels respectively, the number of components of this vector will be ***wxh***. Each pixel is coded by one vector component. The construction of this vector from an image is performed by a simple concatenation - the rows of the image are placed after one another, as shown on the figure 2.1.

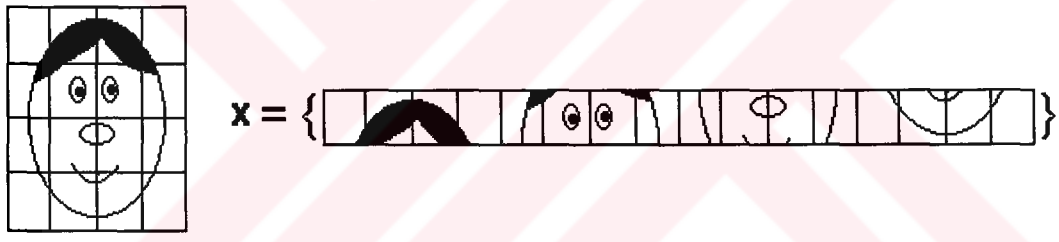


Figure 2.1: Formation of the *face vector* from *face image*.

2.2 Image space

The *face vector* described in the previous section belongs to a space, which is called *image space*, where all images' dimensions are ***w*** by ***h*** pixels. A sample for the image space is given in figure 2.2.

Figure 2.2: Sample image space, dimensions are 2x16.

As seen in figure 2.2 faces look like each other. They all have two eyes, a mouth, a nose, etc. placed at similar locations. Therefore, all the *face vectors* are placed in a very narrow cluster in the huge image space, as shown in the figure 2.3.

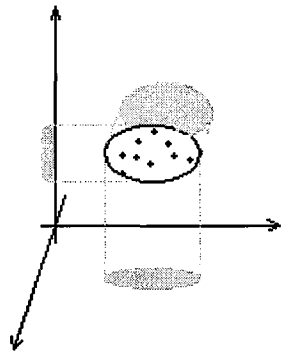


Figure 2.3: Faces lying together in the image space (face cluster).

The full *image space* is not an optimal space for face description, because the meaningful part is concentrated in a small fragment. Hence, the task is to build a *face space*, which better describes the faces. The basis vectors of this *face space* are called the *principal components*.

The dimension of the image space is ***wxh***. Of course, all the pixels of a face are not relevant, and each pixel depends on its neighbors. So, the dimension of the face space is less than the dimension of the image space. The dimension of the face space cannot be determined, but we are sure that it is to be far less than that of the image space.

The goal of the method, *Principal Components Analysis*, is to reduce the dimension of a set or space so that the new basis better describes the typical '*models*' of the set. In our case the '*models*' are a set of learning faces [22]. The new basis will be constructed by linear combination. Components in face space basis will be uncorrelated and will maximize the variance accounted for in the original variables. These properties are more easily understood when considering the example at the end of the section.

Principal Components Analysis aims to catch the total variation in the set of the learning faces, and to explain this variation by few variables. The fact that it reduces the dimension is important. In fact, an observation described by few variables is easier to handle and easier to understand than if it was defined by a huge amount of variables. And when many observations, or faces, have to be processed the dimensionality reduction is of first importance [22].

2.3 PCA theoretically

Principal Components Analysis was first developed by statisticians. Then, it has been reformulated in artificial neural network paradigm. So there are two ways to explain its principles. To gain an understanding of PCA, both these viewpoints

should be considered, for they are complementary. Next, PCA will be first explained from the statistical point of view which is mathematically rigorous, then it will be outlined in artificial neural network (ANN) sense.

2.3.1 PCA in statistical point of view

Though, statistical paradigm is not as intuitive as neural network paradigm, the reason I tell it prior is to give the chronological order. The study of PCA in statistical point of view allows obtaining further dimensionality reduction then the neural network approach, as it will be seen in the subsequent sections.

2.3.1.1 Transformation Matrix

The image space is highly redundant when used to describe faces. So, as has been previously demonstrated, a face space should be constructed. This redundancy is due to the fact that each pixel in a face is highly correlated to the other pixels. In fact, the covariance matrix, ***C***, for a set of faces is highly non-diagonal,

$$C_X = X * X^T = \begin{bmatrix} \sigma_{1,1}^X & \sigma_{1,2}^X & \cdots & \sigma_{1,N}^X \\ \sigma_{2,1}^X & \sigma_{2,2}^X & \cdots & \sigma_{2,2}^X \\ \cdots & \cdots & \cdots & \cdots \\ \sigma_{N,1}^X & \sigma_{N,2}^X & \cdots & \sigma_{N,N}^X \end{bmatrix}, \tag{2.1}$$

where, σ_{ij} represents the covariance between the pixel ***i*** and the pixel ***j***. There is a relation between the *covariance coefficient* and the *correlation coefficient*,

$$r_{i,j} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii} \cdot \sigma_{jj}}}, \tag{2.2}$$

the correlation coefficient is a *normalized covariance coefficient.*

The aim is to construct a *face space*, where each component is not correlated with any other components. This means that the covariance matrix of the new components should be diagonal,

$$C_Y = Y * Y^{T=} \begin{bmatrix} \sigma_{11}^Y & 0 & \cdots & 0 \\ 0 & \sigma_{22}^Y & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \sigma_{w*h,w*h}^Y \end{bmatrix}. \tag{2.3}$$

where

- y_i is the column vector describing the face x_i in the basis of the face space, the principal components,
- X is the matrix containing the faces, x_i (in the basis of the image space), and
- Y is the matrix containing the vectors, y_i.

This diagonal form of the covariance matrix implies that the variance of a variable with itself will be maximized whereas the covariance of a variable with any other variable will be nil. The variables will no longer be correlated any more. So, this construction finds those directions, which maximize the variance.

The principal components are calculated linearly. Let V be the transformation matrix, then $Y = V^T * X$ and $O = V * Y$. In fact $V = V^{-1}$, because V's columns are orthonormal one to each other, that is $V^T * V = I$.

Given the condition that C_Y is a diagonal matrix,

$$\begin{aligned} C_Y &= Y * Y^T = v^T * X * X^T * v \\ C_Y &= V^T * C_X * V \end{aligned} \tag{2.4}$$

So C_Y is the rotation of C_X by V.

If V is chosen as the eigenvectors of matrix C_X,

$$C_X * V = \Lambda * V, \tag{2.5}$$

where Λ is the diagonal matrix of eigenvalues of C_X. So,

$$C_Y = V^T * \Lambda * V = \Lambda * V^T * V = \Lambda, \tag{2.6}$$

and C_Y is the diagonal matrix containing the eigenvalues of C_X, since the diagonal elements of C_Y are the variance of the components of the learning faces in the face space, the eigenvalues of C_X are those variances. A firm grasp of the meaning of the elements of Λ is important in understanding the reduction of components that will be discussed subsequently.

2.3.1.2 Dimensionality Reduction

The aim of PCA is to reduce the dimension of the working space. The maximum number of principal components is the number of variable in the original space. However, in order to reduce the dimension, some principal components should be omitted.

Obviously, the dimensionality of the face space is less than the dimensionality of the image space,

$$\dim(Y) = \text{rank}\left(X * X^T\right) \text{x} T\,, \tag{2.7}$$

$$\dim(y_i) = \text{rank}\left(X * X^T\right) \text{x} 1\,. \tag{2.8}$$

$\text{rank}\left(X * X^T\right)$ is generally equal to ***T***, and ***T*** is the number of faces in the learning set. So a reduction of dimension has been made, and the information carried by $\boldsymbol{y}_i$ is absolutely the same as the information carried by the original face $\boldsymbol{x}_i$. A further reduction of dimension can be made.

The reconstruction of a face $\boldsymbol{x}_i$, by the components $\boldsymbol{y}_i$ is given in the following formula,

$$x_i = V * y_i\,. \tag{2.9}$$

If $\boldsymbol{v}_i$ are the column vectors of ***V***, i.e., the basis vectors, then, the reconstruction of a face, using only some of the components $\boldsymbol{y}_{ij}$ is,

$$\begin{aligned} \overline{x}_i &= \sum_{j=1}^{M} y_{ij} \cdot v_j + \sum_{j=M+1}^{T} y_{ij} \cdot v_j \\ \overline{x}_i &= \sum_{j=1}^{M} y_{ij} \cdot v_j + \sum_{j=M+1}^{T} a_j \cdot v_j \end{aligned} \tag{2.10}$$

where $\boldsymbol{a}_j$ is a constant whose value will be determined. This operation of selecting ***M*** introduces an error and the error vector is,

$$\varepsilon_i = x_i * \overline{x}_i = \sum_{j=M+1}^{T} \left(y_{i,j} - a_j\right) v_j\,. \tag{2.11}$$

The error vector is a random vector and its effectiveness is calculated by its mean square value,

$$\begin{aligned} \left\langle \varepsilon^2 \right\rangle &= \mu\left(\varepsilon^{\mathrm{T}} \varepsilon\right) \\ \left\langle \varepsilon^2 \right\rangle &= \mu\left(\sum_{j=M+1}^{T} \sum_{l=M+1}^{T} (y_{ij} - a_j)(y_{il} - a_l) \right) v_i^T v_l \\ \left\langle \varepsilon^2 \right\rangle &= \sum_{i=M+1}^{T} \mu\left[(y_i - a_i)^2\right] \end{aligned} \tag{2.12}$$

To minimize the error we can take its derivation with respect to $\boldsymbol{a}_i$,

$$\frac{\partial}{\partial a_i}\langle \varepsilon^2 \rangle = 0$$
$$-2 * \mu(y_i - a_i) = 0 \tag{2.13}$$

so,

$$a_i = \mu(y_i), \tag{2.14}$$

where $M < i < T$. This means that the omitted values of y_i have to be replaced by their mean value in order to minimize the reconstruction error. The error vector becomes,

$$\langle e^2 \rangle = \sum_{i=M+1}^{T} \mu[(y_i - \mu(y_i))^2], \tag{2.15}$$

which is in fact the sum of the variance of the omitted components [17]. From above equation, it can be seen that,

$$\langle e^2 \rangle = \sum_{i=M+1}^{T} \lambda_i . \tag{2.16}$$

Then, if the eigenvalues are classified in decreasing order, the last eigenvalues (and their eigenvectors) may be dropped. Which is the dimensionality reduction of PCA.

In effect, this means that some principal components can be dropped because they explain only a small amount of the data, whereas the largest amount of information is concentrated in the other principal components. The amount of information that the i^{th} principal component carries is given by its eigenvalue, λ_i. Usually the principal components (eigenvectors) are ordered in the decreasing order of their eigenvalues and the last ones are dropped. This fact is particularly important when augmented by the knowledge that for faces the decreasing of the eigenvalues is exponential as shown in figure 2.6.

2.3.1.3 Graphical Example

It is easier to understand PCA with a graphical example than with pure mathematical equations. Suppose, there is a random process that yields a two-dimensional result (x1, x2) and a large number of experiments of this process have been made. The results of these experiments are shown on the figure 2.4.

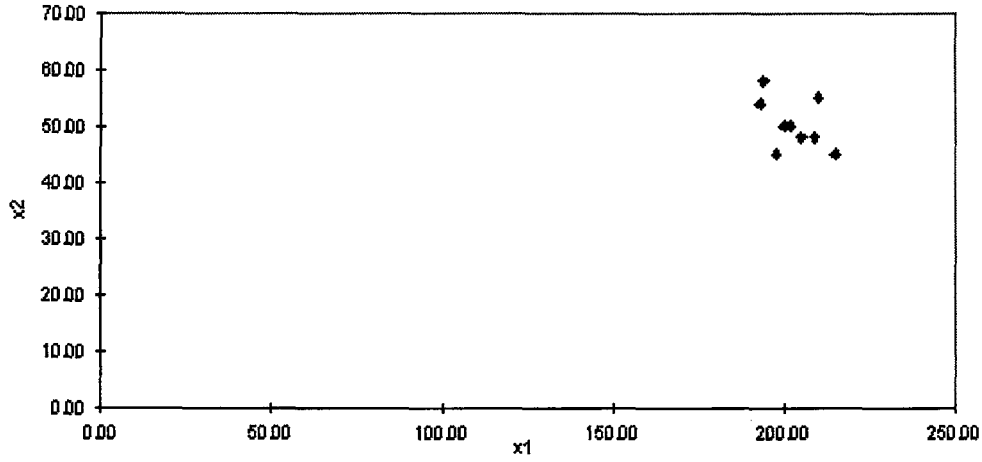


Figure 2.4: Random process in its original coordinates, x1 and x2.

It is clearly observed that, in this random process, x_1 is correlated to x_2. It seems that some axes other than x_1 and x_2, are more convenient to describe the process. The aim of PCA is to seek for the axes that maximize the variance of the data. Those axes are shown at the figure 2.5. They are a kind of feature of the process, and will, accordingly, be called *feature axis*.

It is obvious from the graph in figure 2.5, that the variance of the data is a maximum in the direction $\boldsymbol{v}_1$. Direction $\boldsymbol{v}_1$ maximizes the variation of the projection of the points. The direction that yields the largest variance of the data, provided that it is orthogonal to $\boldsymbol{v}_1$, is $\boldsymbol{v}_2$. This can be viewed, when looking at the weights of the data in each direction $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$: The data variation is the widest in the direction $\boldsymbol{v}_1$, and the next widest variation is in the direction $\boldsymbol{v}_2$.

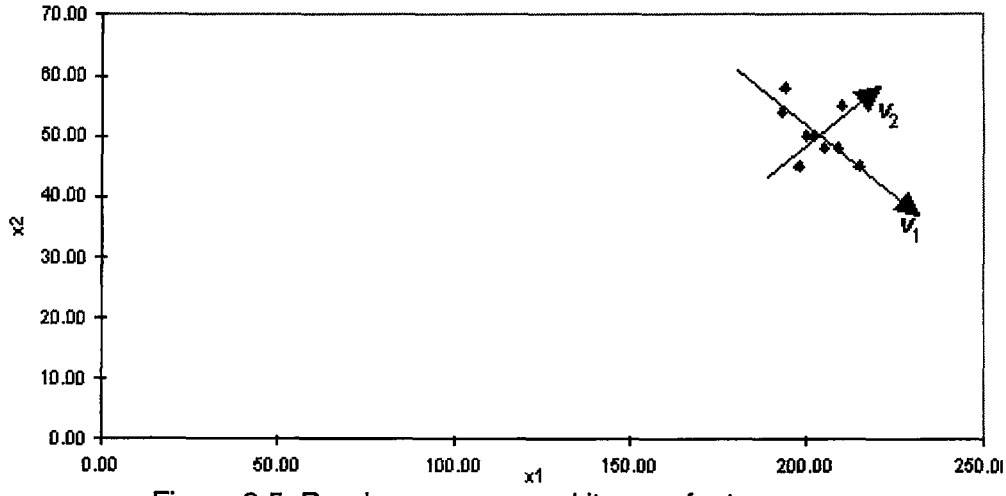


Figure 2.5: Random process and its own feature axes.

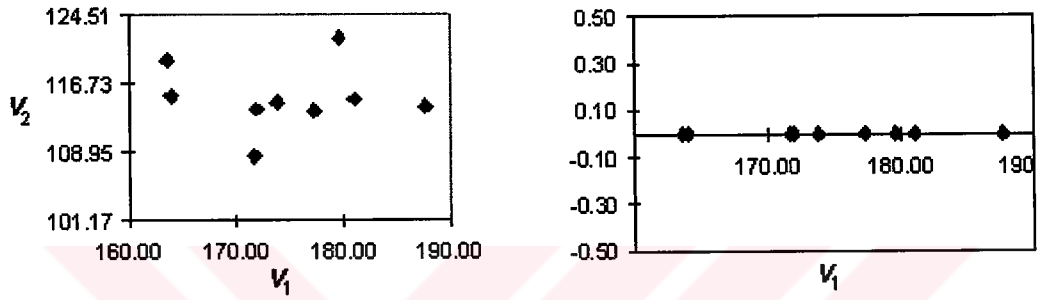
λ_1 the eigenvalue of the eigenvector $\boldsymbol{v}_1$ is 55, and λ_2, the one of $\boldsymbol{v}_2$ is 7. So, 88,7% of the data variance is explained by the first feature $\boldsymbol{v}_1$, and only 11,3% explained by the second feature $\boldsymbol{v}_2$. This means that $\boldsymbol{v}_1$ captures most of the variation (ie. the

distances between observations) in the original two-dimensional space. Hence, depending on the next stage of the process, the least important dimension, v_2, might be suppressed. Since this reduction of dimension suppresses information, it should only be made if the information is not relevant for the next stage of the process.

The principal components are linear combinations of the original variables,

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0.94 & 0.33 \\ -0.33 & 0.94 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \tag{2.17}$$

which is effectively rotating the original axes by 20°.



(a) (b)

Figure 2.6: (a) The random data in the feature space composed of the two features, (b) the data are represented in a space described only by the first feature.

Another property of the first principal component v_1 is that, it minimizes the sum of the squared distances of the observations from their perpendicular projection onto the largest principal component. For this reason, *the first principal component is sometimes referred to as the line of closest fit.* It should also be noted that, the variation of the points in the direction of the second principal component v_2 is smaller than, the variation of the points in the direction of the largest principal component, v_1.

2.3.2 Neural network paradigm

An auto-associative memory (AAM) is a kind of content addressable memory [21]. It aims to yield a response when the memorized key is similar to the input key. In our case, the term 'key' can be considered as being a face.

The linear auto-associative memory (LAAM) is formed by one neural network layer. Each neuron of this layer is associated with one component of the face vector [23]. Thus, the layer contains ***wxh*** neurons each. Each neuron is connected to all the neurons, as it is shown in the figure 2.7.

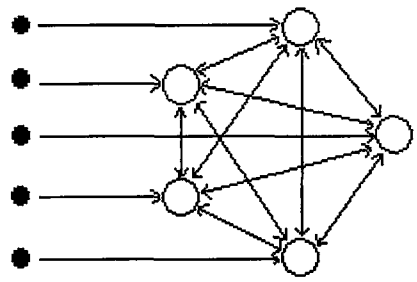


Figure 2.7: Architecture of the Linear Auto-Associative memory.

So the LAAM is constructed by computing the $(wxh)^2$ weights of the neural network. Those weights are calculated during the learning process. In this process, several learning faces are presented to the LAAM, which has to memorize them [22].

2.3.2.1 Widrow-Hoff Learning Rule

During the learning process the weights have to be changed in order to minimize the error. This can be done using several rules. One of them, which is used here, is the Widrow-Hoff learning rule [22].

Let:

- x_i be the i^{th} components of the input face,
- o_j be the output of the j^{th} neuron,
- w_{ij} be the intensity of the connection between the i^{th} and the j^{th} neurons,
- η be the learning constant and
- t be the iteration index.

In the case of an auto-associative memory, the Widrow-Hoff learning rule for a neuron can be formulated as follows,

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \eta * \left(x_j - o_j\right) * x_i . \tag{2.18}$$

For an AAM, the expected output of a neuron, o_j, is its input, x_j. This rule says that if the weight yields a result, which is different than expected, then it has to be modified. The amount of the change is proportional to the excitation of the neuron x_i and to learning constant η.

The learning rule is an iterative process, which has to be carried out for every face of the learning set several times. When this process is carried out on a single face, it is called an *iteration*, when it is achieved on the whole set of learning faces, it is

called an *epoch*. When several epochs are accomplished successively, it is obvious that the error, $\varepsilon_j = o_j - x_j$, is reduced, if η is well chosen.

This rule may be applied to the whole set of ***T*** learning faces using matrix notation.

Let:

- ***N*** = **w***x***h** be the number of pixels of the face or the number of components of the face vector,
- ***T*** be the number of faces in the learning set,
- $\boldsymbol{X} = [x_{nt}]$ be the matrix of dimension ***N***x***T*** defining the learning set. The t^{th} column vector of this matrix, x_t, corresponding to the t^{th} face of the learning set.
- $\boldsymbol{W} = [w_{ij}]$ be the weight matrix of dimension ***N***x***N***.

From now on, it is assumed that the vector x_t are normalized, which means $|x_t| = \sqrt{x_t^T \cdot x_t}$, where the superscript "T" indicates that the matrix is transposed.

So, the learning rule becomes,

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \eta * \left(X - W^{(t)} * X\right) * X . \qquad (2.19)$$

This is formula of the epoch, and here the index ***t*** is the index of the epoch.

Before proceeding, a digression concerning the matrices must be made. A real rectangular matrix, ***X***, can be decomposed into singular values:

Let,

- ***V*** be the matrix of the eigenvectors of the matrix $\boldsymbol{X}^*\boldsymbol{X}^T$,
- ***U*** be the matrix of the eigenvectors of the matrix $\boldsymbol{X}^{T*}\boldsymbol{X}$,
- Δ be the matrix of the singular values (i.e. $\Delta = \sqrt{\Lambda}$, where Λ is the diagonal matrix containing the eigenvalues of the matrix $\boldsymbol{X}^*\boldsymbol{X}^T$ which are the same of the eigenvalues of the matrix $\boldsymbol{X}^{T*}\boldsymbol{X}$).

The decomposition into singular values is,

$$X = V * \Delta * U . \qquad (2.20)$$

It should be noted from their definitions, that $V*V^t = I$ and $U*U^t = I$, where I is the identity matrix.

It is now possible to return to the learning rule, which was given as an iteration rule. The convergence of the iteration can be stated as follows,

$$\begin{aligned}
w^{(0)} &= 0 \\
w^{(1)} &= \eta*X*X^T = \eta*V*\Delta*U^T*U*\Delta*V^T = \eta*U*\Lambda*U^T \\
w^{(2)} &= \eta*V*\Lambda*V^T + \eta\left(V*\Delta*U^T - \eta*V*\Lambda*V^T*V*\Delta*U^T\right)*U*\Delta*V^T \\
w^{(2)} &= \eta*V*\Lambda*V^T + \eta*V*\Delta*U^T*U*\Delta*V^T - \eta^2*V*\Lambda*\Delta*U^T*U*\Delta*V^T \\
w^{(2)} &= \eta*V*\Lambda*V^T + \eta*V*\Lambda*V^T - \eta^2*V*\Lambda^2*V^T \\
w^{(2)} &= V*\left(\eta*\Lambda + \eta*\Lambda - \eta^2*\Lambda^2\right)*V^T \\
w^{(2)} &= V*\left(I - \left(I - \eta*\Lambda\right)^2\right)*V^T
\end{aligned} \tag{2.21}$$

So, epoch at ***t*** is,

$$w^{(t)} = V*\left(I - \left(I - \eta*\Lambda\right)^t\right)*V^T, \tag{2.22}$$

and the AAM converges if and only if,

$$\lim_{t\to\infty}(I - \eta\cdot\Lambda)^t = 0, \tag{2.23}$$

ie., if $0 < \eta < \frac{2}{\lambda_{max}}$, where λ_{max} is the largest eigenvalue of $X*X^T$. As a consequence, when the learning process has been done, the weight matrix has the following form,

$$\overline{W} = V*V^T. \tag{2.24}$$

As it has been noted that the aim of an AAM is to reconstruct the memorized faces. As the weight matrix has been calculated, the output of the AAM is,

$$\begin{aligned}
O &= \overline{W}*X \\
&= V*V^T*X \\
&= V*V^T*V*\Delta*U^T \\
&= V*\Delta*U^T \\
&= X
\end{aligned} \tag{2.25}$$

and the reconstruction is perfect.

2.3.2.2 Principal Components

As has been seen, a face, which has been memorized by the AAM, when presented as input to it and added with some noise, is perfectly reconstructed. This reconstruction process is achieved linearly by the matrix $\overline{W}$. It has also been noted that this matrix is formed by two parts, $\overline{W} = V * V^T$. Thus, the reconstruction process can be completed in two parts,

$$y = V^T * x, \tag{2.26}$$

$$o = V * y. \tag{2.27}$$

Here, ***x*** is a face (column vector), ***o*** is the reconstructed face and ***y*** is an intermediate form. The neural network architecture that brings this intermediate form to light is shown in figure 2.8.

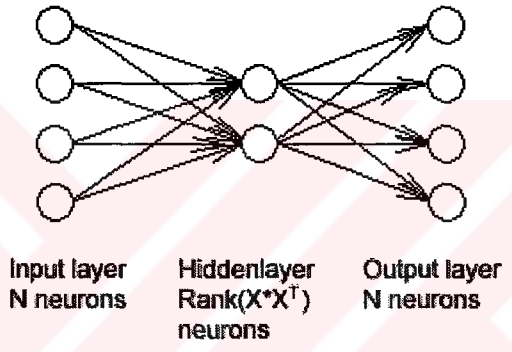


Figure 2.8: Neural net architecture presenting the intermediate form.

The matrix ***V*** transforms a face represented in the image space, ***x***, into a face described in the face space, ***y***. The column vectors of ***V*** are the basis of this face space, and are called the principal components. Due to the construction method of matrix ***V***, its vectors are *orthogonal*, and can be *normalized*. So the face space is described by an *orthonormal basis*. This basis casts a good set of *features* for the whole set of faces.

At this stage, some consideration should be given to the dimension of vectors and matrices involved in this process.

$$\dim(x) = \dim(o) = N\text{x}1, \tag{2.28}$$

$$\dim(V) = N\text{x}\,\text{rank}\left(X * X^T\right), \tag{2.29}$$

where $\text{rank}\left(X * X^T\right) = \min(N, T)$, and ***T*** is the number of faces in the learning set,

$$\dim(y) = \text{rank}\left(X * X^T\right)\text{x}1. \tag{2.30}$$

Often, as in the experiments here, ***N*** is larger than ***T***. Face database used in this work contains 126 faces in the learning set, whose dimensions are 121 pixels by 126 pixels. In this framework, a face coded by the principal components is coded by fewer components than its original pixel form. This is referred to as dimensionality reduction. The reduction factor here is large; in this example, the original face is composed by 15246 pixels and the intermediate form, is composed by 60 components. This dimensionality reduction could only be made because the original face has been projected into a new basis. This basis is given by the column vectors of the matrix ***V***. This basis is the best basis that can been found which describes the learnt faces. It forms a new space, called the *face space*.

The reduction can be done by choosing the eigenvectors that has maximum eigenvalues. In Sivorich and Kirby's study, they showed that, selecting ***M*** maximum eigenvalued eigenvectors results in an efficient classification of the faces. The selection of the ***M*** is important. As seen in figure 2.9, the eigenvalues fall exponentially, based on this result, approximately *one third* of the eigenvectors can be selected for classification.

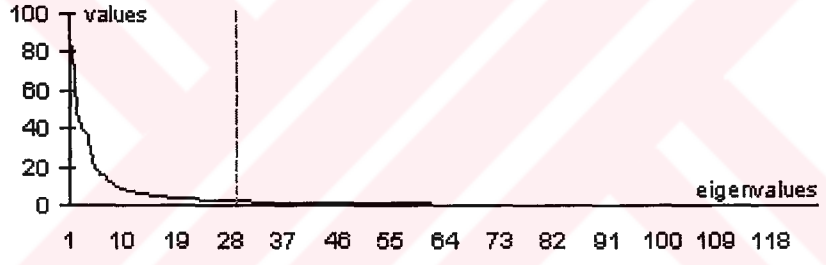


Figure 2.9: First 126 eigenvalues in descending order, for the learning set of the face database.

2.3.2.3 Conclusion

It can be found that the description of PCA in the neural network and linear auto-associative memory paradigm is interesting for several reasons: it gives an intuitive understanding for the principal components, and it shows clearly how PCA may be implemented.

3. PREVIOUS STUDY: EIGENFACE APPROACH

Using principal component analysis for face recognition and classification was first announced by Pentland and Turk [15], following the information theory approach proposed by Sivorich and Kirby [14] for coding of two dimensional face images. The basic theme of eigenface approach is to represent the face images in a relatively low dimensional space by finding the principal components of the huge image space as described in section 2.2.

3.1 PCA as a face recognition scheme

Suppose a face image is ***h*x*w***, so it can be represented in a (***h*x*w***) dimensional space as a point. This results in a learning set of face images that scatters a huge space. To find the best space that increases the variation among these face images, a low dimensional coordinate system must be found. To lead this goal, at first the arithmetic mean of a face is calculated by,

$$\Psi = \frac{1}{T}\sum_{i=1}^{T} x_i , \tag{3.1}$$

where ***T*** is the number of face images in the learning set. To gather zero-mean images, each face in learning set (*Appendix A*) is subtracted from mean face,

$$\varphi_i = x_i - \Psi , \tag{3.2}$$

where ***i***=1...***T***. The images obtained by subtraction are called caricatures. Mean face and the caricature faces are shown in figure 3.1a and b. To determine the image space in orthonormal basis, covariance matrix is to be constructed as follows,

$$C = \frac{1}{T}\sum_{i=1}^{T} \varphi_i \varphi_i^T \qquad C = A \cdot A^T \tag{3.3}$$

where, $A = [\varphi_1 \varphi_2 ... \varphi_T]$.

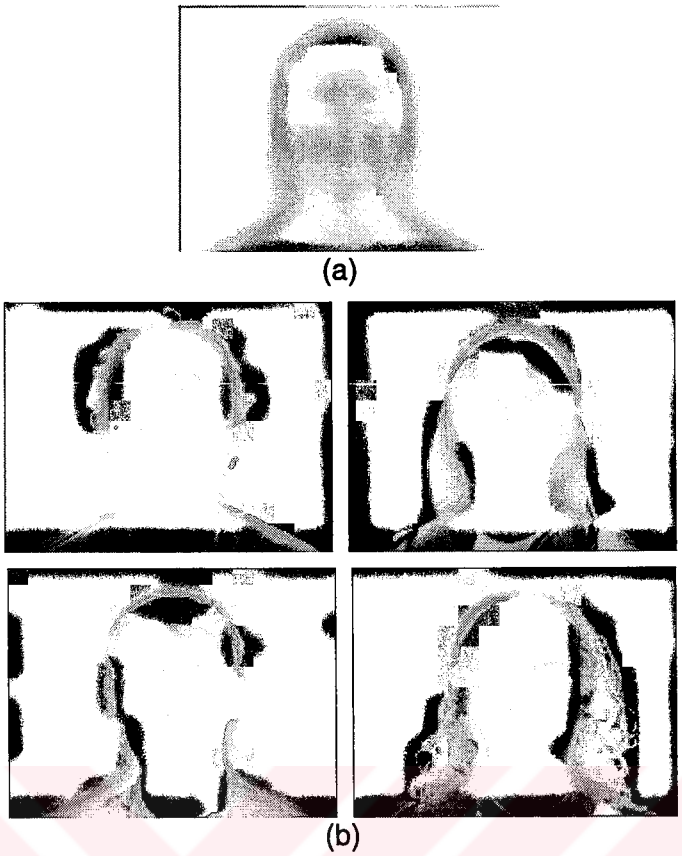


Figure 3.1: (a) Mean face for the learning set given in Appendix A, (b) selected caricature faces.

The dimensions of the covariance matrix are (***w***x***h***)x(***w***x***h***), which is for this case 19481^2 and is a very huge space to deal with. A reduction is to be made to make the calculations handier. To reduce dimension, before applying PCA, let's give a brief analysis of the image space. Since the variation is obtained among face images by finding $C = A \cdot A^T$, it is necessary to look, if $\hat{C} = A^T \cdot A$ holds adequate information for classification of faces, because the dimensions of $\hat{C}$ is only ***T***x***T*** which is generally smaller than (***w***x***h***)x(***w***x***h***).

In PCA paradigm, the eigenvectors of an orthonormal matrix, which is here *C* or $\hat{C}$, is to be calculated.

Let:

- ***u***, μ be the eigenvectors and eigenvalues of the covariance matrix *C*,
- ***v***, μ be the eigenvectors and eigenvalues of the covariance matrix $\hat{C}$.

The eigenvalues of C and $\hat{C}$ are the same (look section 2.3.2) so if a relation between ***u*** and ***v*** can be found then we can use $\hat{C}$, instead of *C* in our analysis.

$$\begin{aligned} \hat{C}v_i &= \mu_i v_i \\ (A^T A)v_i &= \mu_i v_i \\ A(A^T A)v_i &= \mu_i A v_i \\ (AA^T)Av_i &= \mu_i A v_i \\ Cu_i &= \mu_i u_i \end{aligned} \tag{3.4}$$

It is shown from the above analysis that,

$$\begin{aligned} u_i &= Av_i \\ u_i &= [\varphi_1 \varphi_2 \cdots \varphi_T]v_i \\ u_i &= \sum_{j=1}^{T} \varphi_{ij} v_{ij} \end{aligned} \tag{3.5}$$

Here u_i are called the eigenfaces, because of their ghostly shape. Eigenfaces calculated according to the above equation should be normalized. The normalization of the i^{th} eigenface can be calculated by,

$$u_l = \frac{u_l}{\sqrt{\sum_{i=1}^{wxh} u_{li}^2}}, \tag{3.6}$$

where $l = 1...T$. In figure 3.2, selected eigenfaces are given in descending order according to their eigenvalues.

3.1.1 Operations for classification

Since the principal components of the face space is calculated, all faces can be represented in eigenspace (principal component space) by the following equation,

$$w_i = u_i^T (x - \Psi), \tag{3.7}$$

where $i = 1...T$. In figure 3.3, more intuitive representation is given.

Figure 3.2: Eigenfaces in descending order according to their eigenvalues.

Figure 3.3: Sample reconstruction of a face using only four of the weights and the eigenfaces.

The classification of a face image is obtained by finding the distance of the input face to that of the learning set of faces. To accomplish classification an error criterion is defined by,

$$\in_k = \left\| \Omega - \Omega_k \right\|^2, \tag{3.8}$$

where, $\Omega^T = [w_1, w_2, ..., w_{T'}]$ and $k = 1, ..., T'$. This error is also used as recognition performance criterion in the following sections.

To obtain projection error,

$$\epsilon_k = \|\varphi - \varphi_k\|^2, \tag{3.9}$$

is to be found, where φ is the projected face and φ_k are the faces in the learning set.

3.2 Initialization and recognition steps

This approach to face recognition involves the following initialization steps:

1. Acquire an initial set of face images (learning set),
2. Calculate the eigenfaces from the learning set, keeping only ***M*** images that corresponds to the highest eigenvalues. These ***M*** images define the face space. As new faces are experienced, the eigenfaces can be updated or recalculated.
3. Calculate the corresponding distribution in ***M***-dimensional weight space for each known individual, by projecting the face images onto the face space.

Having initialized the system, the following steps are then used to recognize new face images:

1. Calculate a set of weights based on the input image and the ***M*** eigenfaces by projecting the input image onto each of the eigenfaces.
2. Determine if the image is a face at all by checking to see if the image is sufficiently close to face space.
3. If it is a face, classify the weight pattern as either a known person or as unknown.

3.3 PCA as face tracking scheme

Since the operations to project a face into face space is a near real-time process, PCA approach can be used as a face tracking scheme. The only drawback of the approach is to scan for a face when the first frame arrives, where each sub-image building the frame is projected to face space, until a face-like image is found. Though this is a time consuming process ($(H-h)\text{x}(W-w)$ projections, where H is the height and W is the width of the frame), once the face in first frame is found, it will be a simpler process for the succeeding frames, since in next frame the location of the face will be in neighborhood.

However, simple motion detection operation by spatio-temporal filtering (simply taking differences of successive frames) can also be applied to detect the face blob for a well-calibrated camera.

3.3.1 Face detection by projection

Application of $\epsilon_k = \|\varphi - \varphi_k\|^2$, projection error criterion, for making a decision if the image is a face or not, is a very expensive process. An analysis of the projection error yields less time consuming process as follows,

$$\begin{aligned} \epsilon &= \|\varphi - \hat{\varphi}\|^2 \\ \epsilon &= (\varphi - \hat{\varphi})^T (\varphi - \hat{\varphi}) \\ \epsilon &= \varphi^T \varphi - \varphi^T \hat{\varphi} - \breve{\varphi}^T (\varphi + \hat{\varphi}) \\ \epsilon &= \varphi^T \varphi - \hat{\varphi}^T \hat{\varphi} \end{aligned} \tag{3.10}$$

since $\hat{\varphi} \perp (\varphi - \hat{\varphi})$, where φ is caricature face from learning set and $\hat{\varphi}$ is the reconstructed caricature image by projection. Because $\hat{\varphi}$ is the linear combination of eigenfaces and eigenfaces are orthonormal vectors,

$$\begin{aligned} \hat{\varphi}^T \varphi &= \left(\sum_{i=1}^{M} w_i u_i \right)^T \left(\sum_{i=1}^{M} w_i u_i \right) \\ \hat{\varphi}^T \varphi &= \sum_{i=1}^{M} w_i^2 \left(u_i^T u_i \right) \\ \hat{\varphi}^T \varphi &= \sum_{i=1}^{M} w_i^2 \end{aligned} \tag{3.11}$$

and reconstruction error, which is also called the *saliency map*, for sub-image, Γ, at location ***(x,y)*** of frame I is,

$$\epsilon^2 (x, y) = \varphi^T (x, y)\, \varphi(x, y) - \sum_{i=1}^{M} w_i^2 (x, y) . \tag{3.12}$$

$\sum_{i=1}^{M} w_i^2 (x, y)$ can be calculated by a correlation with the ***M*** eigenfaces as follows,

$$\begin{aligned} \sum_{k=1}^{M} w_k^2 (i, j) &= \sum_{k=1}^{M} \varphi^T (i, j) u_i \\ \sum_{k=1}^{M} w_k^2 (i, j) &= \sum_{k=1}^{M} [\Gamma - \Psi]^T u_i \\ \sum_{k=1}^{M} w_k^2 (i, j) &= \sum_{k=1}^{M} \left[\Gamma^T u_i - \Psi^T u_i \right] \\ \sum_{k=1}^{M} w_k^2 (i, j) &= \sum_{k=1}^{M} \left[I(x, y) \otimes u_i - \Psi^T u_i \right] \end{aligned} \tag{3.13}$$

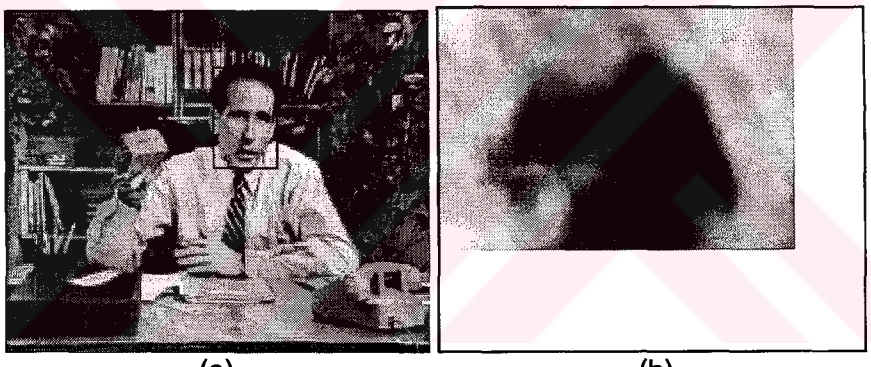
where $\otimes$ is the correlation operator. The first part of projection error $\varphi^T(x,y)\varphi(x,y)$ becomes,

$$\begin{aligned}\varphi^T(x,y)\varphi(x,y) &= [\Gamma-\Psi]^T[\Gamma-\Psi]\\ \varphi^T(x,y)\varphi(x,y) &= \Gamma^T\Gamma-2\Psi^T\Gamma+\Psi^T\Psi\\ \varphi^T(x,y)\varphi(x,y) &= \Gamma^T\Gamma-2\Psi\otimes\Gamma+\Psi^T\Psi\end{aligned} \tag{3.14}$$

So that the projection can be calculated in less time by,

$$\epsilon^2(x,y) = \Gamma^T\Gamma-2\Psi\otimes\Gamma+\Psi^T\Psi-\sum_{k=1}^{M}\left[I(x,y)\otimes u_i-\Psi^T u_i\right], \tag{3.15}$$

where, $\Psi^T\Psi$ and $\Psi^T u_i$ can be calculated ahead of time, and the computation to calculate the error only includes ***M***+1 correlations over the input sub-image $\Gamma = I(x,y)$. In figure 3.4, a frame form an advertisement video, and its corresponding *saliency map* is given.



(a) (b)

Figure 3.4: (a) A frame from an advertisement video and the location of head, (b) corresponding saliency map.

3.4 Problems of face detection using *Eigenfaces*

There are two basic drawbacks of this approach. One of them is that a variation in illumination, which results in a variation in brightness, will degrade the recognition performance. The other one is the presence noise in subject image, which can be a result of an out of focus camera, foggy air, etc.

The performance degradations are due to the fact that eigenface method is highly correlated to local texture information within the face. In figure 3.5, presence of noise and change in illumination in shown.

For achieving high recognition rates, more robust presentations of images, for the conditions given in figure 3.5, must be used. In the next section, edge based representations of images are analyzed for robustness to noise and illumination change.

Figure 3.5: (a) Original image, (b) changed illumination pattern, (c) Blurred image, (d) noisy image.

4. THEORETICAL ANALYSIS OF ILLUMINATION EFFECTS ON EDGES

In this section, an intuitive analysis of effects of illumination variation on the edge maps will be given.

4.1 Introduction

For two point light sources, one turned on at an instant of time, illumination change can drastically alter the appearances of objects. To describe the differences between the object illuminated with the first light source to the object illuminated with the second light source, we can look at the distance from one to the other. In case of a lambertian surface, which scatters light equally in all directions, the radiance map is,

$$R(p,q;p_s,q_s)=\frac{1+pp_s+qq_s}{\sqrt{1+p_s^2+q_s^2}\sqrt{1+p^2+q^2}}, \tag{4.1}$$

where

$$p=\frac{\partial z}{\partial x},\ q=\frac{\partial z}{\partial y}, \tag{4.2}$$

and ***z*** is the depth map of the object.

For input image $I(x,y)$, the radiance map, $R(p(x,y),q(x,y);p_s,q_s)$, gives the gray level intensity at point (x,y). That is,

$$R(p(x,y),q(x,y);p_s,q_s)=I(x,y). \tag{4.3}$$

For two different light sources (p_{s_1},q_{s_1}) and (p_{s_2},q_{s_2}), where $p_{s_2}=p_{s_1}+\delta_{ps}$ and $q_{s_2}=q_{s_1}+\delta_{qs}$, corresponding radiance maps will be,

$$R_1(p,q,p_{s_1},q_{s_1})=\frac{1+pp_{s_1}+qq_{s_1}}{\sqrt{A}\sqrt{1+p^2+q^2}}, \tag{4.4}$$

$$R_2(p,q,p_{s_2},q_{s_2})=\frac{1+pp_{s_1}+qq_{s_1}+p\delta_{ps}+q\delta_{qs}}{\sqrt{B}\sqrt{1+p^2+q^2}}, \tag{4.5}$$

where,

$$A = 1 + p_{s_1}^2 + q_{s_1}^2, \tag{4.6}$$

$$B = 1 + p_{s_1}^2 + q_{s_1}^2 + 2p_{s_1}\delta_{ps} + 2q_{s_1}\delta_{qs} + \delta_{ps}^2 + \delta_{qs}^2, \tag{4.7}$$

and $\delta_{ps} \neq 0$, $\delta_{qs} \neq 0$.

The edge maps of a surface illuminated with a light source can be obtained by finding the zero crossings of the laplacian operator [20]. The laplacian operator is,

$$\nabla^2 R = R_{xx} + R_{yy}, \tag{4.8}$$

where sub-scripts ***xx*** and ***yy*** represent the *second derivatives* with respect to ***x*** and ***y***. In figure 4.1a,b and c, step edge, its first and second derivatives are shown respectively.

To measure the sensitivity of texture and edge maps we can look at the changes between the radiance maps, $R_1(p,q;p_{s_1},q_{s_1})$, $R_2(p,q;p_{s_2},q_{s_2})$ and edge maps, $\nabla^2 R_1$, $\nabla^2 R_2$. So far, the theorem that is to be proofed can be stated as follows,

$$\left|R_1(p,q;p_{s_1},q_{s_1}) - R_2(p,q;p_{s_2},q_{s_2})\right| \overset{?}{>} \left|\nabla^2 R_1 - \nabla^2 R_2\right|, \tag{4.9}$$

where $|f|$ is the absolute value of the function f.

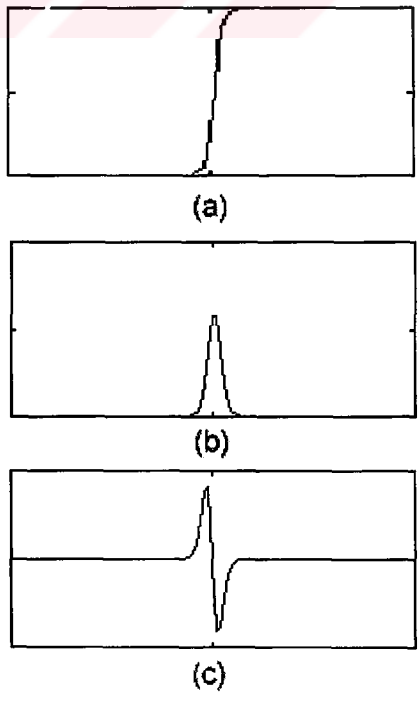


Figure 4.1: (a) Step edge, (b) its first derivative, and (c) second derivative.

In the following sub-sections analysis of changes between edge maps and textures for two distinct point light sources of four different objects: the planar object, spherical object, simulated face and real human face will be given. For the planar, spherical and simulated objects, the surfaces are considered to be lambertian, whereas the real human face has both lambertian and non-lambertian regions.

4.2 Planar objects

In planar lambertian surfaces, the depth information, ***z***, does not change. So,

$$p = \frac{\partial z}{\partial x} = z_x = 0,\ q = \frac{\partial z}{\partial y} = z_y = 0, \tag{4.10}$$

and the image is seen equally bright in all viewing directions. The radiance function corresponding to planar case is,

$$R(p_s, q_s) = \frac{1}{\sqrt{1 + p_s^2 + q_s^2}}, \tag{4.11}$$

since $p = q = 0$. The first and second derivatives of the above function with respect to ***x*** and ***y*** will be,

$$\begin{aligned} R_x = 0 &\rightarrow R_{xx} = 0 \\ R_y = 0 &\rightarrow R_{yy} = 0 \end{aligned} \tag{4.12}$$

So the laplacian of the planar surface is,

$$\nabla^2 R = R_{xx} + R_{yy} = 0 + 0 = 0. \tag{4.13}$$

which means, the planar case does not hold any edges except on the boundaries, because there is no zero crossings within the object. However on the boundary locations there is a texture change that gives an edge.

Two edge maps obtained for two different illuminations will be the same,

$$\left| \nabla^2 R_1 - \nabla^2 R_2 \right| = 0. \tag{4.14}$$

The difference between these two images of the same planar surface obtained from the reflectance of the surface with two distinct light sources is,

$$|R_1 - R_2| = \left|\frac{1}{\sqrt{A}} - \frac{1}{\sqrt{B}}\right|$$
$$|R_1 - R_2| = \left|\frac{\sqrt{B} - \sqrt{A}}{\sqrt{A}\sqrt{B}}\right| \tag{4.15}$$

since $\delta_{ps} \neq 0$ and $\delta_{qs} \neq 0$, the difference, $|R_1 - R_2|$, is always a positive value, greater than zero. It can be concluded that,

$$\left|\nabla^2 R_1 - \nabla^2 R_1\right| < |R_1 - R_2|, \tag{4.16}$$

and the edge-based representation is more robust to illumination changes than the representation based on gray level intensities for the planar case.

4.3 Spherical objects

Now let's analyze if the above proof holds for the spherical lambertian objects. For spherical lambertian objects, the depth information is $z = \sqrt{C}$, where

$$C = r^2 - x^2 - y^2. \tag{4.17}$$

For sphere *p*, *q* will respectively be,

$$p = \frac{\partial z}{\partial x} = \frac{\partial}{\partial x}\left(\sqrt{r^2 - x^2 - y^2}\right) = \frac{-x}{\sqrt{C}}, \tag{4.18}$$

$$q = \frac{\partial z}{\partial y} = \frac{\partial}{\partial y}\left(\sqrt{r^2 - x^2 - y^2}\right) = \frac{-y}{\sqrt{C}}, \tag{4.19}$$

and radiance map, *R*, is,

$$R(p,q;p_s,q_s) = \frac{1 + pp_s + qq_s}{\sqrt{1 + p_s^2 + q_s^2}\sqrt{1 + p^2 + q^2}}$$
$$R(p,q;p_s,q_s) = \frac{1 - \frac{x}{\sqrt{C}}p_s - \frac{y}{\sqrt{C}}q_s}{\sqrt{1 + p_s^2 + q_s^2}\sqrt{1 + \frac{x^2}{C} + \frac{y^2}{C}}}$$
$$R(p,q;p_s,q_s) = \frac{\frac{\sqrt{C} - xp_s - yq_s}{\sqrt{C}}}{\sqrt{1 + p_s^2 + q_s^2}\frac{\sqrt{(r^2 - x^2 - y^2) + x^2 + y^2}}{\sqrt{C}}} \tag{4.20}$$
$$R(p,q;p_s,q_s) = \frac{\sqrt{C} - xp_s - yq_s}{r\sqrt{1 + p_s^2 + q_s^2}}$$

To calculate laplacian, first and second derivatives must be calculated. The first derivative with respect to ***x*** is,

$$\begin{aligned}
\frac{\partial R}{\partial x} &= \frac{\partial}{\partial x}\left(\frac{\sqrt{C}-xp_s-yq_s}{r\sqrt{1+p_s^2+q_s^2}}\right)\\
\frac{\partial R}{\partial x} &= \frac{1}{r\sqrt{1+p_s^2+q_s^2}}\cdot\frac{\partial}{\partial x}\left(\sqrt{r^2-x^2-y^2}-xp_s-yq_s\right)\\
\frac{\partial R}{\partial x} &= \frac{-1}{r\sqrt{1+p_s^2+q_s^2}}\cdot\left(p_s+\frac{x}{\sqrt{C}}\right)\\
\frac{\partial R}{\partial x} &= \frac{-\sqrt{C}p_s-x}{r\sqrt{1+p_s^2+q_s^2}\sqrt{C}}
\end{aligned} \tag{4.21}$$

Similarly derivative with respect to ***y*** is,

$$\begin{aligned}
\frac{\partial R}{\partial y} &= \frac{\partial}{\partial y}\left(\frac{\sqrt{C}-xp_s-yq_s}{r\sqrt{1+p_s^2+q_s^2}}\right)\\
\frac{\partial R}{\partial y} &= \frac{1}{r\sqrt{1+p_s^2+q_s^2}}\cdot\frac{\partial}{\partial y}\left(\sqrt{r^2-x^2-y^2}-xp_s-yq_s\right)\\
\frac{\partial R}{\partial y} &= \frac{-1}{r\sqrt{1+p_s^2+q_s^2}}\cdot\left(q_s+\frac{y}{\sqrt{C}}\right)\\
\frac{\partial R}{\partial y} &= \frac{-\sqrt{C}q_s-y}{r\sqrt{1+p_s^2+q_s^2}\sqrt{C}}
\end{aligned} \tag{4.22}$$

Second derivative with respect to ***x*** is,

$$\begin{aligned}
\frac{\partial^2 R}{\partial x^2} &= \frac{\partial R_x}{\partial x} = \frac{\partial}{\partial x}\left(\frac{-1}{r\sqrt{1+p_s^2+q_s^2}}\cdot\left(p_s+\frac{x}{\sqrt{C}}\right)\right)\\
\frac{\partial^2 R}{\partial x^2} &= \frac{-1}{r\sqrt{1+p_s^2+q_s^2}}\cdot\frac{\partial}{\partial x}\left(p_s+\frac{x}{\sqrt{r^2-x^2-y^2}}\right)\\
\frac{\partial^2 R}{\partial x^2} &= \frac{-1}{r\sqrt{1+p_s^2+q_s^2}}\cdot\frac{r^2-y^2}{\sqrt{C}^3}\\
R_{xx} &= \frac{y^2-r^2}{r\sqrt{1+p_s^2+q_s^2}\sqrt{C}^3}
\end{aligned} \tag{4.23}$$

Second derivative with respect to y is,

$$\begin{aligned}
\frac{\partial^2 R}{\partial y^2} &= \frac{\partial R_y}{\partial y} = \frac{\partial}{\partial y}\left(\frac{-1}{r\sqrt{1+p_s^2+q_s^2}} \cdot \left(q_s + \frac{y}{\sqrt{C}} \right) \right) \\
\frac{\partial^2 R}{\partial y^2} &= \frac{-1}{r\sqrt{1+p_s^2+q_s^2}} \cdot \frac{\partial}{\partial y}\left(q_s + \frac{y}{\sqrt{r^2-x^2-y^2}} \right) \\
\frac{\partial^2 R}{\partial y^2} &= \frac{-1}{r\sqrt{1+p_s^2+q_s^2}} \cdot \frac{r^2-x^2}{\sqrt{C}^3} \\
R_{yy} &= \frac{x^2-r^2}{r\sqrt{1+p_s^2+q_s^2}\sqrt{C}^3}
\end{aligned} \tag{4.24}$$

Following these derivations, laplacian is,

$$\nabla^2 R = R_{xx} + R_{yy} = \frac{x^2+y^2-2r^2}{r\sqrt{A}\sqrt{C}^3}. \tag{4.25}$$

$\nabla^2 R$ is not a function of δ_{ps} or δ_{qs}, and r is constant, zero crossings of $\nabla^2 R$ will occur only when

$$\begin{aligned}
x^2+y^2-2r^2 &= 0 \\
x^2+y^2 &= 2r^2
\end{aligned} \tag{4.26}$$

which are the boundary locations of the sphere. Hence, the difference between edge maps obtained with various lighting conditions will be,

$$\left| \nabla^2 R_1 - \nabla^2 R_2 \right| = 0. \tag{4.27}$$

The difference of intensity values between two sphere images obtained for two different illumination conditions outlined in introduction of the section is,

$$\begin{aligned}
\left| R_1 - R_2 \right| &= \left| \frac{\sqrt{C} - xp_{s_1} - yq_{s_1}}{r\sqrt{A}} - \frac{\sqrt{C} - xp_{s_2} - yq_{s_2}}{r\sqrt{B}} \right| \\
\left| R_1 - R_2 \right| &= \left| \frac{\sqrt{C} - xp_{s_1} - yq_{s_1}}{r\sqrt{A}} - \frac{\sqrt{C} - xp_{s_1} - yq_{s_1}}{r\sqrt{B}} + \frac{x\delta_{ps} + y\delta_{qs}}{r\sqrt{B}} \right| \\
\left| R_1 - R_2 \right| &= \left| \frac{\sqrt{C} - xp_{s_1} - yq_{s_1}}{r\sqrt{A}\sqrt{B}} \left(\sqrt{B} - \sqrt{A} \right) + \frac{x\delta_{ps} + y\delta_{qs}}{r\sqrt{B}} \right|
\end{aligned} \tag{4.28}$$

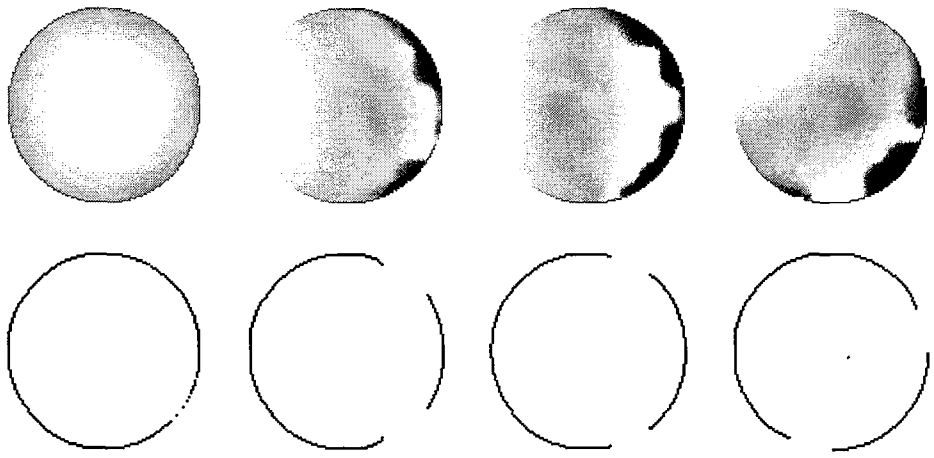
For the above equation, the parameters, which control the distinction between two images, are δ_{ps} and δ_{qs}, so the differences for the texture can be analyzed as follows,

$$\lim_{\delta_{ps},\delta_{qs}\to\infty}\left|R_1 - R_2\right| = \lim_{\delta_{ps},\delta_{qs}\to\infty}\left|\frac{\left(\sqrt{C} - xp_{s_1} - yq_{s_1}\right)}{r\sqrt{A}\sqrt{B}}\left(\sqrt{B} - \sqrt{A}\right) + \frac{x\delta_{ps} + y\delta_{qs}}{r\sqrt{B}}\right|$$

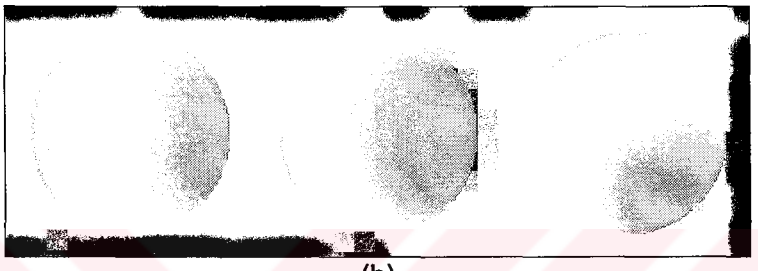
$$\lim_{\delta_{ps},\delta_{qs}\to\infty}\left|R_1 - R_2\right| = \lim_{\delta_{ps},\delta_{qs}\to\infty}\left|\frac{x\delta_{ps} + y\delta_{qs}}{r\sqrt{1 + p_{s_1}^2 + q_{s_1}^2 + 2p_{s_1}\delta_{ps} + 2q_{s_1}\delta_{qs} + \delta_{ps}^2 + \delta_{qs}^2}}\right| \quad (4.29)$$

$$\lim_{\delta_{ps},\delta_{qs}\to\infty}\left|R_1 - R_2\right| = \infty$$

The observation, $0 << \lim_{\delta_{ps},\delta_{qs}\to\infty}\left|R_1 - R_2\right|$, shows that, for two distinct light sources, change in the edge map of the images is smaller, compared to the texture change. This observation is given in the figures 3.2a, 3.2b and 3.3. The first row of figure 4.2a demonstrates the images of a sphere taken for various illumination conditions, while second row of figure 4.2a shows the edge maps corresponding to the spheres. In figure 4.2b $\left|R_1 - R_2\right|$ differences, where R_1 is the left most sphere on the first row of figure 4.2a and R_2 is the rest of the spheres of the same row, are given for intuitive reasoning. As seen, $\left|R_1 - R_2\right| > \left|\nabla^2 R_1 - \nabla^2 R_2\right|$. More descriptive result is given in figure 4.3, where $\left|R_1 - R_2\right|$ difference for spheres are obtained under different illumination conditions. In figure 4.3, R_1 corresponds to illuminated image for p_s=0, q_s=0 and R_2 corresponds to p_s=δ, q_s=0, where δ takes values δ_1=0.5, δ_2=2, δ_3=7 and δ_4=1000, respectively.



(a)



(b)

Figure 4.2: (a) First row is the collection of sphere images for varying illumination, second row is corresponding edge maps, (b) |R1-R2|, where R1 is the left-most sphere in first row and R2 corresponds to the rest of the images in first row in (a)

Figure 4.3: Normalized |R1-R2| distance, where R1 is the image for ps=0, qs=0 and R2 is the varyingly illuminated image for ps=δ, qs=0 (δ1=0.5, δ2=2, δ3=7 and δ4=1000).

To quantitatively evaluate the results for the edges given in figure 4.2a, the conditional probability of a detected edge pixel given an ideal edge pixel, Pr(*DE*|*IE*), the conditional probability of an ideal edge pixel given a detected edge pixel, Pr(*IE*|*DE*), and the mean square distance (MSD) between ideal and detected edge pixels are calculated. The edge map corresponding to the left most image in the first row in figure 4.2a is selected as ideal edge map. The results are displayed in figure 4.4.

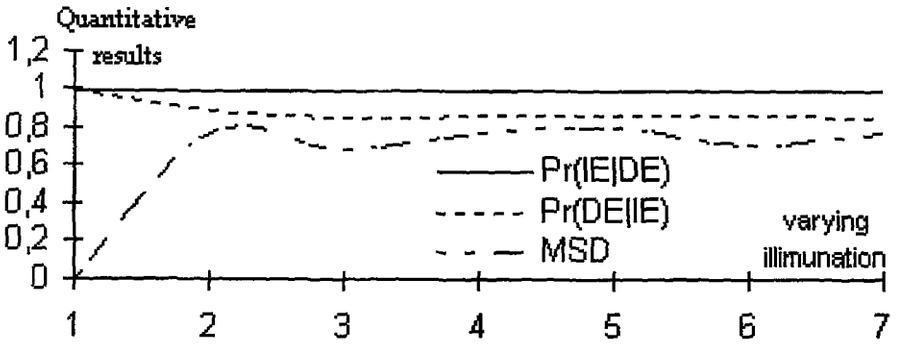
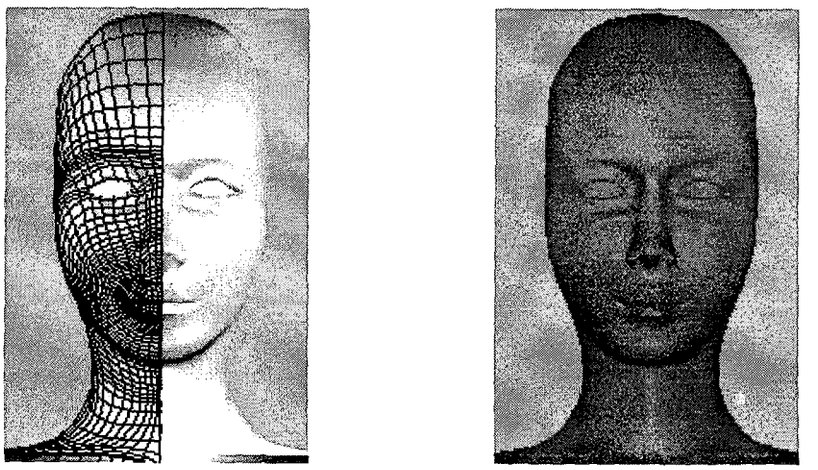


Figure 4.4: Conditional probabilities Pr(IE|DE) and Pr(DE|IE), and mean square distance (MSD) between ideal and detected edges.

From figure 4.4, it can be observed that most of the edges coincide and hold adequate information for recognition purposes even though the light source changes drastically. In the case of Pr(*IE*|*DE*), we conclude that all detected edges are ideal edges, and for Pr(*DE*|*IE*), approximately 90% of the idea edges are detected.

4.4 Simulated faces

So far, the planar and spherical lambertian objects are analyzed and it is shown that edges are more robust to variation in illumination. Human faces, however, have a highly non-convex shape creating cast shadows, even one can find planar and spherical regions on it. In order to observe the importance of this non-convex structure on edge maps, a simulation experiment on a wire-frame face image designed on Maya® software running on Silicon Graphics workstation is constructed. Wire-frame structure and experiment conditions are shown in figure 4.5a and 4.6 respectively.



(a) (b)

Figure 4.5: (a) Wire-frame structure of the simulated face, designed by Maya® software under Silicon Graphics Workstation, (b) face image rendered with only illuminated by an ambient light source.

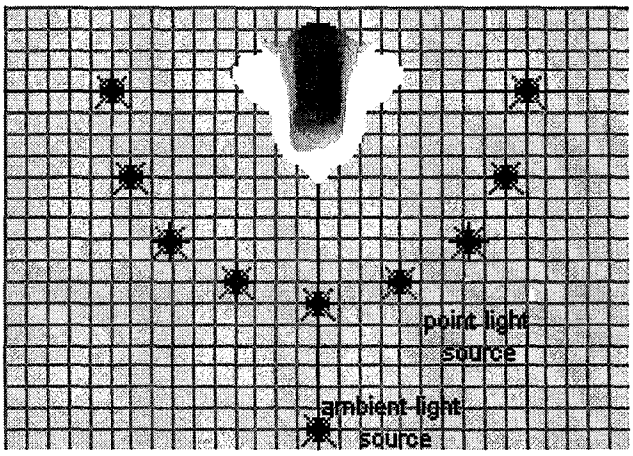


Figure 4.6: Experiment conditions designed in Maya® software for analyzing the response of edges under varying illumination.

To utilize effects of changing illumination conditions, virtual light sources are arranged in a circular region, with 22.5° between each. An ambient light source is placed to simulate natural lighting conditions for gathering the ideal face image, which is shown in figure 4.5b. Rendered face images are captured from wire-frame structure by turning on one light source along with the ambient light. Face images obtained by rendering are shown in figure 4.7a and corresponding edge maps are shown in figure 4.7b.

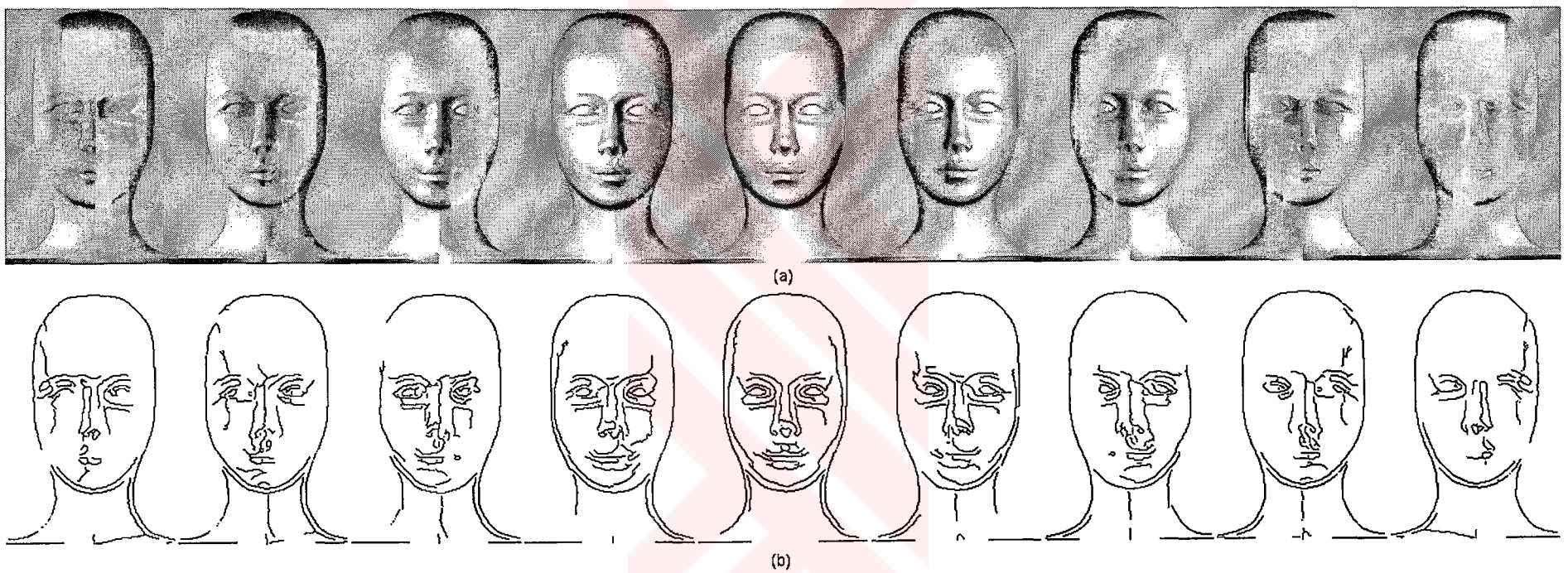


Figure 4.7: (a) Face images rendered for lighting conditions shown in figure 4.6, (b) corresponding edge images for the faces in (a).

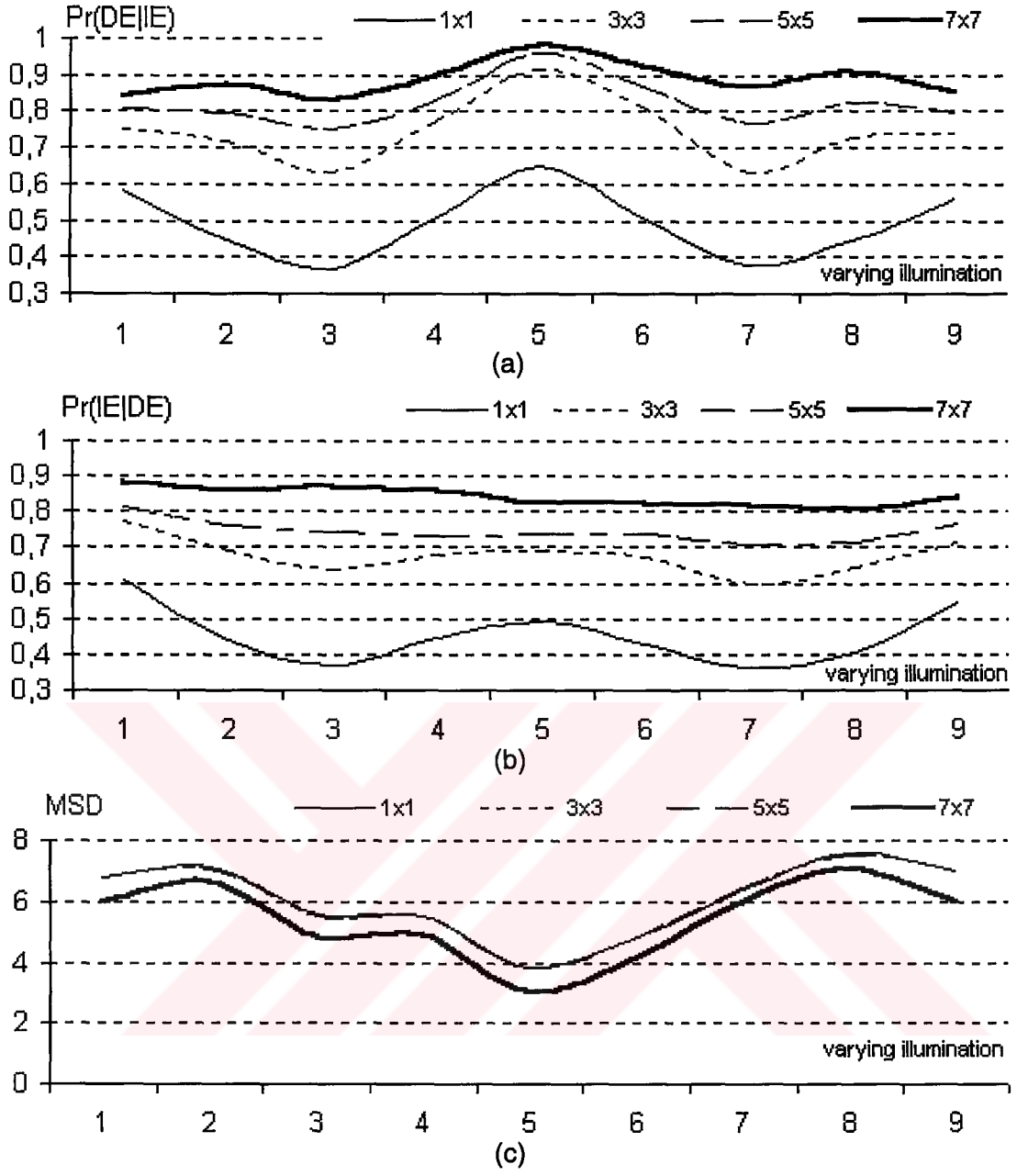


Figure 4.8: (a) Pr(DE|IE) probabilities, (b) Pr(IE|DE) probabilities, (c) mean square distance measure for varyingly illuminated faces.

To quantitatively evaluate the effect of variation in illumination for rendered simulated face images in figure 4.7a, the probabilistic measures Pr(IE|DE), Pr(DE|IE) and MSD which were used for spherical objects are used. The probabilities are calculated for 1x1, 3x3, 5x5 and 7x7 neighborhood masks are plotted in figure 4.8a, b and c respectively. The most accurate result, that is closer to 1, is obtained for 7x7-neighborhood mask. This result is due to compensating the movements of narrow-structured edges, resulting from changing lighting conditions, by *widening a dot (edge) to a 7x7 square.*

From this outcome, it can be argued that edge maps can't be directly applied to robust face recognition algorithms and a post-processing operation should modify their narrow structures. However, this modification should not disturb the localization of the edge.

4.5 Real Faces

Following the simulation experiments for analyzing the effects of illumination variation on edge maps of rendered face images, the case for real world faces of Purdue A&R face database is scrutinized. Natural lighting conditions of 113 subject faces are used as the ideal case and three groups of 339 images are used as the test case. The first of the three test groups is with a light source placed on left side of face and is turned on, second group right side light turned on and last group is both sidelights are turned on. The ideal face, three groups and accompanying edges are shown in figure 4.9a,b and c respectively.

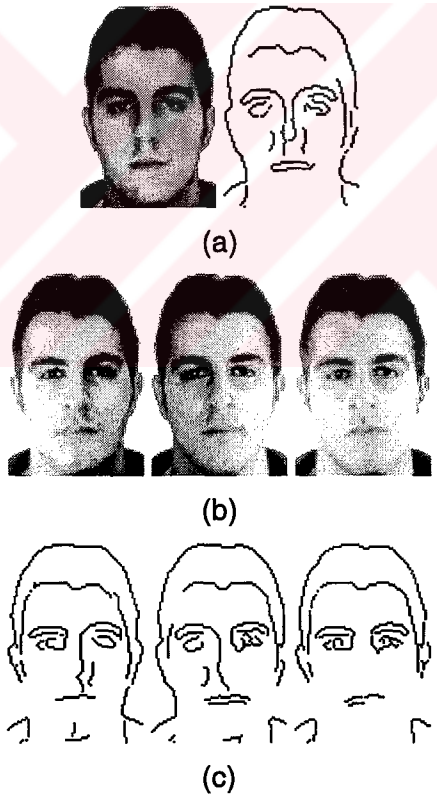


Figure 4.9: Face images from Purdue A&R face database (a) Naturals lighting condition, (b) three groups of illumination variation, (c) corresponding edges for the three groups.

The measures for quantitative evaluations are afore-mentioned probabilistic calculations. The diagrams plotted in figure 4.10 are for 1x1 and 7x7 neighborhood

mask. The use of 7x7 mask is due to the fact of non-ideal alignment of camera in addition to the reason concluded in the case for simulated faces.

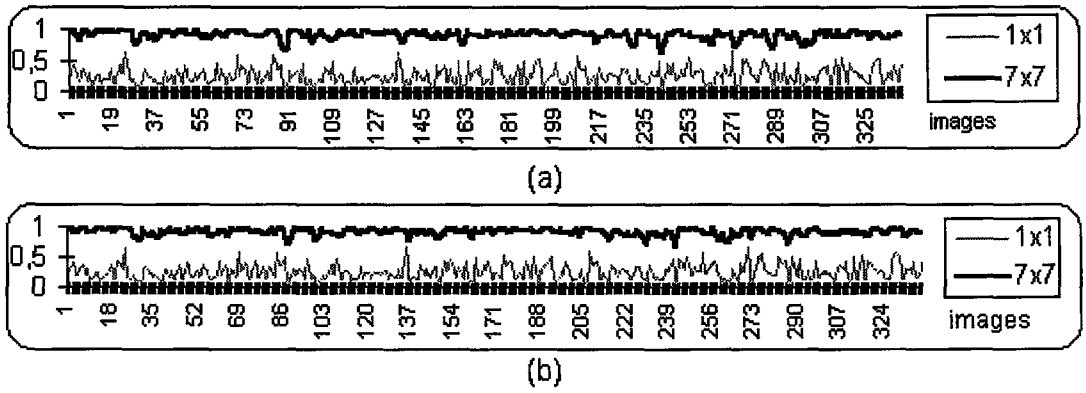


Figure 4.10: Conditional probabilities, (a) Pr(DE|IE) and (b) Pr(IE|DE) for 339 images of 113 subject faces for the illumination patterns shown in figure 4.9.

As seen in the diagram, approximately 90% of the ideal edges are detected. It can be concluded that, the edges obtained on different lighting conditions remain almost the same.

5. DECOMPOSITION OF EDGES USING PCA

An edge originates from rapid changes of gray level intensity in an image. They can also be named as boundary locations, which are high frequency components. The reasons that cause an edge to occur are changing of the surface geometry or shading due to illumination.

5.1 Edge detection

Edge detection is a challenging subject of computer vision, since a good edge detector should satisfy the following criteria [24],

- ✓ high detection performance,
- ✓ good localization,
- ✓ one response to one edge,
- ✓ robustness and applicability for different situations.

Major difficulty to satisfy the properties of a good edge detector is the presence noise introduced during imaging and sampling processes. The detection of sharp changes in image intensity values requires the computation of different derivatives of the noisy image. The numerical computation of derivatives of the noisy data, however, is an unstable process since it amplifies noise [25]. To overcome noise problem edge detection algorithms first apply noise suppression process prior to differentiation process. This can be performed by smoothing the noisy image by a lowpass filter [26],[27], or by fitting a surface or an edge template in a local neighborhood to the noisy data [28], or by fitting a surface globally across the data [29],[30].

5.1.1 Edge detection by derivation

In figure 4.1, a graphical representation and its first and second derivatives are given which corresponds to the edge locations. These derivations can be obtained by taking pixel-wise differences, or by applying a filter.

5.1.1.1 Sobel edge detector

By using $h_x = \begin{bmatrix} -1 & 0 & -1 \\ 2 & . & 2 \\ -1 & 0 & -1 \end{bmatrix}$ and $h_y = \begin{bmatrix} -1 & 2 & -1 \\ 0 & . & 0 \\ -1 & 2 & -1 \end{bmatrix}$ filters $f_x = f * h_x$ and $f_y = f * h_y$ can be obtained. Edges are detected by

$$|\nabla f(x,y)| = \sqrt{f_x^2 + f_y^2}\,, \tag{5.1}$$

operation. However a thresholding and thinning post-processes must be applied to detected edges.

5.1.1.2 Robert edge detector

It is similar to the Sobel's edge detector. The filters used are, $h_x = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ and $h_y = \begin{bmatrix} 0 & 1 & 0 \\ 0 & . & -1 \\ 0 & 0 & 0 \end{bmatrix}$. The convolution $f_x = f * h_x$ and $f_y = f * h_y$ and the gradient operation,

$$|\nabla f(x,y)| = |f_x| + |f_y|\,, \tag{5.2}$$

gives the edge map of *f(x,y)* image.

These gradient-based approaches can be improved by applying a smoothing operation via *Gaussian*,

$$g(x;\sigma) = e^{\frac{-x^2}{2\pi\sigma^2}}\,. \tag{5.3}$$

Although Gaussian can be applied prior to $|\nabla f(x,y)|$, the derivation of Gaussian can also be convolved with the image to obtain edge map as a result of the following analysis,

$$\frac{\partial}{\partial x}(f * g) = f * \frac{\partial g}{\partial x}\,, \tag{5.4}$$

$$\frac{\partial g}{\partial x} = -\frac{x}{\sigma^2} e^{-x^2/2\pi\sigma^2} \tag{5.5}$$

Gaussian filter of size 21x21 with $\sigma = 1.0$ is given in figure 5.1a.

5.1.1.3 Laplacian based edge detection

Laplacian based edge detection can be obtained by taking the second derivative of the image. Two-dimensional laplacian and edge detection is,

$$\nabla^2 f(x,y)=\frac{\partial^2 f}{\partial x^2}+\frac{\partial^2 f}{\partial y^2}, \tag{5.6}$$

$$\nabla^2 f(x,y)= f * h. \tag{5.7}$$

The *h* filter can be found as,

$$\frac{\partial f}{\partial x} \rightarrow f_x = f(n_1+1,n_2)- f(n_1,n_2), \tag{5.8}$$

$$\frac{\partial^2 f}{\partial x^2} \rightarrow f_{xx} = f_x(n_1,n_2)- f_x(n_1-1,n_2), \tag{5.9}$$

$$\begin{aligned} f_{xx} &= f(n_1+1,n_2)-2f(n_1,n_2)+f(n_1-1,n_2) \\ f_{yy} &= f(n_1,n_2-1)-2f(n_1,n_2)+f(n_1,n_2+1) \end{aligned} \tag{5.10}$$

$$\begin{aligned} \nabla^2 f \rightarrow f_{xx}+f_{yy} &= f(n_1+1,n_2)+f(n_1-1,n_2)+f(n_1,n_2+1) \\ &+ f(n_1,n_2-1)-4f(n_1,n_2) \end{aligned} \tag{5.11}$$

and resulting *h* filter is $h=\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$. $h=\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ or $h=\begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}$ are other examples of laplacian-based edge detection filters.

Edges detected by Laplacian can include many false edge contours. To determine an edge is real or not, local variance can be used. Local variance,

$$\sigma_f^2(n_1,n_2)=\frac{1}{(2M+1)^2}\sum_{k_1=n_1-M}^{n_1+M}\sum_{k_2=n_2-M}^{n_2+M}\left[f(k_1,k_2)-m_f(k_1,k_2)\right], \tag{5.12}$$

where

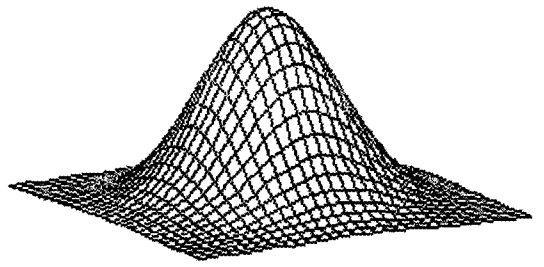
$$m_f(n_1,n_2)=\frac{1}{(2M+1)^2}\sum_{k_1=n_1-M}^{n_1+M}\sum_{k_2=n_2-M}^{n_2+M} f(k_1,k_2), \tag{5.13}$$

shows the variation in a ***MxM*** neighborhood. If this variation is high then a pre-determined threshold then it is concluded as an edge location.

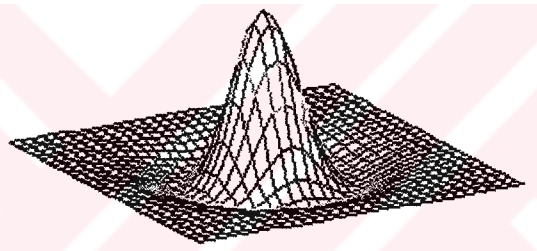
Marr and Hildreth [24] developed their edge detection approach based on laplacian and they used Gaussian for noise suppression. The application of second derivation to Gaussian is,

$$\nabla^2 g = \left(x^2 + y^2 - 2\pi\sigma^2\right)\frac{e^{-\left(x^2+y^2\right)/2\pi\sigma^2}}{\left(\pi\sigma^2\right)^2}. \qquad (5.14)$$

$\nabla^2 g$ is given in figure 5.1b.



(a)



(b)

Figure 5.1: (a) Gaussian filter of size 21x21 with $\sigma = 1.0$, (b) laplacian of the filter given in (a)

5.1.1.4 Canny edge detector

In order to obtain an optimal edge operator, Canny [27] first mathematically formulated the edge detection criteria given in section 5.1. Then he found a filter, which maximizes the product of three performance criteria. He showed that the first derivative of a Gaussian,

$$f(x) = -xe^{-x^2/2\sigma^2}, \qquad (5.15)$$

is very similar to this optimal filter. In 1D, local maxima in the convolved image, $f * I$, are edge candidates, while in 2D, local maxima along the direction normal to the edge direction are edge candidates. These local maxima are thresholded by using a hysteresis thresholding technique. In this technique there are two threshold levels, T_1 and T_2 where $T_1 < T_2$. If the gradient magnitude is greater than T_2, (x,y) is an edge point or If the gradient magnitude is greater than T_1 and is on the

connected segment of contour then (*x,y*) is edge point, otherwise (*x,y*) is not an edge point.

5.1.1.5 Generalized edge detector (GED)

The edge detector to be used in face detection must be a general-purpose edge detector, so that best presentation of edge maps can be found. Gökmen and Jain [31] proposed such a general-purpose detector called *Generalized Edge Detector*, which is later presented in 2D by Kurt and Gökmen [33].

5.1.1.5.1 Hybrid model and $\lambda\tau$ space

Regularization provides a general framework to convert ill-posed problems in early vision into well-posed problems by restricting the class of admissible solutions using constraints such as smoothness [25]. In regularization the smoothness can be imposed on the solution by minimizing energy functional containing derivatives of the solution. Given noisy data *d*(*x,y*), the regularized solution *f*(*x,y*) can be found by minimizing,

$$E_m(f;\lambda)=\iint_\Omega\left[(f-d)^2+\lambda(f_x^2+f_y^2)\right]dxdy \tag{5.16}$$

where $E_m(f;\lambda)$ is the energy functional associated with a *membrane* or by minimizing,

$$E_p(f;\tau)=\iint_\Omega\left[(f-d)^2+\tau(f_{xx}^2+2f_{xy}^2+f_{yy}^2)\right]dxdy \tag{5.17}$$

where $E_p(f;\tau)$ is the energy functional associated with a *thin plate*. The hybrid model obtained by *membrane* and *thin plate* functional is,

$$\begin{aligned}E_m(f;\lambda,\tau)=&\iint_\Omega(f-d)^2\,dxdy\\&+\iint_\Omega\lambda(1-\tau)\left(f_x^2+f_y^2\right)dxdy\\&+\iint_\Omega\lambda\tau\left(f_{xx}^2+2f_{xy}^2+f_{yy}^2\right)dxdy\end{aligned} \tag{5.18}$$

In hybrid model approach, λ determines the multiscale representation, while τ determines the continuity of the solution. Thus, pattern is separated into scale and continuity ordinates in $\lambda\tau$ space.

5.1.1.5.2 Edge detector

To determine the relation of membrane functional to convolution, let's take one-dimensional representation into consideration for simplicity. In one-dimensional case membrane functional is called string functional,

$$E_m(f;\lambda)=\int_\Omega\left[(f-d)^2+\lambda f_x^2\right]dx. \tag{5.19}$$

For boundary conditions $\left.\frac{\partial f}{\partial x}\right|_{-a,b}=0$, the Eular-Lagrange equation associated with the string functional is,

$$f-d-\lambda\frac{\partial^2 f}{\partial x^2}=0. \tag{5.20}$$

For (a, b=∞), the solution of the Eular-Lagrange equation is,

$$f(x)=\int_{-\infty}^{\infty}\frac{1}{2\lambda}e^{-|x-x'|/\lambda}d(x')dx', \tag{5.21}$$

and its impulse response is first order regularization filter [29][32][33],

$$R_1(x;\lambda)=\frac{1}{2\lambda}e^{-|x|/\lambda}. \tag{5.22}$$

Two-dimensional regularization filter is easily realized as follows,

$$R_1(x,y;\lambda)=\frac{1}{2\lambda}e^{\frac{-(|x|+|y|)}{\lambda}}. \tag{5.23}$$

In the same manner, *thin plate* functional in one dimension is,

$$E_p(f;\tau)=\int_\Omega\left[(f-d)^2+\tau f_x^2\right]dx. \tag{5.24}$$

where its corresponding regularization filter is a second order regularization filter,

$$R_2(x;\lambda)=\frac{1}{2\lambda^{1/4}}e^{-|x|/\sqrt{2}\lambda^{1/4}}\cos\left(\frac{|x|}{\sqrt{2}\lambda^{1/4}}-\frac{\pi}{4}\right). \tag{5.25}$$

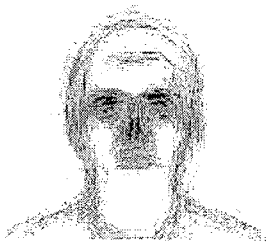
First and second derivatives of R_1 and R_2 are used as the edge detectors in *Generalized Edge Detector.*

5.2 Principal components of edges

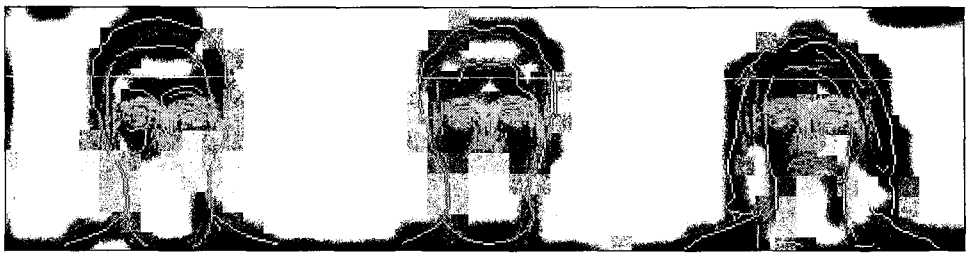
According to the analysis given in section 4, edge representations of images are more robust to changes in visual stimuli theoretically. Based on this fact, pure edge based approach can be considered as a face recognition scheme.

As outlined in section 3, determining principal components of a face can be applied to edge map representation simply by inputting edge maps, generated by *generalized edge detector*, to the recognition system instead of using face images directly. In this approach, eigenvectors of covariance matrix constructed from edge maps are called "*eigenedges*". Figure 5.2a shows arithmetic mean of edge maps, figure 5.2b shows sample caricature edges and figure 5.2c shows *eigenedges* of the learning set given in *Appendix A* in descending order of eigenvalues.

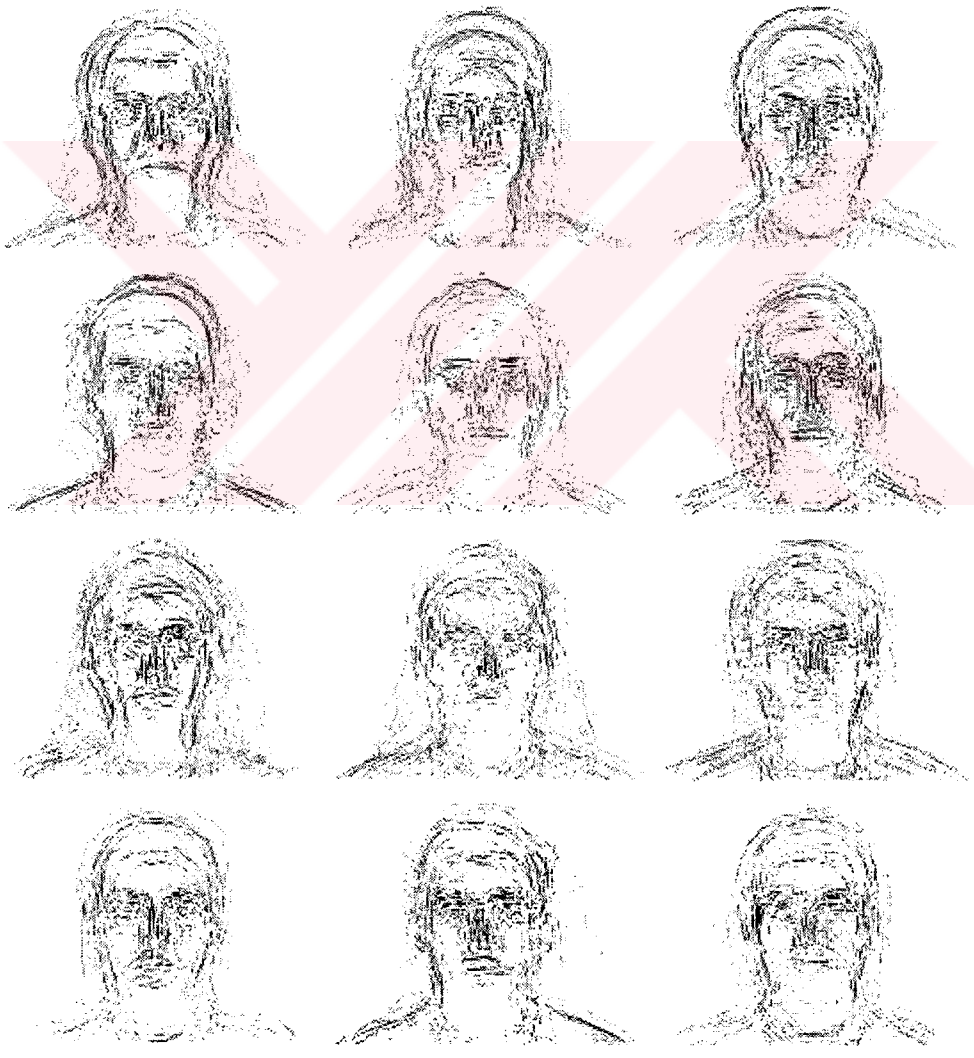
It can be observed that, edges scatter higher dimension in *edge space* compared to *face space*. This seems as a drawback of the system, however this increased dimensionality can be compensated by using higher number of eigenvectors to include as much variation as possibly.



(a)



(b)



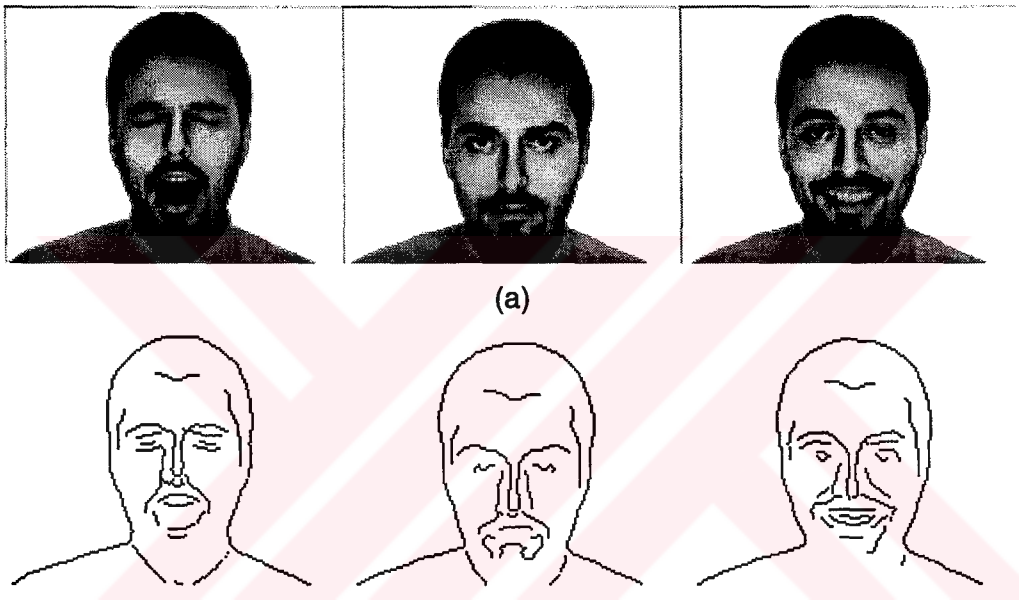
(c)

Figure 5.2: Eigenedges in descending order according to eigenvalues.

5.3 Drawback of *eigenedges*

The most important drawback of eigenedge approach is the locality of edges. Facial expression changes or shifts in edge locations due to small rotation, non-ideal alignment or changed shadowing will result in recognition performance degradation. In figure 5.3 facial expressions from the test set are given.

On the other hand, if the images are taken in strictly controlled manner (face is in an upright position and the facial expression is the same as in the learning set), eigenedge system will give acceptable performance. However, it is not guarantied to attain strict control on the orientation and expression of face.



(a)

(b)

Figure 5.3: (a) Different facial expressions (screaming, angry, laughing), (b) corresponding edge maps.

6. AN ALTERNATIVE: EIGENHILLS

Due to the problems outlined in section 5.3, the solution discussed in section 4.4 can be applied. In this study this solution is carried out by a regularization theory approach, membrane covering, which corresponds to applying the functional,

$$E_m(f,\lambda)=\iint_\Omega (f-d)^2\,dxdy+\iint_\Omega \lambda(1-\tau)\left(f_x^2+f_y^2\right)dxdy\,, \qquad (6.1)$$

to the edge map. Minimizing the Eular-Lagrange equation, which associated with this functional, yields second-order regularization filter outlined in section 5.1.1.5.1. Two-dimensional R_1 filter is shown in figure 6.1.

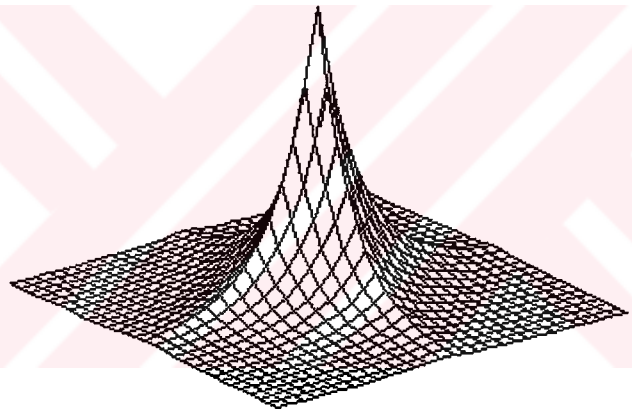
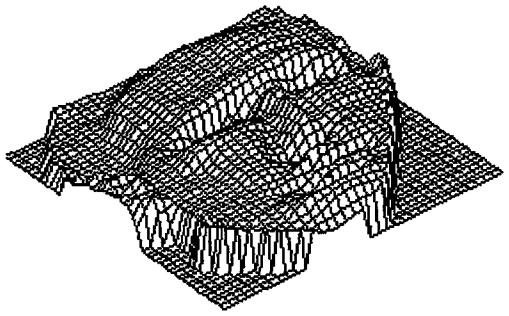


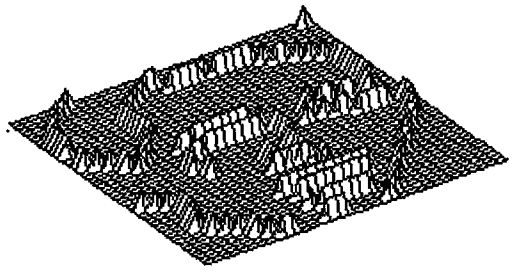
Figure 6.1: Two-dimensional, first order regularization filter, R1, of size 37x37 and λ=2.

The basic characteristic of this filter is its separability. After convolving, it also provides the edge locations to be uniquely identified, since the local maxima of the edges are preserved. Gaussian and filters alike (similar to covering thin plate over the edge map) do not support the idea of strong preservation of the edges. As seen from the Gaussian filter given in figure 5.1a, the smoothing operation makes neighboring edge pixels to appear as edge locations.

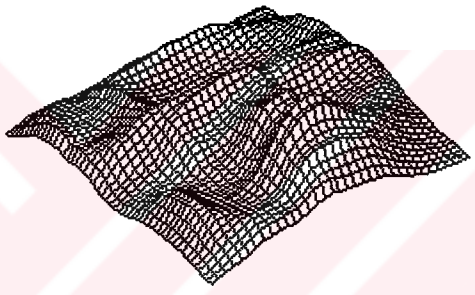
The smoothed version of the edges is called "hills". In figure 6.2, 3D representations of grey level intensities of a face, edges and "hills" is given. In figure 6.3, three different facial expressions from test set and their hills are given.



(a)

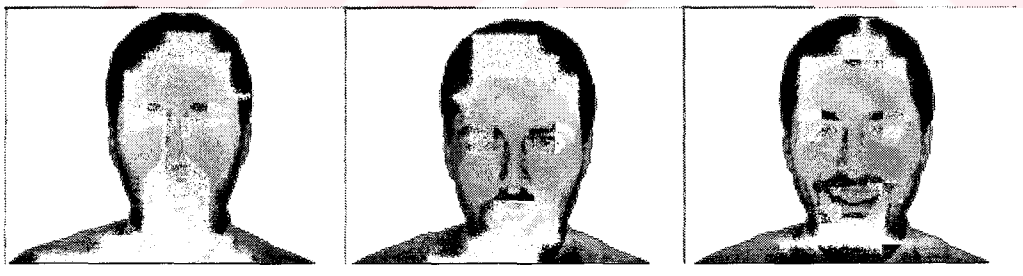


(b)

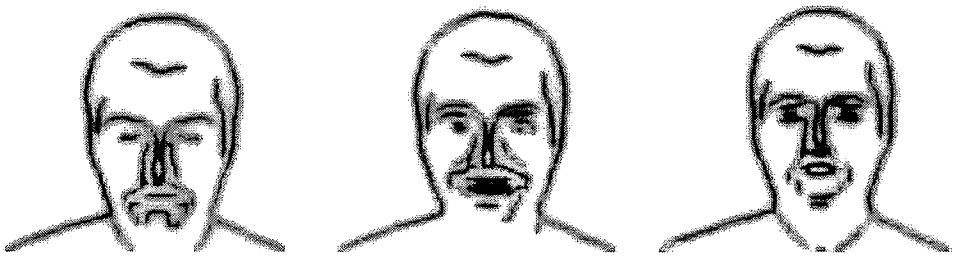


(c)

Figure 6.2: (a) Grey level intensities, (b) edges, (c) hills obtained by R1.



(a)



(b)

Figure 6.3: (a) Selected faces from test set, (b) hills of each image.

6.1 Face recognition

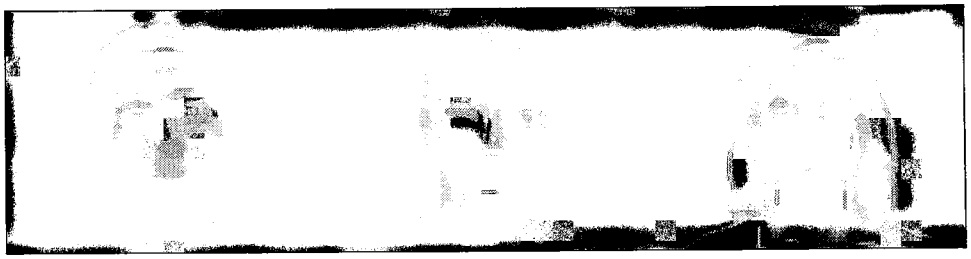
The principal components of the hills are obtained as follows:

- Find the mean hill by $\Psi = 1/T \sum_{i=1}^{T} \zeta_i$, where ***T*** is number of faces in learning set, ζ is the hill images,
- Calculate caricature hill, $\theta_i = \zeta_i - \Psi$,
- Determine covariance matrix, $C = 1/T \sum_{i=1}^{T} \zeta_i \zeta_i^T$,
- Figure the eigenvectors of the covariance matrix, and select highest ***M*** maximum eigenvalued eigenvectors as principal components.
- Evaluate weight vectors of each face in learning set by $w_k = u_k^T (x_i - \Psi)$, where w_k is the weight, x_k is the face from learning set.

When a new face is presented to the system, first its hill is obtained, then the hill is projected to the low dimensional feature space, then distance to each face in learning set is calculated, smallest of them is the recognized face.

Contrary to pure edge based recognition, the dimensions of the feature space is smaller in this new approach, because applying R_1 filter increases the correlation among the images. In figure 6.4a, mean hill image, in 6.4b, sample caricature hill images and in 6.4c, eigenvectors that increase the variation of the hills, also called *eigenhills*, are given.

(a)



(b)

(c)

Figure 6.4: (a) Average hill, (b) Caricature hill, (c) Eigenhills in descending order according to eigenvalues.

6.2 Advantages of eigenhills over eigenface and eigenedge

The problem associated with the eigenface approach is obtaining different texture for different illuminations. To overcome this problem, edge maps, which is another representation of images is used. However, this new representation brings new problems associated to the structure of the edges.

In this study, a new, connectionist approach (eigenfaces + eigenedges) is proposed. In addition, to overcome the locality of the edges for increased face recognition purposes, covering a membrane over the edges is discussed. The advantage of the new system is its combinatory nature. Eigenhills combines the advantages of PCA based face recognition, while solving the illumination problem by introducing spread edge maps.

7. EXPERIMENTAL RESULTS

The goal of this study is to develop a new approach to face recognition paradigm, which solves one of the most important problems associated with the famous *eigenfaces* approach to face recognition. Hence, this section is the core of the study for presenting the advantages of the new approach, *eigenhills*.

7.1 Software

To perform this comparison, software developed on Windows 98©, running on 333 MHz Pentium II© with 64 Mbytes of RAM is used. The software supports bitmap, and hips file formats. The code of the software is written pure object oriented on Borland C++ Builder 3.0©. There are four main classes corresponding to face images, and recognition methods. The interface of the software is given in figure 7.1.

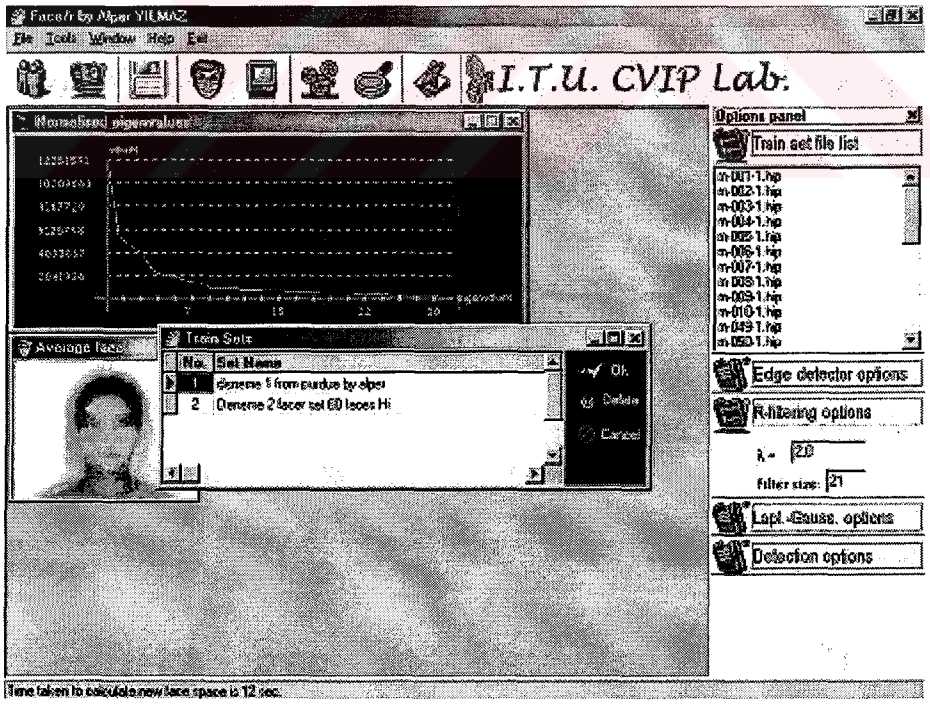


Figure 7.1: Interface of the designed program.

7.2 Quantitative evaluation criteria

The quantitative evaluations of the three approaches are obtained by using two error criteria. First criterion is the measure used in recognition,

$$\varepsilon = \left\| W_l - W_t \right\|_2 , \tag{7.1}$$

where the sub-scripts ***l*** and ***t*** corresponds to *learning set* and *test set* respectively. Second criterion is the measure used to determine how far the reconstructed face, edge or hill to the original one in the learning set,

$$\varepsilon = \left\| \overline{\Gamma} - \Gamma_i \right\|_2 , \tag{7.2}$$

where $\overline{\Gamma}$ is reconstructed face and Γ_i is recognized face.

In the following sections a detailed comparison of the virtual and real world performances for the three approaches will be given.

7.3 Results on simulated faces

Along with the designed program, Maya® running on Silicon Workstation is used for designing controlled simulation based experiment. A wire-frame face image is modified to obtain 7 simulated face with lambertian surfaces. The faces illuminated with an initial light source are shown in figure 7.2.

The goal to test each system on simulation conditions was to see, whether the performance of eigenhills will support the outcome discussed in section 4.4. Since the simulated faces are designed on a strictly controlled environment, there is no non-ideal alignment of edges due to change of location of the head blob. So that the performance comparison would be made solely on varying shading patterns of the subject faces resulting from the lighting conditions shown in figure 4.7.

The comparison for the virtual faces is acquired by measuring the distance of the reconstructed face/hill to the recognized face in the face/hill space. Resulting diagrams for face space distance and weight distance are given in figure 7.3a and b respectively.

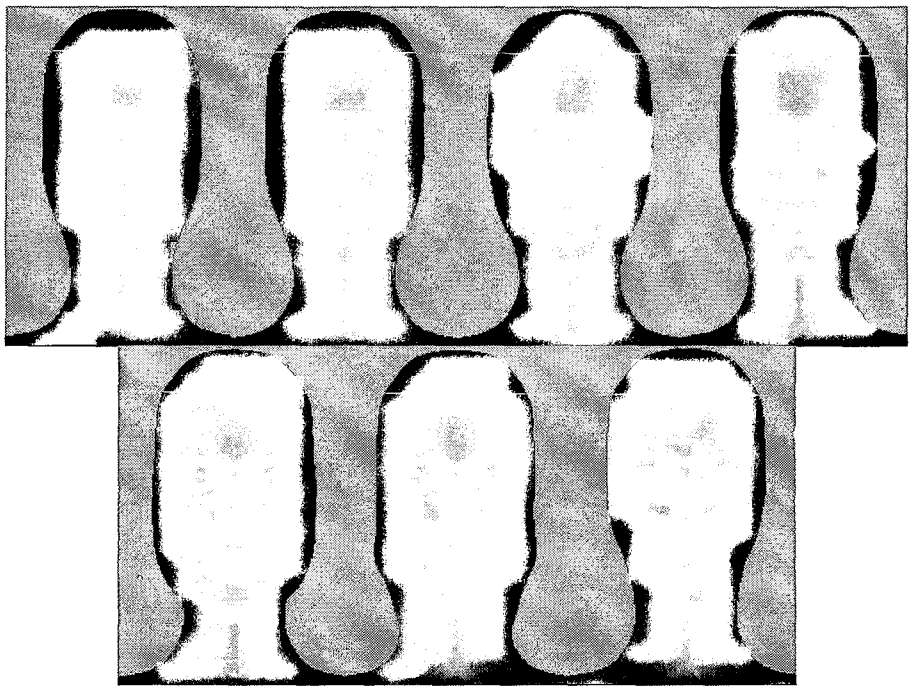
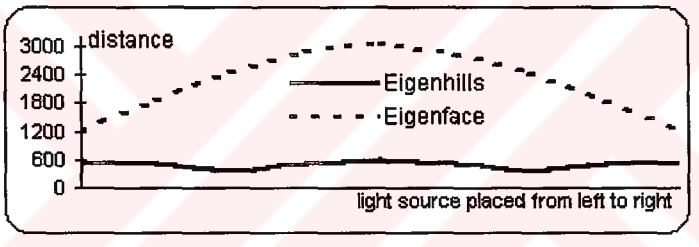
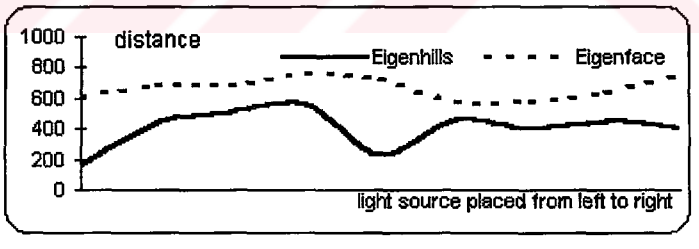


Figure 7.2: Simulated faces designed by using Maya® software running on Silicon Workstation.



(a)



(b)

Figure 7.3: (a) Normalized distance to face space of reconstructed image, (b) normalized weight distances between the recognized face and the input face.

Since the light sources are arranged on the contours of a circle and the face symmetric, the distance between the reconstructed face from the test set and the corresponding face of the learning set is symmetric for eigenface and eigenhills approaches. Figure 7.3a reveals that the hills reconstructed are more similar to the hills of the learning set than the reconstructed face images to the faces of the learning set. The distance in weights, seen in figure 7.3b indicates that hills

representation is more robust than texture representation of face images for recognition purposes.

7.4 Results on face database

To observe if the performance for simulated faces holds for the real world images, three approaches were tested by using Purdue A&R face database.

This face database was created by Aleix Martinez and Robert Benavente at the Computer Vision Center (CVC) of Purdue University. It contains over 4,000 images corresponding to 126 people's faces (70 men and 56 women). Images feature frontal view faces with different facial expressions, illumination conditions, and occlusions (sunglasses and scarf). The pictures were taken at the CVC under strictly controlled conditions. No restrictions on wear (clothes, glasses, etc.), make-up, hairstyle, etc. were imposed to participants. Each person participated in two sessions, separated by two weeks (14 days) time. The same pictures were taken in both sessions.

The images for each individual includes,

- Neutral expression,
- Smile,
- Anger,
- Scream,
- Left light on,
- Right light on,
- All side lights on,
- Wearing sun glasses,
- Wearing sun glasses and left light on,
- Wearing sun glasses and right light on,
- Wearing scarf,
- Wearing scarf and left light on,
- Wearing scarf and right light on.

The face database is divided into two sets: learning set and test set. In figure 7.4a, sample images for learning and in figure 7.4b sample from test set is given. Learning set is composed of a total of 126 individual face images obtained in natural lighting conditions (given in appendix A). Test set is constituted of face images

photographed when left light was on, right light was on, all sidelights were on, along with images including three facial expressions: laughing, angry and screaming.

(a)

(b)

Figure 7.4: (a) Female image from learning set, (b) images from test set.

The quantitative comparisons for the real world images are demonstrated by calculating reconstruction errors and recognition percentages. The performances of the three approaches for the six test categories of figure 7.4b are shown in figures 7.5, 7.6 and 7.7.

7.5 Performance comparisons

To demonstrate the responds of the three approaches to illumination changes, three analysis are made. In the diagrams to demonstrate the comparison of the approaches, ***bold line refers to eigenhill method,*** *gray refers to eigenedge method and black line refers to eigenface method.* The first experiment is held for the images taken under all sidelights are on.

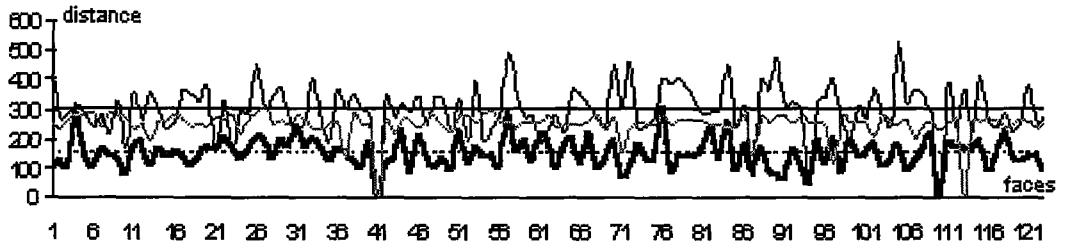
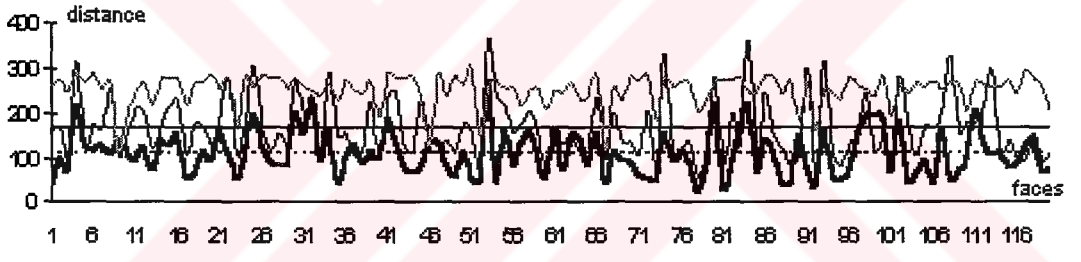
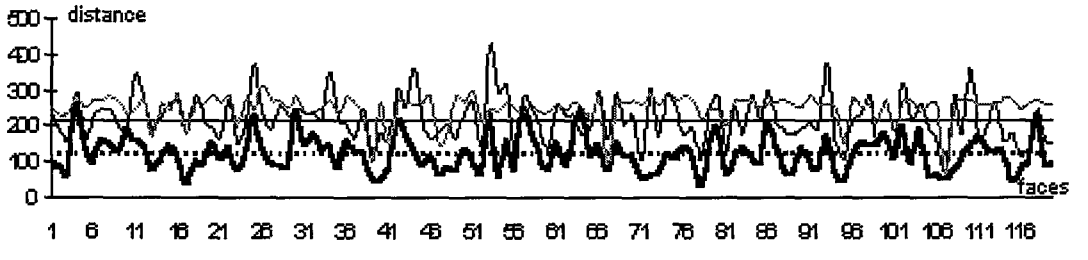


Figure 7.5: Comparison chart for all side lights on.

In figure 7.5, where both sidelights are on and the shading of the face is far beyond the case of natural lighting, it is observed that the performance of eigenhill approach, which is based on spread edge representation, is not effected immoderately. However, since there is the effect of shadowing on edge maps and non-ideal alignment of the heads, eigenedge approach wasn't as successful as eigenhills. It can be argued that non-ideal alignment can be overcome by applying a post process on edge maps using distance maps, however since the time cost of this process is excessive and it is not a necessity for eigenhills' spread edge representation, it is out of this study's scope.



(a)



(b)

Figure 7.6: Comparison chart when (a) left sidelight is on, and (b) right sidelight is on.

When the both sidelights are on, increased brightness coarsely changes the texture of the face. However, when one light source is turned on, whether it is on the left side or the right side light, the change in texture is less than the previous case. On the other hand, turning on one of the sidelights at an instant of time introduces its own problems. New edges resulting from the convex shape of the face result in a

decline in the performance of the eigenedge approach. For example, the nose produces cast shadows or symmetric line of the face from forehead to chin results as an edge. In figure 7.6a and b, the quantitative comparisons for three approaches are demonstrated. The eigenhills approach outperforms other two methods.

The outcome of the comparisons made for lighting condition demonstrates that proposed approach to face recognition, eigenhills, is more robust than the eigenface or eigenedge approaches.

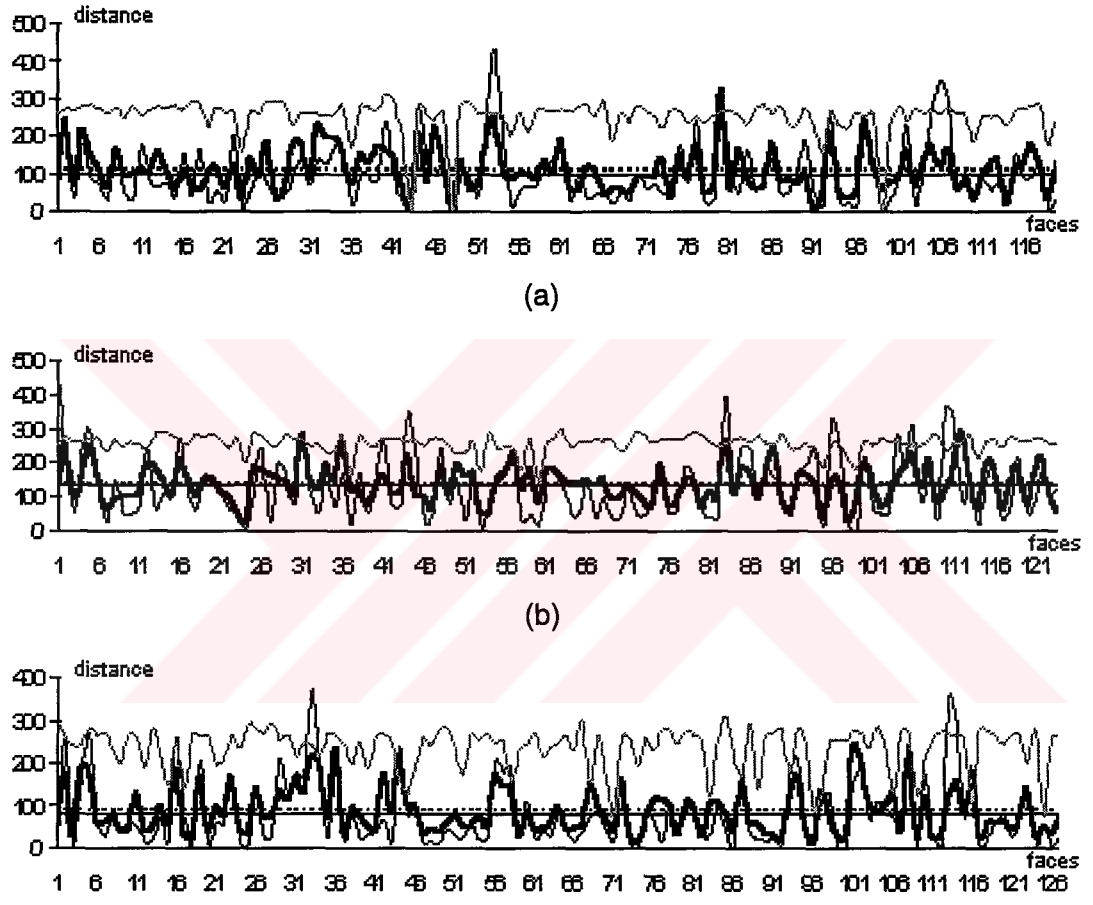


Figure 7.7: Comparison charts for (a) angry, (b) screaming and (c) laughing facial expressions.

To conclude whither the performance of eigenhills is acceptable as compared to other two approaches', we had to check the case for various facial expressions. In the test set, which is constructed from A&R face database, there are three categories for this purpose: angry, laughing and screaming. The quantitative comparisons for these three cases are shown in figure 7.7a, b and c. As it was expected, eigenedge's performance is very poor in all three categories. This degraded performance is related to additional or lacking edges around the mouth,

nose and eyes. On the other hand, eigenhills and eigenfaces have given similar performances.

For intuitive comparison, arithmetic mean of the errors for the six categories of the test set is given in table 7.1. The results are obtained from the data in figure 7.5, 7.6a-b and figure 7.7a-c. From table 7.1, it can be interpreted that eigenhills approach has outperformed eigenface and eigenedge approaches for various illumination conditions, while the performances of eigenface and eigenhills are similar for different facial expressions. This result was expected since eigenhills approach was argued to combine the benefits of the edge based representation of images and the eigenface approach to face recognition.

Table 7.1: Average values of errors for six test categories of the eigenface, eigenedge and eigenhill methods.

	All light sources	Left light source	Right light source	Screaming	Laughing	Angry
Eigenface	300,5	166,9	219,3	131,5	79,6	98,4
Eigenedge	250,8	251,15	250,7	258,4	236,7	252
Eigenhill	151,4	108	120,31	140,75	86,35	109,63

Correct recognition performances for the three systems are given in table 7.2. It is observed from table that, eigenhill method has an average of ***86.4%*** recognition performance for all three lighting conditions, while eigenface method has only ***69.8%***. Eigenedge method, however, has the poorest recognition performance because of the reason stated above. Not interestingly, the average recognition performance for the facial expression changes for eigenhill is ***88.8%***, which is a little above eigenface's performance, ***88.4%***.

Table 7.2: Recognition percentages of eigenface, eigenedge and eigenhill methods for six categories of the test set.

	Both light sources	Left light source	Right light source	Screaming	Laughing	Angry
Eigenface	44,7	85,8	79,1	83,8	92,9	89,1
Eigenedge	18,7	34,2	27,5	16,7	42,5	30
Eigenhill	80,2	91,6	87,5	79,8	95	91,6

Note that, overall recognition performance of eigenhill is ***89.4%***, which is above eigenface's overall recognition performance, ***82.3%***. Also, the overall performance of the eigenedge method, ***38.8%***, is far below the acceptable range.

8. CONCLUSION

From theoretical analysis of the illumination effect on texture maps and edge maps, it is proved that edges are more robust representations for subject images rather than the gray level intensity. Based on this proof, a new face recognition algorithm, named "eigenhills", is proposed, and it is shown that eigenhills approach is more tolerant to varying lighting conditions and different facial expressions as compared to eigenface and eigenedge approaches.

To demonstrate the performances of eigenhills, eigenface and eigenedge methods experiments on simulated and real life face images are held. Simulation experiments are made on 7 subject wire-frame faces in a strictly controlled environment utilizing 7 different lighting conditions, while real life experiments are made on 126 subject face images taken from Purdue A&R face database utilizing a total of 3 different lighting conditions and 3 different facial expressions. Experimental results show that eigenface's recognition performance degraded rapidly for different illumination patterns, however, eigenhill's recognition performance remained considerably high for such degradations.

As a conclusion, spread edge profiles, "hills", reduce the effect of shifts and changes in edge locations caused by facial expressions, illumination variations and non-ideal placement of head in the image and eigenhills approach is a more robust face recognition algorithm than eigenface and eigenedges.

9. REFERENCES

[1] **Carey, S., Diamond, R.,** 1977. "From Piecemeal to Configurational Representation of Faces," *Science*, vol. 195, pp. 312-313.

[2] **Bledsoe, W. W.,** 1966. "The Model Method in Facial Recognition," *Panoramic Research Inc. Paolo Alto, CA.*, Rep. PRO: 15.

[3] **Bledsoe, W. W.,** 1966. "Man Machine Facial Recognition," *Panoramic Research Inc. Paolo Alto, CA.,* Rep. PRO: 22.

[4] **Fischler, M. A., Elschlager, R. A.,** 1973. "The Representation and Matching of Pictorial Structures," *IEEE Transactions on Computers, c-22*, (1).

[5] **Yuille, A. L., Cohen, D. S. and Hallinan, P. W.,** 1989. "Feature Extraction from Faces Using Deformable Templates," *Proceedings of CVPR.*

[6] **Kohonen, T., Lehtio, P.,** 1981. "Storage and Processing of Information in Distributed Associative Memory System." in G. E. Hinton and J. A. Anderson (Eds.). *Parallel models of associative memory.* Hillsdale, NJ, Fulbaum, pp. 105-143.

[7] **Flemming, B., Cottrell, G.,** 1990. "Categorization of Faces Using Unsupervised Feature Extraction," *Proceedings of IJCNN,* vol. 2.

[8] **Stonham, T. J.,** 1986. "Practical Face Recognition and Verification with WISARD." in H. Ellis, M. Jeeves, F. Newcombe and A. Young (Eds.), *Aspects of Face Processing.* Dordrecht: Martinus Nijhoff.

[9] **Kaya, Y., Kobayashi, K.,** 1972. "A Base Study of Human Face Recognition." in S. Vatanabe, (Ed.). *Frontiers of Pattern Recognition*, Academic Press New York.

[10] **Canon, S. R., Jones, G. W., Campbell, R. and Morgan, N. W.,** 1986. "A Computer Vision System for Identification of Individuals," *Proceedings of IECON,* vol. 1.

[11] **Craw, Ellis and Lischman,** 1987. "Automatic Extraction of Face Features," *Pattern recognition letters,* vol. 5, pp. 183-187.

[12] **Wong, K., Law, H. and Tsaug, P.,** 1989. "A System for Recognizing Human Faces," *Proceedings of ICASSP,* pp. 1638-1642.

[13] **Kanade, T.,** 1973. "Picture Processing and Recognition of Human Faces," *Dept. of Information Science, Kyoto University*, Japan.

[14] **Kirby, M., Sivorich, L.,** 1990. "Application of Karhunen-Loeve Procedure for The Characterization of Human Faces," *IEEE Trans. On PAMI.* Vol. 12(1).

[15] **Turk, M., Pentland, A.,** 1991. "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience,* Vol. 3, Num. 1, pp. 71-86.

[16] **Nayar, S. K., Murase, H. and Nene, S. A.,** 1996. "Parametric Appearance Representation." in S. K. Nayar, T. Poggio (Eds.), *Early visual learning*, Oxford University Press USA, pp. 131-160.

[17] **Moghaddam, B. and Pentland, A.,** 1996. "Probabilistic Visual Learning for Object Representation." in S. K. Nayar, T. Poggio (Eds.), *Early Visual Learning*, Oxford University Press USA, pp. 99-130.

[18] **P.N. Belhumeur, J.P. Hespanha and D.J. Kriegman,** 1997. "Eigenfaces vs. Fischerfaces: Recognition Using Class Specific Linear Projection," *IEEE Trans. on PAMI,* Vol. 19, No. 7.

[19] **Takàcs, B., Weschler, H.,** 1998. "Face Recognition Using Binary Image Metrics," *Proceedings of the 3rd Int. Conf. On Face and Gesture Recognition*, Nara, Japan, pp. 294-299.

[20] **Jae S. Lim,** 1990. Two-dimensional Signal and Image Processing, Prentice Hall, NY.

[21] **H. Abdi,** 1988. "A Generalized Approach for Connectionist Auto-associative Memories: Interpretation, Implication & Illustration for Face Processing." in J. Demongeot, et al. (Eds.) *Artificial Intelligence and Cognitive Sciences*, Manchester University Press, pp. 149-165.

[22] **G. H. Dunteman,** 1989. Principal Components Analysis, Sage publications.

[23] **D. Valentin, H. Abdi, A. J. O'Toole,** 1994. "Categorization and Identification of Human Face Images by Neural Networks: A Review of the Linear Auto-associative and Principal Component Approaches, *Journal of Biological Systems*, vol. 2, pp. 413-429.

[24] **Marr, D., Hildreth, E.,** 1980. "Theory of Edge Detection," *Proceedings of Roy. Soc.,* (Sec. B, 207), pp. 187-217.

[25] **Bertero, M., Poggio, T. and Torre, V.,** 1987. "Ill-posed Problems in Early Vision," *Technical Report, MIT AI Lab.,* AI. Memo 924.

[26] **Marr, D., Hildreth, E.,** 1980. "Theory of Edge Detection," *Proceedings of Roy. Soc.,* (Sec. B, 207), pp. 187-217.

[27] **Canny, J. F.,** 1986. "A Computational Approach to Edge Detection," *IEEE Trans. on PAMI,* Vol. PAMI-8, pp. 679-698.

[28] **Haralick, R. M.,** 1984. "The Digital Step Edge from Zero Crossings of Second Directional Derivatives," *IEEE Trans. on PAMI,* Vol. PAMI-61, pp. 58-68.

[29] **Blake, A., Zisserman, A.,** 1987. Visual Reconstruction, MIT Press, London.

[30] **Terzopoulos, D.,** 1988. "The Computation of Visual Surface Representations," *IEEE Trans. on PAMI,* Vol. PAMI-10, pp. 417-438.

[31] **Gökmen, M., Jain, A. K.,** 1997. "$\lambda\tau$-Space Representation of Images and Generalized Edge Detector," *IEEE Trans. on PAMI,* Vol. 19, No: 6, pp. 545-563.

[32] **Gökmen, M., Li, C. C.,** 1992. "Multiscale Edge Detection Using First Order R-filter," *Proceedings on ICPR*, Holland, pp. 307-310.

[33] **Kurt, B., Gökmen, M.,** 1999. "2D Generalized Edge Detector," *Proceedings of EEE National Conf. on Signal Processing and App.*

10. APPENDIX A: LEARNING SET

Figure A.1: Learning set, males.

Figure A.2: Learning set continued, females.

11. APPENDIX B: LEARNING SET EDGE MAPS

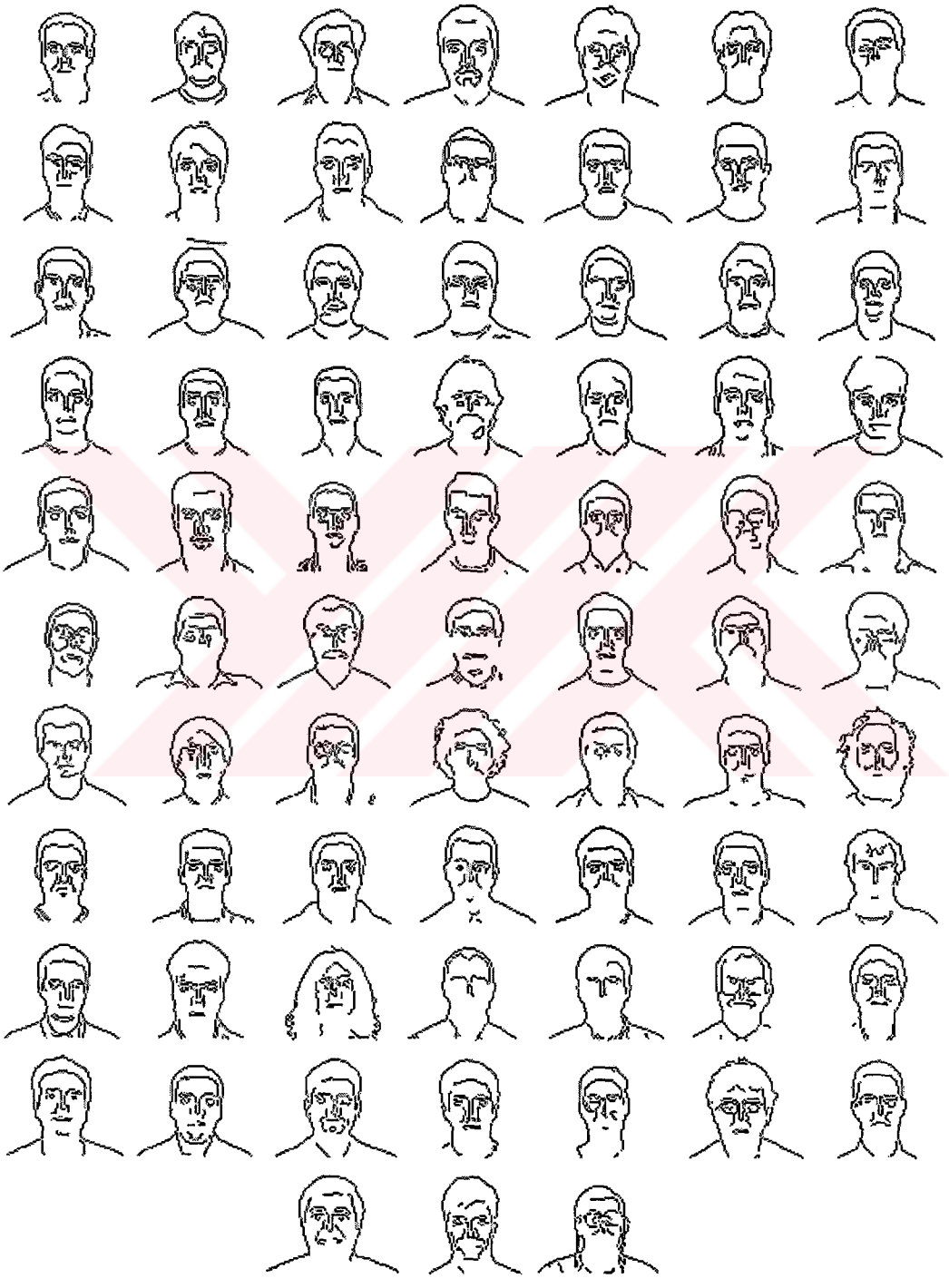
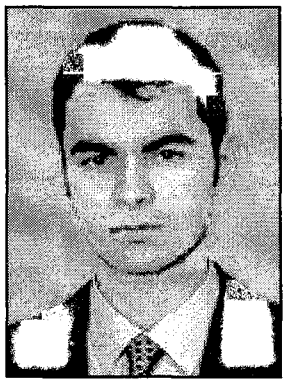


Figure B.1: Learning set edge maps, males.

Figure B.2: Learning set edge maps continued, females.

12. BIOGRAPHY



Alper YILMAZ has received his B.Sc. in Computer Science from Yıldız Technical University, Istanbul, Turkey in 1997 with the highest honors. He had the highest GPA (A) in all Faculties and Schools of Yıldız Technical University and graduated with number one ranking. He has received his M.Sc. in Computer Engineering from Istanbul Technical University, Istanbul, Turkey in 2000 with GPA of A. He is currently admitted to University of Central Florida, Orlando, USA, as Ph.D. candidate in Computer Science.

He has been with the Computer Vision and Image Processing Laboratory of Istanbul Technical University since 1998 as researcher on Pattern Recognition and has various printed papers on face recognition in international and national conferences and Journals. He has earned several scholarships including Turkish Informatics Foundation's one-year high academic performance scholarship. His present research interests include pattern recognition, image processing, neural networks, computer vision, cryptography and watermarking of images.

Alper YILMAZ is currently a member of IEEE Society, IEEE Computer Society, IEEE Signal Processing Society and International Association of Pattern Recognition (IAPR) Society.