CAR-TR-695 DACA76-92-C-0009
CS-TR-3176 November 1993

HUMAN FACE SEGMENTATION AND IDENTIFICATION

Saad Ahmed Sirohey

Computer Vision Laboratory
Center for Automation Research
University of Maryland
College Park, MD 20742-3275

Abstract

This thesis considers segmentation and identification of human faces from grey scale images with clutter. The segmentation developed utilizes the elliptical structure of the human head. It uses the information present in the edge map of the image and through some preprocessing separates the head from the background clutter. An ellipse is then fitted to mark the boundary between the head region and the background. The identification procedure finds feature points in the segmented face through a Gabor wavelet decomposition and performs graph matching. The segmentation and identification algorithms were tested on a database of 48 images of 16 persons with encouraging results.

The support of the Advanced Research Projects Agency (ARPA Order No. 8459) and the U.S. Army Topographic Engineering Center under Contract DACA76-92-C-0009 is gratefully acknowledged, as is the help of Sandy German in preparing this paper.

Dedication

To my grandmother

Masooda Begum

and my parents

Iftikhar A. Sirohey

and

Zarina Sirohey

Acknowledgements

I would like to take this opportunity to thank all the people who have been instrumental the accomplishment of this task. First of all Professor Rama Chellappa, with his superb guidance and support during the course of this thesis. Dr. Q. Zheng for his time and effort and the lively discussions that we had concerning the topic.

I would also like to thank Sandy, Ting, Micheal, Phillipe, Oh-jin, Bruce, Shridhar, Shyam, Claudia, Charles, Murad and David for putting up with me for the last year. Our discussions together brought new insight for me regarding the research.

And of course this work would not have been possible if not for the undying love, support and encouragement of my parents, Iftikhar A. Sirohey and Zarina Sirohey. To them any gratitude I may express will not be enough.

Contents

List of Figures

Chapter 1

Introduction

Every day humans and animals (to an extent) employ visual recognition in their affairs. We as human beings have become very adept in this form of recognition. The human brain can easily identify scores of different faces, some seen in day to day life and others seen over varying durations of time. It has been the goal of scientists for decades to try to determine the method that produces this remarkable ability that we as humans take for granted. The advent of computers has prompted a new cadre of scientists and engineers to embark on the task of having a computer emulate such visual recognition. Granted that humans use other stimuli than vision for the task, the primary one is of course vision.

The benefits of computer based face identification can be manifold, They range from a computer terminal being able to identify the person sitting in front of it, to applications in law enforcement (identification of criminals, etc.), to surveillance by customs for immigration purposes and for verification of identity at security sites, as well as a possible application to verification at Automated Teller Machines (ATM). In many instances a real time recognition system is required for these applications. With advances in parallel computers such real time identification systems are not an impossibility.

The design of a face recognition system is heavily dependent on the nature of the input to the system. These face image inputs can range from images taken in a controlled environment, i.e. target asked to pose for a picture with a predetermined background and illumination, to an uncontrolled situation where background and illumination are variables across images, i.e. target's image taken surreptitiously without the target being aware of it, and with background and illumination depending on where the instance occurred. Another issue of importance is that of robustness. Robustness encompasses a plethora of factors ranging from correct identification of subject under normal conditions to attempted recognition when the subject is disguised. There is also the issue of the time required by the identification system. This requirement changes with the use of the recognition system. However, for many applications it is necessary to have a real-time recognition system. A tried and tested approach to real-time computation is parallel computing. It is, therefore, important that whatever algorithm is developed for identification, it should be easily transformed into a parallel form.

Most model-based applications, especially computer graphics, use some sort of analytical primitives to construct shapes/objects. These shapes/objects have an underlying artificial construction and can be very easily represented by analytical expressions. On the other hand natural/biological objects have structures that cannot be easily (if at all) described by such expressions. This fact makes a computer aided recognition system very complex. A trivial solution to the problem would be to match, pixel by pixel, a target image to its stored counterpart in a database. This would require large image databases and would have to enforce the condition that the target image, to be recognized, be present in the database.

A more robust technique would involve locating and identifying "salient" features in the target image and then comparing them to corresponding features in a database. The problem is how to determine what these "salient" features in an image are that provide adequate information for a recognition system to be efficient.

One such feature based approach, utilizing a bank of dilated Gabor wavelet filters, has been used in various image understanding/recognition algorithms. Applications of these algorithms range from image registration [16] to motion analysis [Wu & Chellappa] to a face identification system [10].

Intuitively it can be argued that for any system performing face recognition, the first step would be locating the face in the image. Finding the face gives the recognition system a specific area in the image to work on. Under the constraints of real-time implementation, it is required that the algorithm for finding the face be fast. Hence the algorithm developed for locating the face should have the property of being easily transformable into a parallel structure.

In Manjunath and Chellappa [10], the image database used for the face identification system consists of face images with a uniform background. This forces the feature points to be located within the head/shoulder region. Since no prior consideration is given to locating the detected features in the face region, images which include part of the upper chest/shoulder area will contain feature points in those areas too. The purpose of this thesis is to extend the face identification system of [10] to a more general class of input images, namely, images that need not necessarily have uniform backgrounds, and also to maintain the integrity of the face identification system by segmenting the face region and using it for the purpose of recognition. In this connection, a new face image database was obtained from MIT. This new database consists of face images with moderately cluttered backgrounds. If the algorithm in [10] is used without any preprocessing on this database, feature points are detected which are outside the face region. An attempt is made to localize the feature points within the face region of the image. To accomplish this, it is required to segment the image into face region and non-face region.

The problem is now confined to segmentation of the face region from the image. Although we have stated that it is not possible to accurately describe a natural object (i.e. human face) using analytical functions, it can still be fruitful to use the underlying elliptical structure of the head and to approximate it as an ellipse.

The approach used in this thesis relies on the underlying elliptic nature of the face. The edge map is processed to find contiguous edge segments. These edge segments are then paired with other edge segments and fitted to a linearized equation of the ellipse. The parameters of the ellipse, i.e. semi-major and semi-minor axes and center point, are found by using the pseudo inverse method for an over-determined system of equations. After all possible pairs of segments are fitted to the ellipse, classification is used to determine parameters which are of interest. A reverse procedure is employed to fit segments to the parameters of interest. Finally a voting method is used to find the best parameter values, i.e. those which have the most segments, compensated for their sizes, fitted to them.

When the approach was tested on a database of 48 images, more than 80% correct segmentation was achieved. Since we had available to us the recognition algorithm of Manjunath and Chellappa [10], it was used to test the hypothesis that segmentation of the face first will improve the recognition rate. Without segmentation, an accuracy of less than 50% was achieved. This improved to more than 70% with the segmentation algorithm. A schematic diagram of the system with segmentation is depicted in Figure 1.1.

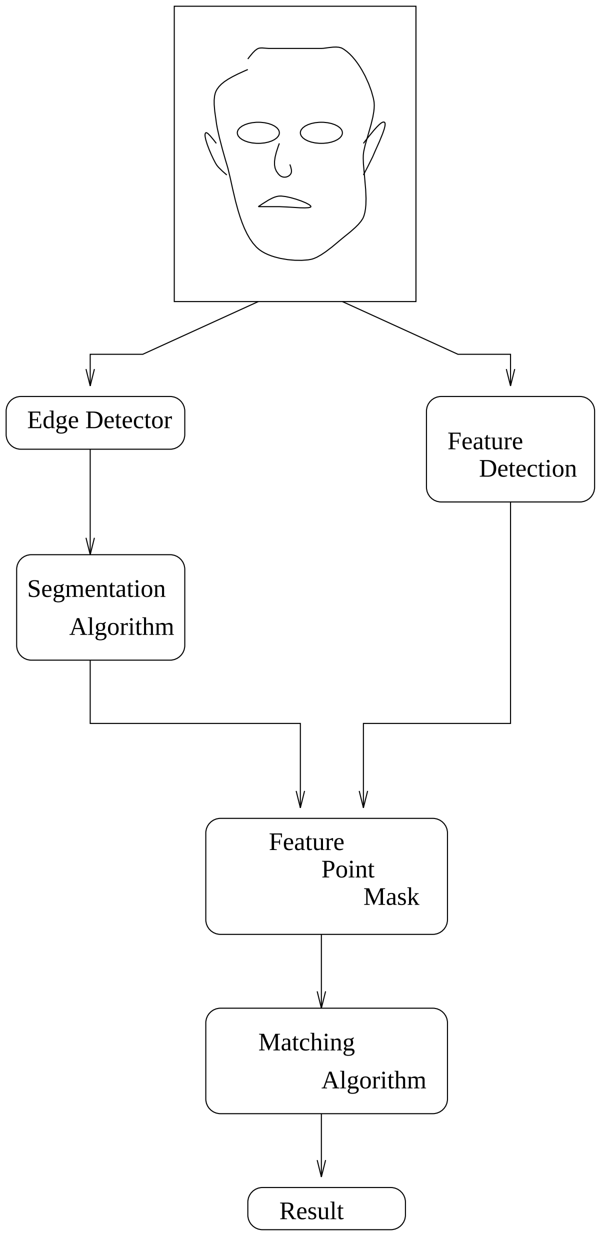


Figure 1.1: Outline of the Face Identification System

Chapter 2

Literature Review

Human face identification has proponents in well established though differing schools of thought. They can be broadly categorized into two major branches, one which relies on a 3-D input image, i.e. range data, while the other works with a 2-D representation (intensity data) of the image. Intuitively, a 3-D representation provides an added dimension to the useful information available from the data for the description of the face. Furthermore, the problem of segmentation of the face from the background becomes a trivial exercise. Two studies [7, 14] have demonstrated the advantages of having this additional information. On the other hand, proponents of a 2-D face identification scheme would cite the abundance of data present in the world today, from police mug shots to newspaper photographs, as a major incentive to do research on face identification based on 2-D data.

Kanade was one of the earliest researcher on automatic face recognition, as reported in his pioneering work [9]. He used projection analysis on a binary image obtained by applying a Laplacian operator to the grey scale image. Brunelli and Poggio [2] augmented the projection analysis by performing vertical and horizontal edge detection. They also discussed the classification of automatic face identification methods into two broad classes of techniques, geometric feature based matching and template matching. Geometric feature based matching involves decomposing the face image into pertinent features like eyes, nose, mouth, chin, etc. Several studies [3, 5, 6, 10, 11] involve, in one way or another, feature based identification. The features range from the locations of eyes, nose, mouth and chin and their spatial relationship to one another [3] to feature point detection utilizing Gabor wavelet decomposition [10]. For matching or identification purposes, the relevant information present in the spatial structure of these features, compensated for scale and orientation, is compared with a known database. Work on this technique started primarily with Kanade [9]; it has blossomed into a very promising technique for face identification.

Another class of techniques is the template matching approach. The general idea behind this approach is the construction of an artificial template to match with the prominent features of the face. Yuille et al. [15] described two such templates, one for the eye and the other for the mouth. Utilizing a condition of deformability, and allowing the template to translate and deform to fit its most likely representation in the image, they were able to locate the eyes and the mouth within the face. However, the starting point of the template was critical to their approach. Govindaraju, Srihari and Sher [8] attempted to locate the face in an image using a model template constructed of a hair curve and face side curves. They used a cost function approach to group together prospective left and right face side curves having the appropriate displacement and angular orientation. They do not mention testing their scheme on a large image set, and they state that they often encounter false alarms.

Another approach, slightly different from those mentioned above, is used by Turk and Pentland [13]. In their approach face images are projected onto a feature space called the "face space". The face space is defined by "eigen-faces", which are determined by the set of eigenvectors of the set of training faces. A new target face is projected onto this face space. The norm is used as the measure for matching. This method does not rely on features within the face for matching. To include a new face in the training set, the whole procedure has to be repeated. Another drawback of this approach is the total number of faces that can be recognized. The use of the norm as a criterion for matching does not seem robust enough to give results with high accuracy for databases with a few hundred to a few thousand faces.

In most of the approaches defined above it is imperative to be able to distinguish the face region in an image from the background clutter. Many of the recognition systems work very well in a controlled environment, but when moved over to an uncontrolled environment they begin to falter. We have attempted to produce a segmentation algorithm which will, in effect, translate face images from an uncontrolled environment to a controlled one.

Chapter 3

Segmentation of Faces from Images

The outline of the human head can be generally described as being roughly elliptic in nature. To segment the face region from the rest of the image, we have tried the approach of fitting the best possible ellipse to the outline of the head.

Several methods were considered to realize this approach. These methods were tested both on images with uniform backgrounds and on images having moderately cluttered backgrounds. One such method, utilizing the Hough transform [1, 12], was considered, at first, to be the best approach. This method is described below, where we will also mention our reason for not opting for this method.

3.1 Elliptic Hough Transform

The basis of the Hough transformation is the use of a parameter domain. In the case of the elliptic Hough transform this means, given a point (x, y) in the plane, find the parameters of an ellipse passing through that point (see Ballard [1]). The parameters are the center point (x_0, y_0), the semi-major axis a, and the semi-minor axis b of the ellipse. The method of finding them is outlined below.

1. Let the equation of the ellipse be

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1$$

Let $X = x - x_0$ and $Y = y - y_0$; then

$$\frac{X^2}{a^2} + \frac{Y^2}{b^2} = 1 \tag{3.1}$$

Differentiate with respect to X:

$$\frac{2X}{a^2} + \frac{2Y}{b^2}\frac{\mathrm{d}Y}{\mathrm{d}X} = 0 \tag{3.2}$$

2. Using (3.1) and (3.2) and the added information of the gradient $\frac{\mathrm{d}Y}{\mathrm{d}X}$ at each point (x, y), we can solve for x_0, y_0 over a range of values for a, b.

This technique utilizes the information present in the edge map of the image and tries to find the best ellipse given all the edge pixels and their gradient information. It can also handle rotation of the ellipse, i.e. the axes of the ellipse need not coincide with the axes of the image.

When this approach was tried on real images, even with uniform backgrounds, the computational complexity increased with the size of the parameter range for every edge pixel, a majority of them relating to features inside the face region. To limit the computation to a reasonable level, the edge map was constrained to contain only the outer shell of the face. In the case of a uniform background this was easily accomplished, but when moderately cluttered images were used, the computation time increased rapidly and it was decided that the Hough transform approach would not be viable for the task at hand.

3.2 Ellipse Fitting Technique

To avoid the problems encountered with the Hough transform method, a different approach was used which should be practicable even when used with images with moderately cluttered backgrounds. This method too derives its inspiration from the elliptical nature of the human face. It uses the edge information from the edge map and, through some preprocessing, labels contiguous edge segments. These labeled segments are then fitted to an ellipse, as pictorially represented in Figure 3.1. The results that were obtained show that this method is capable of eliminating straight line edge segments from the image. An implementation of this method is given in Figure 3.4.

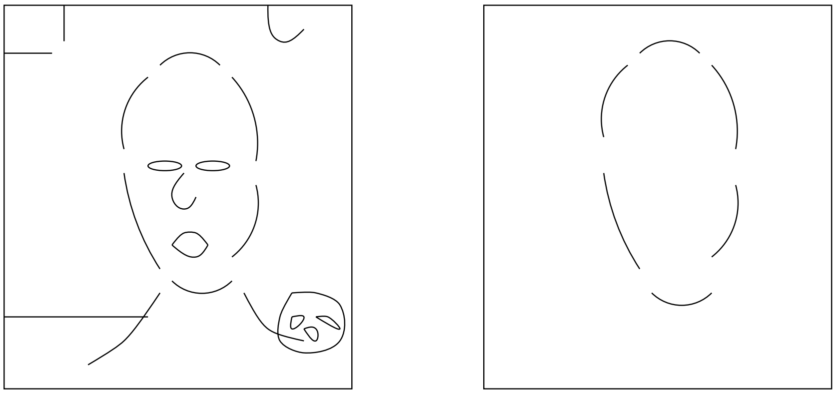


Figure 3.1: Rationale for using ellipse fitting approach.

Of the two methods described for the segmentation of the face from an image, it was decided to use the second one. Since the overall goal of this thesis is to be able to identify human faces on moderately cluttered backgrounds, a method was needed to use the results of the segmentation process in the identification process. In Figure 1.1, it should be noted that the two processes work independently of each other up to the point where the segmentation information is used to act as a mask to retain only the feature points that are within the region of the mask. This is a crucial point which needs to be addressed. If an image is segmented so that the only part of the original image that we keep is the elliptical region containing the face, and the remaining background is set to an arbitrary value, then the boundary of this segmentation will cause false feature points to be detected using [10]. Furthermore, since most of the fitted ellipses fall within a small range of parameter values, the matching scheme (Section 4.2) will yield false identification results caused by these elliptically configured feature points. Hence a mask is used which will circumvent the problem of having erroneous feature points in the image. The face identification scheme is described in the schematic diagram in Figure 1.1.

3.3 Segmentation of Faces from Uniform Background

The problem of segmentation of faces from a uniform background is not very complex. The edge map of the image can give a good outline of the image containing the face region. All edges in such an image represent the face region. A simple way to determine the outline of the face can be achieved by finding the outermost edge pixels in both the x and the y direction. If we let (x_i, y_i) represent the location of an edge pixel then

$$x_0 = \sum_{i=0}^{N} \frac{x_i}{N} \quad , \quad y_0 = \sum_{i=0}^{N} \frac{y_i}{N} \tag{3.3}$$

where N is the total number of edge pixels in the image and x_0, y_0 is the center of mass (centroid) of the edge image.

Since the idea is to try to fit the best possible ellipse to the edge image, the following equation needs to be solved:

$$\frac{(x_i - x_0)^2}{a^2} + \frac{(y_i - y_0)^2}{b^2} = 1 \tag{3.4}$$

where (x_0, y_0) is the center of the ellipse, a and b are the semi-major and the semi-minor axes respectively, and (x_i, y_i) represents the edge pixel location. The center of the ellipse can be approximated by (3.3) since it is assumed that the only object in the image is the face and (3.3) gives its center. The parameters a, b can be obtained by solving the overdetermined system of equations

$$\begin{pmatrix} (x_1 - x_0)^2 & (y_1 - y_0)^2 \\ \vdots & \vdots \\ (x_n - x_0)^2 & (y_n - y_0)^2 \end{pmatrix} \begin{pmatrix} \frac{1}{a^2} \\ \frac{1}{b^2} \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \tag{3.5}$$

The above equation is of the form $\mathbf{AX} = \mathbf{C}$ and can be solved by pseudo-inverting it:

$$\mathbf{X} = (\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t\mathbf{C} \tag{3.6}$$

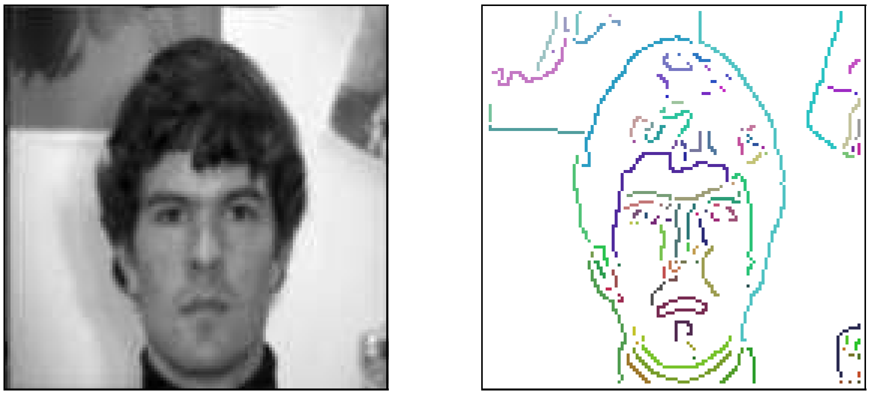
The results obtained by this method are accurate as long as the outer edge map of the face conforms to an elliptical structure. However, when this method was applied to segment a face with long hair (as in the case of female faces), the results were not always accurate. Another drawback to using this approach was its sensitivity to a cluttered background. To build a more robust segmentation algorithm one needs to be able to handle a varied input image. The next section considers a more general segmentation algorithm which takes into account the case of a moderately cluttered background; a uniform background can be regarded as a special case of a cluttered background.

3.4 Segmentation from Moderately Cluttered Background

The feature extraction algorithm (Chapter 4) [10] locates the feature points at maxima of local curvature in the intensity image. If the background is other than uniform then the intensity discontinuities in the background will give rise to feature points that are not part of the face region. Since the matching algorithm (Chapter 4) [10] requires all the feature points in the image to be matched in a graph matching manner, the feature points that lie in the background lower the accuracy of the face identification system. To compensate for these feature points, an algorithm was developed to segment the image into facial region and non-facial region, and then collect only those edges that lie within the facial region. The following procedure was developed for the image database obtained from MIT. It assumes that a good edge map of the image is available. This is obtained by using the Canny edge detector [4] from the KBVision software package. The remaining discussion will assume the availability of an edge map of the intensity image.

3.4.1 Removal of Intersection Points

Figure 3.2 shows an example of an intensity image and its edge map. The edge map contains numerous intersecting edge segments. If it is cross-referenced with its parent intensity image,



(a) (b)

Figure 3.2: Intensity image (a) and edge map from Canny's edge detector (b).

these intersections can be attributed to occlusions of objects. Let us call these intersection points "break-points". It is necessary to remove these points in such a manner as to preserve the integrity of the edge segment belonging to the object in the foreground (in this case the face). The reason for doing this will be made clearer in the section dealing with linking of similar edge segments.

It is required to disconnect the intersections of edge segments from the edge map in such a way that if two edge segments come from different objects they should be at least distance $\sqrt{2}$ apart. Let $\mathcal{E}$ represent the edge map and $\mathcal{E}(x, y)$ be the binary intensity value at pixel location (x, y) where $1 \leq x, y \leq n$. A square image is chosen for simplicity. Consider a 3×3 neighborhood centered about $\mathcal{E}(x, y)$. $\mathcal{E}(x, y)$ is selected as a break-point if, counting itself as one, there are more than four discontinuous edge pixels in its neighborhood. We define "discontinuous" as involving pixels which are more than $\sqrt{2}$ apart in distance, excluding $\mathcal{E}(x, y)$.

Table 3.1: (a), (b), (c) represent cases of break-points; (d), (e), (f) are non break-points.

×		×
	×	
		×

(a)

×		×
	×	
×		

(b)

×		
	×	
×		×

(c)

×	×	
	×	
		×

(d)

×		
	×	
×	×	

(e)

×	×	
	×	
	×	×

(f)

Some of the situations are represented in Table 1, where $\mathcal{E}(x, y)$,the center pixel, is designated as a break-point or as part of a continuous segment. Once it is decided that $\mathcal{E}(x, y)$ is a break-point it and its 3×3 neighborhood are removed from the edge map. A problem arises when a neighborhood clique is removed and a continuous edge segment, belonging to an object, is broken up into smaller components. The method of determining an ellipse relies on grouping edge segments of the same object (face) together to fit them to the ellipse. So it is necessary to link similar edge segments together to form a continuous segment. This process is described in the following sections.

3.4.2 Labeling of Contiguous Edge Segments

Each contiguous edge segment is assigned a label L_i. The assignment is done in a systematic manner, with segments getting labeled according to the rule that pixels are assigned the same label if they are at most distance $\sqrt{2}$ apart.

3.4.3 Joining Similar Segments

As was mentioned in the section on removal of break-points, some continuous segments, belonging to the same object, are broken up into smaller components. These smaller components need to be assigned the same label so that they can be recognized as belonging to the same object. The search for the segments is confined to the neighborhood around the break-points. The vector dot product is employed as the criterion to determine the similarity between edge segments. Let $\mathcal{E}(x, y)$ be a particular break-point and $L_i, i \in \{1, 2, 3\}$ be labeled segments found in the neighborhood of the break-point. This labeling is different from the global labeling and a particular label L_i may be present in another break-point neighborhood. We are primarily concerned with segments having different labels in a particular neighborhood and we also confine the linking process to at most three labeled segments in a neighborhood, which is the maximum number of segments that arise from the occlusion of two objects in all the cases that have been studied. If there were four labeled segments present it would not represent a situation of one object occluding the other in the image, such a case is therefore ignored as far as linking is concerned. Let us further define $\mathcal{E}_i(x, y), i \in \{1, 2, 3\}$ to be the edge pixels of the different segments present in the neighborhood. We find edge pixels which are n pixels away, along the edge segment, from the break-point. Define the vectors

$$\mathbf{v_i} = (\mathcal{E}(x, y) - \mathcal{E}_i(x', y')), \quad \forall i \in \{1, 2, 3\} \tag{3.7}$$

where (x', y') is the n^{th} edge pixel along the labeled segment. In our experiments $n = 5$. All vectors are normalized to unit vectors $\mathbf{e_i}, \forall i \in \{1, 2, 3\}$ by

$$\mathbf{e_i} = \frac{\mathbf{v_i}}{\|\mathbf{v_i}\|}, \quad \forall i \in \{1, 2, 3\} \tag{3.8}$$

When three separate edge segments are present in the neighborhood of the break-point, vector dot products are taken between all pairs of the three unit vectors $\mathbf{e_i}, \forall i \in \{1,2,3\}$:

$$\mathbf{e_i} \cdot \mathbf{e_j} = \cos\theta_{ij} \quad \forall i,j \ni i \neq j, \quad i,j \in \{1,2,3\}^2 \tag{3.9}$$

The two vectors with the minimum absolute value of the dot product are then declared to be part of the same object's edge and are both assigned (linked to) the same label. Thus the same labels are assigned to segments satisfying the following condition:

$$L_i = L_j, \tag{3.10}$$

$$\text{if } \cos\theta_{ij} = \min_{k,l \in \{1,2,3\}^2} |\mathbf{e_k} \cdot \mathbf{e_l}| \tag{3.11}$$

The absolute value is necessary to compensate for obtuse angles; since the directions of the vectors are always taken from the n^{th} edge pixel of the segment to the break-point, similar segments may be more than 90 degrees apart. Figure 3.3 depicts a possible situation where three segments are present in the neighborhood and labels 1 and 3 are linked.

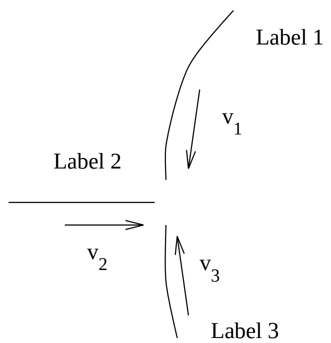


Figure 3.3: Linking of segments at break-point.

The case of two segments in the neighborhood corresponds to line ends, corners or curves in the edge map which do not have single edge pixel width. These two segments are tested for the possibility of arising from the same object's edge. Since a comparison between three vectors to determine the one with the smallest absolute angle is not possible, a threshold τ is set to determine whether or not the two segments should be linked. If the value of the dot product $\mathbf{v_i} \cdot \mathbf{v_j} \geq \tau$ then segments L_i and L_j will be merged together.

After all break-points have been checked for possible linking of segments the resulting "linked" edge map is in the correct form for grouping different edge segments to fit best possible ellipses to them.

3.4.4 Grouping to Form Likely Ellipses

The grouping of segments to form an ellipse is the goal of the segmentation algorithm. When all segments have been labeled and linked, the grouping algorithm takes pairs of segments and tries to fit an ellipse to them. First the equation of the ellipse is linearized; though the equation is non-linear in nature, under certain conditions it can be expected to behave in a linear manner. The equation of the ellipse

$$\frac{(x_i - x_0)^2}{a^2} + \frac{(y_i - y_0)^2}{b^2} = 1 \tag{3.12}$$

is multiplied by a^2 on both sides and rearranged in the following manner. The assumption that $a \neq 0$ is used; it is valid for all ellipses of consequence for the segmentation algorithm.

$$2x_i a_0 - y_i^2 a_1 + 2y_i a_2 - a_3 = x_i^2 \tag{3.13}$$

where

$$\begin{aligned} a_0 &= x_0 \qquad , \qquad a_1 = \frac{a^2}{b^2} \\ a_2 &= \frac{a^2}{b^2} y_0 \qquad , \qquad a_3 = x_0^2 + \frac{a^2}{b^2} - a^2 \end{aligned}$$

Here $i \in \{1, 2, 3, \ldots, N\}$ where N is the total number of edge pixels for a particular labeled segment. For a given labeled edge segment, an ellipse can be fit as follows:

$$\begin{pmatrix} 2x_1 & -y_1^2 & 2y_1 & -1 \\ 2x_2 & -y_2^2 & 2y_2 & -1 \\ \vdots & \vdots & \vdots & \vdots \\ 2x_N & -y_N^2 & 2y_N & -1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \vdots \\ x_N^2 \end{pmatrix} \tag{3.14}$$

Equation (3.14) is of the form $\mathbf{AX} = \mathbf{C}$ where $\mathbf{A}$ is $n \times 4$, $\mathbf{X}$ is 4×1, and $\mathbf{C}$ is $n \times 1$. It can be solved using the pseudo inverse method

$$\mathbf{X} = (\mathbf{A}^t \mathbf{A})^{-1} \mathbf{A}^t \mathbf{C} \tag{3.15}$$

where $\mathbf{A}^t$ is the transpose of $\mathbf{A}$. Here $\mathbf{A}^t\mathbf{A}$ and $\mathbf{A}^t\mathbf{C}$ are as follows:

$$\mathbf{A}^t\mathbf{A} = \begin{pmatrix} 2\sum_{i=1}^{N} x_i^2 & -2\sum_{i=1}^{N} x_i y_i^2 & 2\sum_{i=1}^{N} x_i y_i & -2\sum_{i=1}^{N} x_i \\ -2\sum_{i=1}^{N} x_i y_i^2 & \sum_{i=1}^{N} y_i^4 & -2\sum_{i=1}^{N} y_i^3 & \sum_{i=1}^{N} y_i^2 \\ 2\sum_{i=1}^{N} x_i y_i & -2\sum_{i=1}^{N} y_i^3 & 4\sum_{i=1}^{N} y_i^2 & -2\sum_{i=1}^{N} y_i \\ -2\sum_{i=1}^{N} x_i & \sum_{i=1}^{N} y_i^2 & -2\sum_{i=1}^{N} y_i & N \end{pmatrix} \tag{3.16}$$

$$\mathbf{A}^t\mathbf{C} = \begin{pmatrix} 2\sum_{i=1}^{N} x_i^3 \\ -\sum_{i=1}^{N} x_i^2 y_i^2 \\ 2\sum_{i=1}^{N} x_i^2 y_i \\ -\sum_{i=1}^{N} x_i^2 \end{pmatrix} \tag{3.17}$$

Each subscript i corresponds to a pixel location (x_i, y_i) of a labeled edge segment. Grouping initially requires two labeled segments L_i and L_j to be fitted to an ellipse. After finding $\mathbf{X}$ we can obtain x_0, y_0, a, b from equation (3.13). The aspect ratio a/b of most faces can be bounded within a range. This criterion along with the maximum and minimum bound on the semi-major axis a of the ellipse is used to select a class of possible candidate ellipses. Let the class be denoted by $\mathcal{C}_\mathcal{E}$. Each candidate ellipse in the class $\mathcal{C}_\mathcal{E}$ with its parameter values $\mathcal{P}_i = x_{0i}, y_{0i}, a_i, b_i$ is tested for accuracy by a simple error threshold given by the following:

$$e = \frac{(x_i - x_0)^2}{a^2} + \frac{(y_i - y_0)^2}{b^2} - 1, \quad \forall \quad i \in \{1, 2, 3, \ldots, N\} \tag{3.18}$$
$$\forall \quad L_j, j \in \{1, 2, 3, \ldots, M\}$$

Each (i^{th}) edge pixel of each (j^{th}) labeled segment is fitted to the ellipse with parameter set $\mathcal{P}_i$ belonging to the set $\mathcal{C}_\mathcal{E}$. If $e < \tau$ (where τ is a threshold) then the edge pixel (x_i, y_i) in labeled segment L_j is said to belong to the ellipse with parameter set $\mathcal{P}_i$. If more than half of the edge pixels of a particular segment belong to $\mathcal{P}_i$ then the labeled segment L_j is said to belong to the ellipse with parameter set $\mathcal{P}_i$. The parameter set $\mathcal{P}_i$ with the most segments is chosen as the best candidate for the face.

This procedure is further explained below in a stepwise manner.

1. Pairs of labeled segments L_i, L_j are fitted to the ellipse.

2. If the parameter set $\mathcal{P}_i$ for this pair satisfies the criteria of the aspect ratio (a/b) bound and the bound on the semi-major axis a then $\mathcal{P}_i \in \mathcal{C}_{\mathcal{E}}$.

3. If $e < \tau$ then $(x_i, y_i) \in L_j$ is said to belong to $\mathcal{P}_i$.

4. If more than half of $(x_i, y_i) \in L_j$ belong to $\mathcal{P}_i$ then $L_j \in \mathcal{P}_i$.

5. Choose $\mathcal{P}_i$ which has the most labeled segments belonging to it.

An example of this procedure is depicted in Figure 3.4. Figure 3.4 (a) shows an edge map of the image and Figure 3.4 (b) shows the resulting grouped segments after applying the algorithm. In this experiment the threshold τ is set to 0.07 and $1.3 \leq a/b \leq 2.0$ is the bound on the aspect ratio of the ellipse.

(a) (b)

Figure 3.4: Edge map of image (a) and the resultant segments left after grouping (b).

Chapter 4

Face Recognition System[1]

The face recognition system uses a biologically motivated algorithm developed by Manjunath and Chellappa [10]. It employs a Gabor wavelet decomposition and local scale interaction to extract features at points of curvature maxima in the image, corresponding to orientation and local neighborhood. These feature points are then stored in a database and subsequent target face images are matched using a graph matching technique. In [10] the algorithm was shown to work with an accuracy of up to 94%, using a database of about 300 images of 88 persons with uniform background. The algorithm is briefly outlined in the following sections.

4.1 Feature Point Extraction

A Gabor function is a Gaussian modulated by a complex sinusoid. A family of such functions can be obtained by varying the frequency of the complex sinusoid. The 2-D Gabor function and its Fourier transform are

$$g(x, y : u_0, v_0) = \exp(-[x^2/2\sigma_x^2 + y^2/2\sigma_y^2] + 2\pi i[u_0 x + v_o y]) \tag{4.1}$$

$$G(u, v) = \exp(-2\pi^2(\sigma_x^2(u - u_0)^2 + \sigma_y^2(v - v_0)^2)) \tag{4.2}$$

where σ_x and σ_y represent the spatial widths of the Gaussian and (u_0, v_0) is the frequency of the complex sinusoid. A salient feature of these functions is their ability to achieve minimum

[1]Paraphrased from [10].

joint resolution in the spatial and frequency domains.

The Gabor functions form a complete though non-orthogonal basis set. As in the case of Fourier series, a function $g(x, y)$ can easily be expanded using Gabor functions. Consider the following wavelet representation of the Gabor functions:

$$\Phi_\lambda(x, y, \theta) = \exp\left[(-\lambda^2(x'^2 + y'^2)) + i\pi x'\right] \tag{4.3}$$

$$x' = x\cos\theta + y\sin\theta \tag{4.4}$$

$$y' = -x\sin\theta + y\cos\theta \tag{4.5}$$

where θ is the preferred spatial orientation and λ is the aspect ratio of the Gaussian. For convenience the subscripts are dropped in the further discussion. In our experiments, λ was set to 1, and θ was discretized into four orientations. The resulting family of wavelets is given by

$$(\Phi(\alpha^j(x - x_0), \alpha^j(y - y_0), \theta_k)), \alpha \in \mathbf{R}, j = \{0, -1, -2, \ldots\} \tag{4.6}$$

where $\theta_k = k\pi/N$ $N = 4, k = \{0, 1, 2, 3\}$ and $\alpha^j, j \in \mathbf{Z}$. The corresponding wavelet transformation is obtained by convolving the function f with a bank of Gabor filters whose responses are simple dilation and translation of the basic wavelet expressed as

$$W_j(x, y, \theta) = f \otimes \Phi(\alpha^j x, \alpha^j y, \theta), \; j = \{0, -1, -2, \ldots\} \tag{4.7}$$

4.1.1 Feature Detection

Feature detection utilizes a simple mechanism to model the behavior of the end-inhibition. It uses interscale interaction to group the responses of cells from different frequency channels. This results in the generation of the end-stop regions. The orientation parameter θ determines the direction of the edge. The hypercomplex cells are sensitive to oriented lines and step edges of short lengths, and their response decreases if the lengths are increased.

$$I_{m,n}(x, y) = \max_\theta g\left(\| W_m(x, y, \theta) - \gamma W_n(x, y, \theta) \|\right) \tag{4.8}$$

where g is the sigmoid non-linearity, γ is a normalizing factor, and $n > m$. The final step is to actually localize these features; this is done by looking at the local maxima of the feature responses. A feature point is selected by taking the maximum in a neighborhood of the pixel (x, y). Let this neighborhood be N_{xy}:

$$I_{m,n}(x, y) = \max_{(x',y') \in N_{xy}} I_{m,n}(x', y') \tag{4.9}$$

The feature points thus detected are stored in a data file for the purpose of identification, which is described in the next section.

4.2 Graph Matching Used for Identification

The identification process utilizes the information present in a topological graph representation of the feature points. The feature points are represented by nodes V_i where $i = \{1, 2, 3, \ldots\}$, a consistent numbering technique. The information about the feature point is contained in $\{S, \mathbf{q}\}$, where S represents the spatial location and q is the feature vector defined by

$$\mathbf{q_i} = [Q_i(x, y, \theta_1), \ldots, Q_i(x, y, \theta_N)] \tag{4.10}$$

corresponding to the i^{th} feature point. N_i represents the set of neighbors which are of consequence for the feature point in question. The neighbors satisfying both the maximum number N and the Euclidean distance d_{ij} between the two points V_i and V_j are said to be of consequence for the i^{th} feature point.

To match an input graph to a stored graph which differs either in total number of feature points or in face location, we proceed in a stepwise manner. If i, j refer to nodes in the input graph $\mathcal{I}$ and x', y', m', n' refer to nodes in the stored graph $\mathcal{O}$, then the two graphs are matched as follows:

1. The centroids of the feature points of $\mathcal{I}$ and $\mathcal{O}$ are aligned.
2. Let N_i be the i^{th} feature point $\{V_i\}$ of $\mathcal{I}$. Search for the best feature point $\{V_{i'}\}$ in $\mathcal{O}$ using the criterion

$$S_{ii'} = 1 - \frac{\mathbf{q}_i \cdot \mathbf{q}_{i'}}{||\mathbf{q}_i||||\mathbf{q}_{i'}||} = \min_{m' \in N_i} S_{im'} \tag{4.11}$$

3. After matching, the total cost is computed taking into account the topologies of the graphs. Let nodes i and j of the input graph match nodes i' and j' of the stored graph and let $j \in N_i$ (i.e., V_j is a neighbor of V_i). Let $\rho_{ii'jj'} = \min\{d_{ij}/d_{i'j'}, d_{i'j'}/d_{ij}\}$. The topological cost is given by

$$T_{ii'jj'} = 1 - \rho_{ii'jj'} \tag{4.12}$$

4. The total cost is computed as

$$C_1(\mathcal{I}, \mathcal{O}) = \sum_i S_{ii'} + \lambda_t \sum_i \sum_{j \in N_i} T_{ii'jj'} \tag{4.13}$$

where λ_t is a scaling parameter assigning relative importances to the two cost functions.

5. The total cost is scaled appropriately to reflect the possible difference in the total numbers of feature points in the input and stored graphs. If $n_{\mathcal{I}}, n_{\mathcal{O}}$ are the numbers of feature points in the input and stored graphs respectively, then the scaling factor $s_f = \max\{n_{\mathcal{I}}/n_{\mathcal{O}}, n_{\mathcal{O}}/n_{\mathcal{I}}\}$ and the scaled cost is $C(\mathcal{I}, \mathcal{O}) = s_f C_1(\mathcal{I}, \mathcal{O})$.

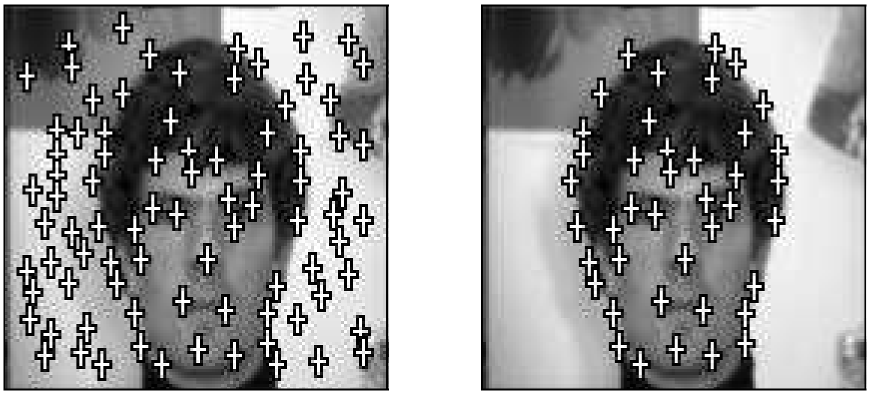
6. The best candidate is the one with the least cost, i.e. it satisfies

$$C(\mathcal{I}, \mathcal{O}^*) = \min_{\mathcal{O}'} C(\mathcal{I}, \mathcal{O}') \tag{4.14}$$

4.3 Comparison Between Segmented and Non-Segmented Face Images

As mentioned earlier, feature points selected on a moderately cluttered image include quite a few features outside the region of interest, the face. Employing the segmentation algorithm effectively confines them to the face region. This is illustrated in Figure 4.1. If the matching algorithm were employed on a non-segmented image the feature points in the background would also be considered. These points do not represent pertinent information and would give rise to misleading and incorrect matches.

The matching will involve the feature points that are in Figure 4.1 (b). The experimental results for both the segmentation algorithm and the recognition algorithm are presented in the next chapter.



(a) (b)

Figure 4.1: Feature points (a) without mask, (b) after mask.

Chapter 5

Experimental Results

5.1 Segmentation Results

When the segmentation algorithm was employed on a database of 48 images with cluttered backgrounds, 80% accuracy was achieved. Figures 5.1 (a) and (b) represent the intensity and edge map inputs to the algorithm, and Figure 5.1 (c) shows the linked edges that conform to the algorithm discussed in Chapter 3. The aspect ratio was in the range $1.3 \leq a/b \leq 2.0$ and $\tau = 0.07$ for these experiments. The final result is the mask which can be seen in Figure 5.1 (d), where the result of the segmentation algorithm is represented as an intensity image. Figure 5.2 shows good segmentations; Figure 5.3 shows inadequate segmentations. It is important to note that the segmentation algorithm performs its task quite quickly, but its accuracy decreases as the clutter increases, as anticipated.

5.2 Recognition Results Employing Segmentation

A database of 30 images was used for the recognition experiments. They were tested both with segmentation and without segmentation. When tested without the segmentation algorithm less than 50% accuracy was achieved. When the segmentation algorithm was used this improved to more than 70%. The parameters were set at $m = -2, n = -5$ and the number of orientations $N = 4$ in (4.8). For graph matching the number of points in the neighborhood was set to 5.

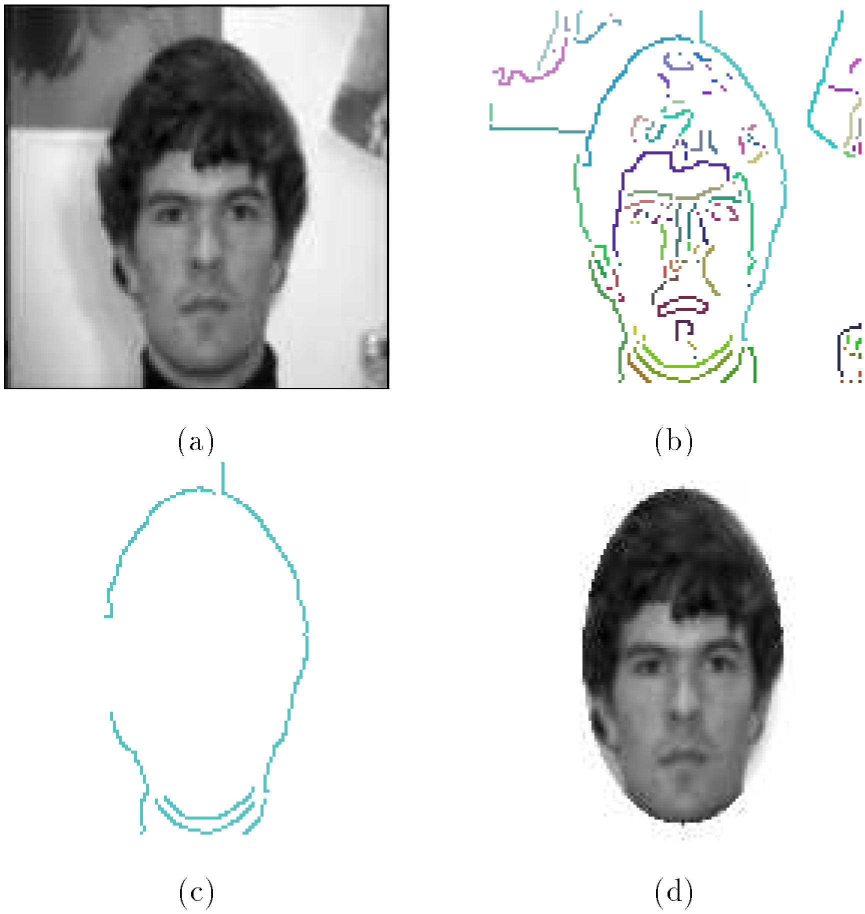
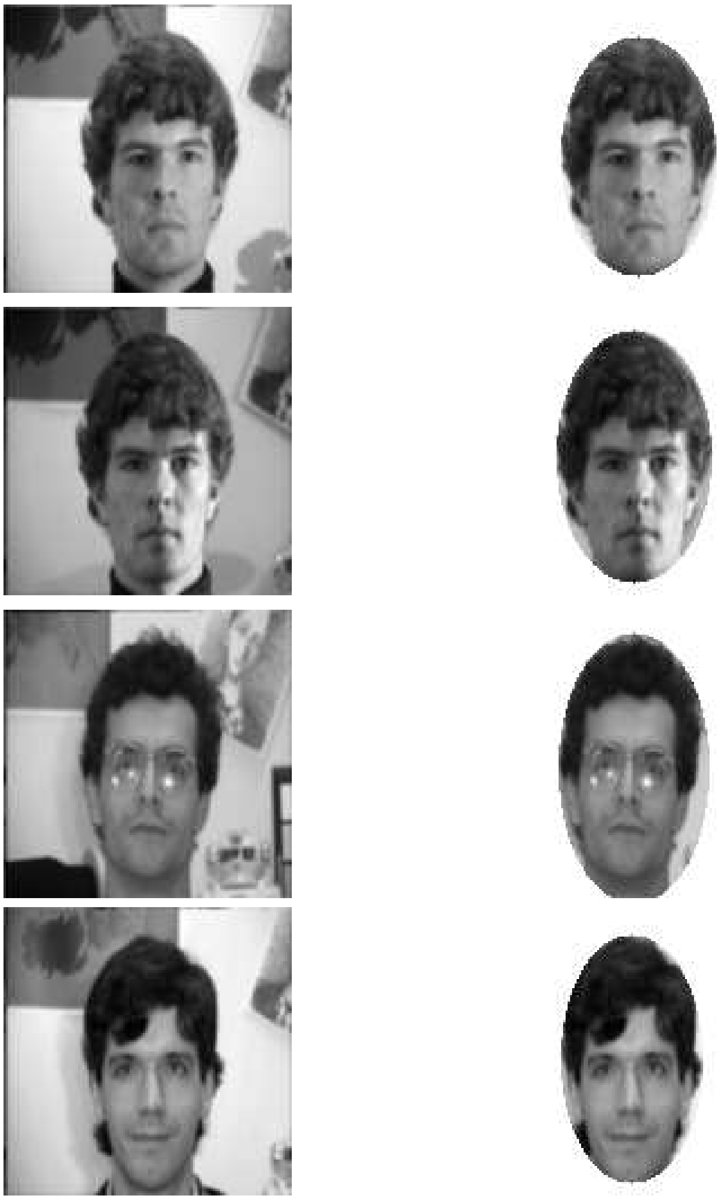


Figure 5.1: (a) Input image, (b) edge map, (c) linked segments, and (d) extracted image.

Although in [10] an accuracy of more than 94% is reported, the new database which was used for the experiments consisted of images taken with a change in the illumination direction. This causes variation in the detection of features between images of the same person because of the changes in intensity discontinuities due to the change in the illumination direction. As such, this database is not a fair test of the identification algorithm.

Figure 5.4 depicts the stages of the recognition algorithm employing the segmentation algorithm. Here (a) shows the input images, (b) shows the feature points that are detected without utilizing the segmentation algorithm as the mask, (c) depicts the mask in action, and (d) is the result of the segmentation algorithm. Figure 5.5 shows some accurate recognition results; (a) shows the input images, and (b) the results.



(a) (b)

Figure 5.2: Good results of segmentation. (a) Input image, (b) Extracted image.

(a) (b)

Figure 5.2: Contd.

(a) (b)

Figure 5.2: Contd.

(a) (b)

Figure 5.3: Inadequate segmentations. (a) Input image, (b) extracted image.

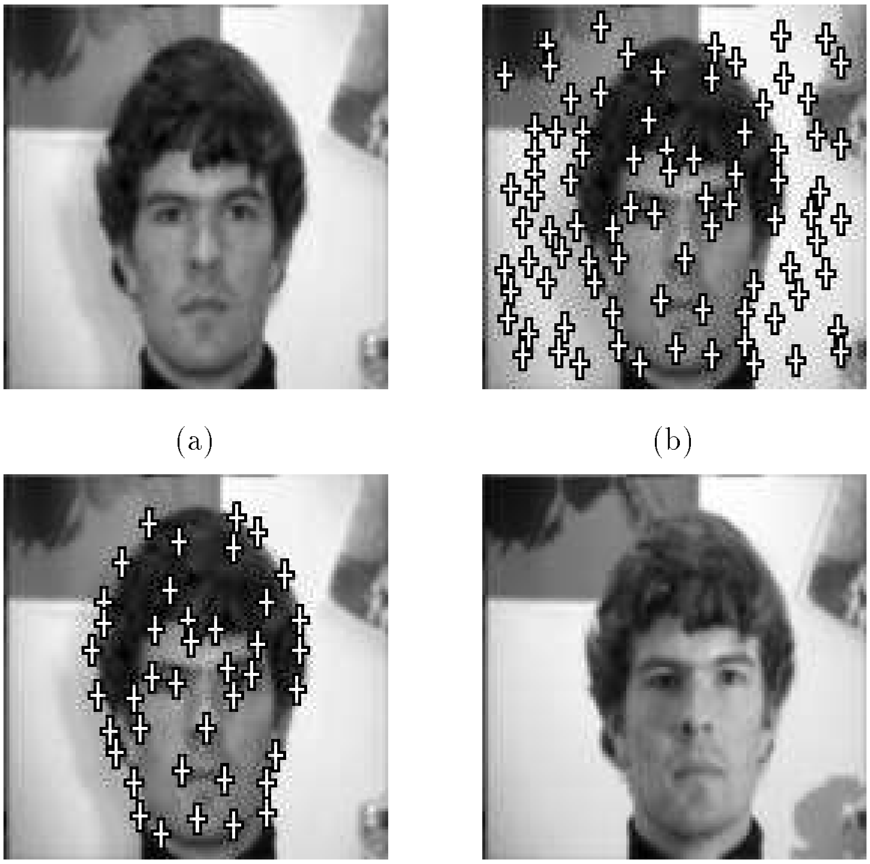


Figure 5.4: (a) Input image, (b) feature points without mask, (c) after mask, and (d) recognized image.

(a) (b)

Figure 5.5: Results of recognition algorithm. (a) Input image, (b) matched image.

Chapter 6

Conclusion

A relatively fast and simple algorithm has been developed for human face segmentation from images with moderately cluttered backgrounds. This algorithm utilizes the inherently elliptical nature of the human head and fits an ellipse to the head. The resultant information about the ellipse is then used to mask out unwanted feature points in the recognition phase of the identification system. When the algorithm was used in conjunction with the identification algorithm, a marked improvement in performance was achieved. It should be noted that the segmentation algorithm is a stand-alone algorithm and any other identification scheme can be used with it.

6.1 Future Work

Future work related to this thesis involves locating the positions of the eyes and the mouth to improve segmentation and provide added information to be used for identification purposes. We are pursuing a feature based approach to face identification in 2-D intensity images, and will extend its results to a 3-D system.

Bibliography

[1] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13:111–122, 1981.

[2] R. Brunelli and T. Poggio. Face recognition through geometrical features. *Proc. European Conf. Computer Vision*, pages 792–800, May 1992.

[3] R. Brunelli and T. Poggio. Hyperbf networks for gender classification. *Proc. DARPA Image Understanding Workshop*, pages 311–314, 1992.

[4] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:679–689, November 1986.

[5] I. Craw, D. Tock, and A. Bennett. Finding face features. *Proc. European Conf. Computer Vision*, pages 92–96, May 1992.

[6] S. Edelman, D. Reisfeld, and Y. Yeshurun. Learning to recognize faces from examples. *Proc. European Conf. Computer Vision*, pages 787–791, May 1992.

[7] G. G. Gordon. Application of morphology to feature extraction for face recognition. In *Proc. SPIE, Vol. 1658*, February 1992.

[8] V. Govindaraju, S. N. Srihari, and D. B. Sher. A computational model for face location. *Proc. Intl. Conf. Computer Vision*, pages 718–721, 1990.

[9] T. Kanade. *Picture Processing System by Computer Complex and Recognition of Human Faces*. PhD thesis, Dept. of Information Science, Kyoto University, Japan, November 1973.

[10] B. S. Manjunath, R. Chellappa, and C. von der Malsburg. A feature based approach to face recognition. *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pages 373–378, 1992.

[11] D. Reisfeld and Y. Yeshurun. Robust detection of facial features by generalized symmetry. *Proc. Intl. Conf. Pattern Recognition*, Vol. 1, pages 117–120, August 1992.

[12] S. D. Shapiro. Feature space transforms for curve detection. *Pattern Recognition*, 10:129–143, 1978.

[13] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pages 586–591, June 1991.

[14] Y. Yacoob and L. S. Davis. Qualitative labeling of human face components from range data. Technical Report CAR-TR-642, Center for Automation Research, University of Maryland, College Park, MD, 1992.

[15] A. L. Yuille, D. S. Cohen, and P. W. Hallinan. Feature extraction from faces using deformable templates. *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pages 104–109, June 1989.

[16] Q. Zheng and R. Chellappa. Automatic feature point extraction and tracking in image sequences for arbitrary camera motion. Technical Report CAR-TR-628, Center for Automation Research, University of Maryland, College Park, MD, 1992.