The Hong Kong Polytechnic University

Department of
Electronic and Information Engineering

Real-time Face Recognition with Live Detection

Wan Kwok-Wai

September 2004

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best to my knowledge and belief, it reproduces no material previously published or written nor material which has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

________________(Signed)

Wan Kwok Wai (Name of student)

Abstract

Human face recognition is one of the most useful techniques for identifying or authenticating a person. Although research on this topic has been conducted for more than twenty years, many problems still remain, and better techniques for facial feature detection and face recognition are needed. Therefore, the objective of this thesis is to devise and develop efficient methods for preprocessing facial images and recognizing human faces. In this thesis, different approaches for facial feature extraction and human face recognition are reviewed. Facial feature extraction is one of the preprocessing steps for automatic human face recognition. Its accuracy will directly affect the performance of the recognition system. In addition, the location of a face, the facial expression and the lighting conditions in an image may be unknown. The head orientation, face scale and the image quality of faces may be different between the query image and the stored image. The recognition procedure will become more difficult and computationally intensive in order to reduce the effect of the above mentioned problems. Therefore, human face recognition is a challenging research topic.

In this research, we propose a modified shape model which can adapt to face images under perspective variations. To make the model represent a face more flexibly, the representations of the important facial features, i.e. the eyes, nose and

mouth, and the face contour are separated. An energy function is defined that links up these two representations of a human face. In order to represent a face image under different poses, three models are employed to represent the important facial features: the left-viewed, right-viewed, and frontal-viewed models. Furthermore, the genetic algorithm (GA) is applied to search for the best representation of a face image.

One of the major difficulties in human face recognition systems is the pose variation problem. Most of the face recognition approaches assume that the pose of an input face is of upright and frontal view. In our work, we estimate the pose angle of the input face image by the shape model parameters, which are derived from a training data set. Then we use Gabor wavelets as local feature information extracted at the facial feature points for classification. The high-dimensional Gabor feature vectors are reduced by the Principal Component Analysis (PCA). The weighting similarity measure based on the pose angle is proposed in classification. The weighting function incorporates class discriminability of feature parameters to emphasize the significance of feature parameters to a particular pose. The face recognition approach proposed in this thesis can provide a reasonable performance level.

Acknowledgements

I would like to express my sincere gratitude to my Chief Supervisor, Dr. K. M. Lam. Without his support, this research work would not have been completed. He also offered me many invaluable ideas and suggestions for writing my thesis.

I would also like to thank all members of the DSP Research Laboratory, past and present. The countless discussions I had with them have proved to be both fruitful and inspiring.

I gratefully acknowledge the financial support of The Hong Kong Polytechnic University through the Innovation and Technology Fund of the Innovation and Technology Commission (Project No. UIT/034), which made possible this research by granting me a studentship over the course of my degree program.

Finally, I would also like to take this opportunity to thank the Centre for Multimedia Signal Processing of the Department of Electronic and Information Engineering.

Table of Contents

List of Figures

List of Tables

Author's Publications

The following technical papers have been published or submitted for publication based on the result generated from this work.

Journal papers:

1. Kwok-Wai Wan, Kin-Man Lam and Kit-Chong Ng, "An Accurate Active Shape Model for Facial Feature Extraction", Accepted to appear in Pattern Recognition Letters.

Conference papers:

2. Kwok-Wai Wan, Kin-Man Lam and Kit-Chong Ng, "An Accurate Active Shape Model Using Genetic Algorithm for Facial Feature Extraction", Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, pp. 109-112, 2004.

CHAPTER 1

Introduction

1.1 Background

Face recognition is an important capability of human beings in their everyday social interactions. People can easily recognize one another by looking at each other’s face. Recognizing a human face is such a fundamental task that even a child can do it. Hence, it is logical for us to imagine that a computer can be easily taught how to recognize an individual by looking at his/her face. However, many researchers have discovered that such a “trivial” task is not simple for a computer to perform. Computers are known to be capable of performing highly repetitive tasks, so much effort is still being made to “teach” them to recognize human faces.

Interest and research activities in the development of automatic human face recognition have been increasing significantly over the past 20 years, especially during the past few years. This is because security is now considered an increasingly important task in recent years. People are looking for more secure methods to protect their assets and valuable information without losing their identity. Three main types of security approach are commonly used: (1) password, (2) smart cards and (3) biometric personal identification. Generally, we need both a password and a smart card to get cash from an ATM, and a password to access a computer control or internet services. However, these methods are not really secure. For example, passwords can be guessed easily, as people often pick ones that are easy to remember, such as a nickname, their child's name or a favorite pet's name; while a smart card can be easily lost or stolen. Only biometric characteristics cannot be borrowed, stolen or forgotten. People cannot pass these characteristics onto anyone else. As mentioned above, the biometric approach is the most secure identification method of the three security approaches.

The common biometric personal identification methods include fingerprints, head geometry, palm geometry, retina, iris, speech, and face. Different technologies may be appropriate for different applications and environments. A comparison of some common biometrics is presented in Table 1.1 [1].

	Fingerprint	Retina	Iris	Speech	Face
Accuracy (error rate)	High	Very High	Very High	Medium	High
Ease of use	High	Low	Medium	High	Medium
Barrier to attack	High	Very High	Very High	No	Medium
Uses personally distinct characteristics	High	Very High	Very High	Medium	High
User friendliness	Medium	High	Medium	Medium	High

Table 1.1 Comparison of biometrics.

In the comparison of the biometric technologies, the retina and iris are the most secure identification methods. However, they are not user friendly. This is because people feel that these kinds of identifications are intrusive. In general, human face recognition is the most user friendly method and the most natural. Face recognition provides us with a convenient way to identify and recognize a person in a database. With face recognition, we can recognize a person by just taking a photo of that person. The user no longer needs to touch anything to scan his/her personal characteristics for personal identification, but just to stand in front of a camera. The system can then check its database to recognize the person from the image or video captured.

Apart from the convenience provided by face recognition, it can also be applied in a multimedia search engine. Fast-growing multimedia and Internet technologies now allow for searches for multimedia data such as video clips. However, information retrieval within a huge amount of multimedia data is still a challenging task. With face recognition and video segmentation technology, we can easily find a particular person in video clips by simply feeding a picture of that person into the search engine. Currently, the accuracy and reliability of existing face recognition are still limited; we therefore focus this thesis on investigating efficient algorithms for face recognition.

1.2 Investigated Approaches

The objectives of this research are to investigate and develop efficient techniques for human face recognition under perspective variations. The human face recognition has raised much attention in recent years due to its wide application in area such as access control, surveillance and multimedia search engine. Most face recognition approaches assume that the pose of an input face is an upright view; they seldom consider the face images under different poses. In this research work, we are primarily interested in both facial feature extraction and face recognition under perspective variations. Face detection or facial feature extraction should be performed before face recognition is carried out.

The system consists of two major parts: the first part is facial feature extraction based on a modified face shape model, and the second part is human face recognition based on the Gabor feature used as local feature information extracted at the facial feature points. In the first part, the location of important facial points such as the face contour, eyes, nose and mouth are extracted by a modified shape model which can adapt to face images under perspective variations. After that, the extracted facial features are projected into a face shape model to estimate the corresponding pose angle. Finally, the Gabor features at each location of the facial feature points are extracted. In addition, an appropriate weighting function based on estimated the pose angle is employed in the similarity measure to provide a better classification performance level.

1.3 Introduction to Human Face Recognition

The configuration of an automatic human face recognition system is illustrated in Figure 1.1, which is divided into two main steps: preprocessing and face recognition. The preprocessing step includes the procedures to detect the existence and position of the human face, and to extract the important facial features such as eyes, nose and mouth. Face detection is a step which determines whether a human face appears in a source image or not. The human face may appear in any position in an image. The face size and orientation are unknown. In addition, unknown background, lighting conditions and facial appearance also affect the accuracy and the effectiveness of the detection result. Therefore, the task of face detection is a difficult and computationally intensive process. In order to perform face detection efficiently and accurately, algorithms based on the template-matching approach, the knowledge-based approach, the feature-based approach, the shape information approach, etc. have been proposed.

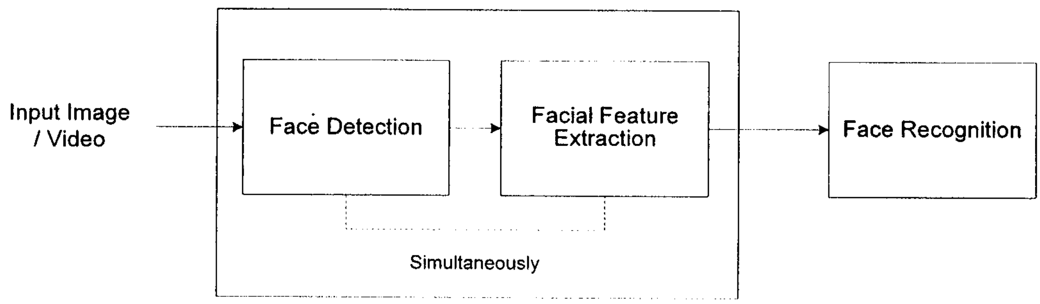


Figure 1.1 Configuration of an automatic human face recognition system.

Facial feature extraction is a part of the preprocessing step in automatic human face recognition. It is also a key part of the animation and recognition of facial expressions. Face detection and facial feature extraction are always achieved simultaneously, as indicated in Figure 1.1. For example, face detection may employ the characteristics of facial features, such as the eyes, so these features are also extracted simultaneously with face detection. Otherwise, a human face is detected first, and then the respective facial features can be extracted based on the geometric structure of a human face.

Techniques for facial feature detection and extraction can be divided into three approaches: the feature-based approach, the template-based approach, and the structural matching-based approach. The feature-based method identifies facial features based on their geometric properties. However, this method is computationally intensive and inflexible to scene and noise variations. The template-based approach models the boundary shape of natural objects such as the eyes and mouth. However, the template-based approach is associated with problems such as slow convergence and a lengthy processing time. To detect the features more reliably, recent approaches have used the structural matching-based approach. This approach is based on statistical methods. Hence, it may not be able to provide a good fit if the shapes are quite different from the training data set. Consequently, efficient and reliable methods for face and facial feature detection are still under investigation.

In an automatic human face recognition system, a face region is extracted and then normalized based on the position of the two eyes. The normalized human face is aligned with those human faces in a database, and they are then compared. In other words, the accuracy of face detection and facial feature extraction will significantly affect the performance of an automatic human face recognition system.

Many techniques for human face recognition have been proposed, and can be divided into three categories: holistic-based approaches, feature-based approaches and hybrid approaches. In the holistic-based approach, recognition takes into account the global properties of a pattern, such as the whole face region, as the raw input to the recognition system. One of the most widely used representations of the face region is the eigenface [2], which is based on principal component analysis (PCA). The eigenface method represents a human face by a linear combination of weighted eigenvectors. However, to achieve a reasonable performance, the images under consideration must also be aligned to each other. In the feature-based approaches, local features such as the eyes, nose and mouth are, typically first extracted, and their locations and local statistics (geometric and/or appearance) are fed into a classifier. Hybrid approaches such as the human perception system use both local features and the whole face region to recognize a face. A machine recognition system should use as much relevant information as possible in the training and recognition process. One can therefore argue that the methods based on the hybrid approach should potentially offer the best performance.

1.4 Organization of this Thesis

The rest of this thesis will introduce the existing techniques for facial feature extraction and human face recognition, as well as the respective techniques devised and developed in this thesis. Chapter 2 is a review of the state-of-the-art technologies in facial feature extraction and human face recognition. Chapter 3 outlines our efficient approach for facial feature extraction under perspective variations. The technique used in our approach includes three statistical face shape models under three different poses. A modified face shape model is also proposed. In order to represent a face more flexibly, the representations of the important facial features, i.e. the eyes, nose and mouth, and the face contour are separated. The models are then fit to a human face by means of the genetic algorithm. In Chapter 4, an effective method for pose estimation will be described. Two shape model parameters in the statistical face shape model are used in our approach. They can provide more detailed information about the pose of a human face, and so can be used to make the estimate more accurately and reliably. In Chapter 5, our proposed efficient approach for human face recognition based on the Gabor features and weighting similarity measurement will be presented. The experimental results show that our proposed method outperforms normal similarity measures. Finally, a summary of the major developments and a conclusion of this research work are provided in Chapter 6.

CHAPTER 2

Literature Reviews

2.1 Introduction

The first step in any automatic human face recognition system is the detection of faces in images or video sequences. After a face has been detected, the task of feature extraction is to obtain features for face classification. Depending on the type of classification system, the features can be a holistic face or local features such as lines or facial features. Face detection may also employ facial features in the detection process, during which facial features and the face are extracted simultaneously. Facial feature extraction is also a key component of the animation and recognition of facial expressions.

The performance of an automatic human face recognition algorithm is quite dependent on the accuracy of face detection and facial feature extraction. In other words, without accurate face and facial feature location, noticeable degradation in recognition performance is observed. The close relationship between facial feature extraction and face recognition motivates us to review a few extraction methods and face recognition methods that are used in the human face recognition approaches.

2.2 Facial Feature Extraction

The importance of facial features for face recognition cannot be overstated. Many face recognition systems need facial features in addition to the whole face. It is well known that even the holistic-based approaches, e.g. eigenfaces [2] and Fisherfaces [3], need accurate locations of some key facial features such as the eyes, nose, and mouth to normalize the detected faces [4, 5].

Facial feature extraction is one of the most challenging research topics, even though it might not be difficult for people to perceive human faces and facial features in an image. A lot of research on facial feature extraction has been presented in [6-21]. The techniques in [6-9] define a head model for extracting the facial features after the human faces have been detected. Various approaches to extracting the position of the facial features have been proposed. However, they can only roughly estimate the position. One of the early approaches is to compute the horizontal and vertical projections of an [6, 9, 10] image to determine the position of the eyes and the mouth. As the eyes, nose and mouth regions appear darker than other regions in a face, the summation of the grey-level intensities for these regions will exhibit as a local minimum.

In the following sections of this chapter, we will describe three popular algorithms namely active contour model (snakes) [22-25], deformable template

[14-20] and active shape model (ASM) [26-33] for facial feature extraction.

2.2.1 Active Contour Model

Active contour model (Snake) was first proposed by Kass *et al.* [22] in 1987 as an application for representing image contour. The snake can stick to edges accurately. The model is an energy-minimizing spline that can be operated under the influence of internal contour forces, image forces and external force. It is represented as a parametric curve $\boldsymbol{v}(s) = [x(s), y(s)]$, where the arc length s is a parameter. The energy functional [24] of a snake is given as follows:

$$\begin{aligned} E^*_{snake} &= \int_0^1 E_{\text{snake}}[v(s)]ds \\ &= \int_0^1 E_{\text{internal}}[v(s)] + E_{\text{image}}[v(s)] + E_{\text{constraint}}[v(s)]ds \end{aligned} \tag{2.1}$$

The solution to the energy function can be found using different calculus, but this method has the problem of numerical stability. The difficulty in the initialization and slow processing speed are the drawbacks of this approach. Hence, a fast iteration process approach has been proposed, the greedy algorithm [24], which allows a contour with controlled first- and second-order continuity to converge in an area with high image energy. Another fast approach based on the greedy algorithm was presented in [23, 24]. In this approach, two alternate search patterns are used, and a reduction in execution time of about 30% can be achieved with the same performance as the greedy algorithm.

2.2.2 Deformable Template

The deformable template, proposed by Yuille *et al.* [14] in 1991, is a common method to extract facial features such as the eyes and mouth in images. It uses parametric models to describe the physical shape of the eyes and mouth. This technique allows the *a priori* knowledge about the expected shape of features due to the fact that the general shapes of the human eye and mouth are more or less fixed. This knowledge of the shape guides the detection process. An eye template and a mouth template [14] are shown in Fig. 2.1 and 2.2, respectively.

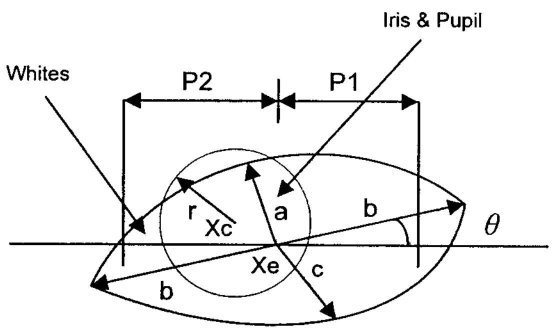


Figure 2.1 A deformable template for a human eye.

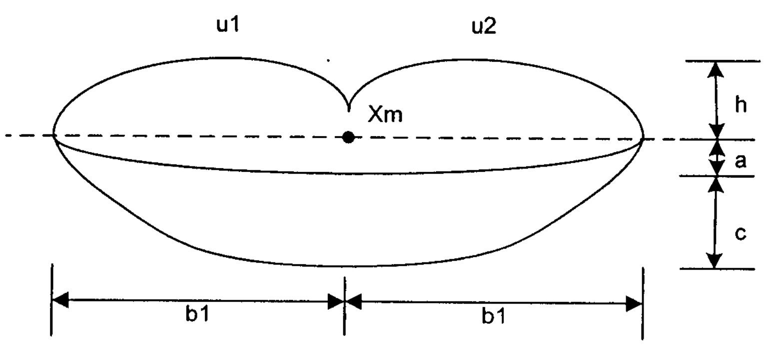


Figure 2.2 A deformable template for a mouth.

The eye template has totally eleven parameters represented by *(*$\vec{x}_c$, $\vec{x}_e$, p_1, p_2, r, a, b, c, θ). This template is modeled by two parabolic curves representing the upper and lower parts of the boundary. It has a center $\vec{x}_e$, with a width of 2b, maximum heights of a and c for the upper and lower boundaries, and an angle of orientation θ. r is the radius of the iris represented by a circle centered at $\vec{x}_c$.

For the mouth template, its center is at point $\vec{x}_m$ and its orientation is θ. The widths of the left and right parts of the mouth template are b_1 and b_2 from $\vec{x}_m$, respectively. The lower two parabolas have maximum distances of a and $a+c$ from the central line. The intersection of the upper two parabolas, u_1 and u_2 occurs at a height of h above $\vec{x}_m$. These templates act on three representations of an image, which are the peak, valley, and edge, as well as on the image itself. An energy function is devised based on these four representations and used to guide the deformation of these templates. The final size, shape and orientation of the eye and mouth templates are obtained by determining the local minimum of the respective energy functions. It is a time consuming procedure to determine all the parameters through the optimization process. Furthermore, in order to extract the eye, the template must be started at or below the eye. If it is started above the eye, the valley force from the eyebrows may cause problems. Hence, [15, 16, 20] and [17, 18] introduced a reliable method to locate the eye and the mouth, respectively. In [15, 16], the corners of an eye are first located by means of a corner detection scheme. Based on the corner position, the shape of the eye can be estimated accurately. The exact shape of the eye is then extracted by a new scheme which is similar to snake [22]. In [17, 18], a mouth boundary curve is initially formed by three control points. The exact locations of these control points are then determined through an optimization process by using a set of cost functions.

2.2.3 Active Shape Model

Active Shape Model (ASM), proposed by Cootes and Taylor [26] in 1992, is a commonly used technique for facial feature extraction. This technique is similar to the snakes, but has the advantage that instances of an ASM can only deform in the ways found in its training set. ASM also allows considerable variability in shape modeling, but the model is specific to the class of target objects or structures that it intends to represent.

The Shape Model

A shape model is described by n landmark points that represent the important positions in the object to be represented. These points are generated based on a set of training shapes. Each training shape $\boldsymbol{x}$ is represented as a shape vector, which is a collection of landmark points called a point distribution model [26],

$$\boldsymbol{x} = (x_0, y_0, x_1, y_1, \ldots, x_k, y_k, \ldots, x_{n-1}, y_{n-1})^T, \tag{2.2}$$

where T represents the transpose operation, and (x_k, y_k) are the coordinates of the k^{th} landmark point. Figure 2.3 shows a training image with its landmark points marked.

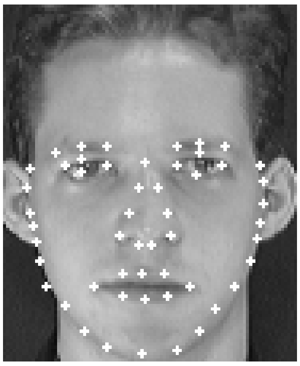


Figure 2.3 Locations of the points used to represent a face.

The training shapes are all aligned by translation, rotation and scaling for minimizing the sum of squared distances between their corresponding landmark points. Then, the mean shape $\bar{\boldsymbol{x}}$ and the deviation of each training shape from the mean are calculated. Principal component analysis (PCA) is then applied to capture most of the shape variations. Therefore, a shape model can be approximated as follows:

$$\boldsymbol{x} \approx \bar{\boldsymbol{x}} + \boldsymbol{P}\boldsymbol{b}, \tag{2.3}$$

where $\boldsymbol{P} = (\boldsymbol{p}_1\ \boldsymbol{p}_2\ \ldots\ \boldsymbol{p}_t)$ is the matrix whose columns are the first t eigenvectors with the largest eigenvalues arranged in descending order, and $\boldsymbol{b} = (b_1 \quad b_2 \quad \ldots \quad b_t)^T$ is a weight vector for the t eigenvectors, referred to as the shape parameters. When fitting the shape model to an object, the value of b_i is constrained to lie within the range ± 3 standard deviations. This can ensure that this range of the shape parameters can represent most of the shape variations in the training set. The number of eigenvectors t to be used is determined such that the eigenvectors can represent a certain amount of the shape variations in the training shapes, usually ranging from 90% to 95%. The desired number of eigenvectors t is given by the smallest t which satisfies

$$\sum_{i=1}^{t} \lambda_i \geq 0.95 \sum_{i=1}^{N} \lambda_i \ , \tag{2.4}$$

where N is the total number of eigenvectors available.

Modeling the Gray-Level Appearance

The gray-level appearance model [29], which describes the local texture feature around each landmark, is the normalized derivative of the profiles sampled perpendicular to the landmark contour and centered at the landmark. This gray-level information is used to estimate the best position of the landmarks in the searching process. The normalized derivative of the profiles is invariant to the offsets of the gray levels.

The gray-level profile, $\boldsymbol{g}_{ij}$, of the landmark j in the image i is a $(2n+1)$-D vector, in which n pixels are sampled on either side of the landmark under consideration,

$$\mathbf{g}_{ij} = \lfloor g_{ij0} \quad g_{ij1} \quad \cdots \quad g_{ij(2n+1)} \rfloor, \tag{2.5}$$

where g_{ijk}, $k = 0, \ldots, 2n+1$, is the gray-level intensity of a corresponding pixel. The derivative profile of $\boldsymbol{g}_{ij}$ has a length of $2n$, and is given as follows:

$$d\mathbf{g}_{ij} = \lfloor g_{ij1} - g_{ij0} \quad g_{ij2} - g_{ij1} \quad \cdots \quad g_{ij(2n+1)} - g_{ij(2n)} \rfloor \tag{2.6}$$

The normalized derivative profile is given by

$$\mathbf{y}_{ij} = {d\mathbf{g}_{ij}} \Big/ {\sum_{k=0}^{2n} \left| dg_{ijk} \right|}, \tag{2.7}$$

where $dg_{ijk} = g_{ij(k+1)} - g_{ijk}$. The covariance matrix of the normalized derivative profile for N training images is

$$\mathbf{C}_{yj} = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{y}_{ij} - \overline{\mathbf{y}}_j\right)\left(\mathbf{y}_{ij} - \overline{\mathbf{y}}_j\right), \tag{2.8}$$

where $\overline{\mathbf{y}}_j$ is the mean profile. The ASM employs the information obtained from modeling the gray-level statistics around each landmark to determine the desired movement or adjustment of each landmark such that a face shape model can fit into the target object accurately. To determine the movement of a landmark, a search profile (Figure 2.4) – which is a line passing through the landmark under consideration and perpendicular to the contour formed by the landmark and its neighbors – is extracted. A number of sub-profiles will be generated when the best set of shape parameters is being searched. These sub-profiles are matched to the corresponding profiles obtained from the training set. The difference between a sub-profile **y** and the training profile is computed using the Mahalanobis distance as follows:

$$f(\mathbf{y}) = (\mathbf{y} - \bar{\mathbf{y}}_j)^T C_{yj} (\mathbf{y} - \bar{\mathbf{y}}_j) \tag{2.9}$$

Minimizing $f(y)$ is equivalent to maximizing the probability of y matched to $\bar{\mathbf{y}}$ according to a Gaussian distribution.

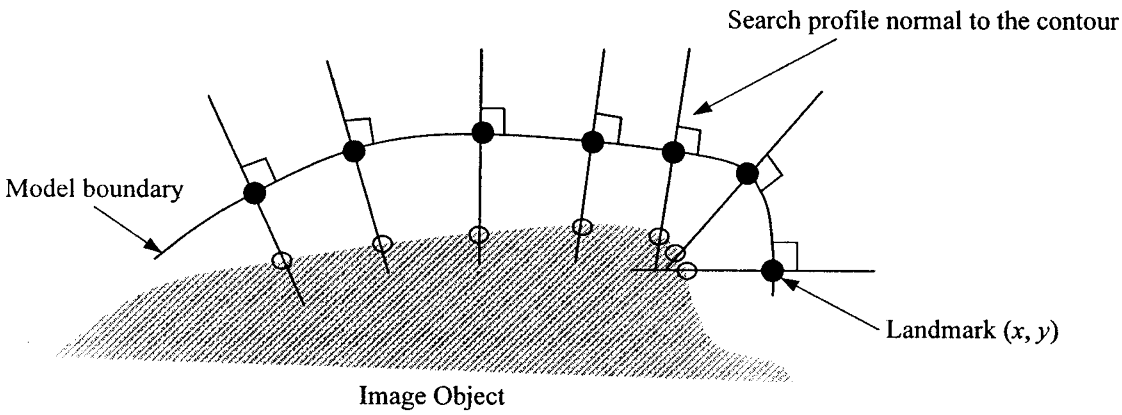


Figure 2.4 A search profile normal to the model boundary.

Optimization Algorithm

Modifying the point distribution model (PDM) to fit into the object in an image is an iterative optimization process. Starting from the mean shape representation of the model, each point of the model is allowed to move dynamically until it fits the object. At each model point or landmark, a profile perpendicular to the contour is extracted and a new and better position of that point is estimated along this profile.

Different approaches [30-32] can be used to search for a better position for the points. The simplest way is to find the strongest edge along the searching profile. Another approach is to create the gray-level appearance model or profile of each point, which will maximize the probability of the gray-level profile, as described in the last section. After searching, the shape parameters $\boldsymbol{b} = (b_1 \quad b_2 \quad \ldots \quad b_t)^T$ and the pose parameters (i.e. rotation, scale, and translation of the model) are adjusted in such a way as to minimize the overall distance between the new position of the points and the position of the original points. The adjusting process is repeated until no significant change in the model points is observed.

2.3 Human Face Recognition

Many methods of face recognition have been proposed during the past 20 years, and many literature reviews of face recognition have been conducted [35, 36]. Face recognition is such a challenging and interesting problem, which is why the literature on face recognition is so vast and diverse. A single face recognition system involves mixture techniques. The use of these mixture techniques makes it difficult to classify the type of techniques in a system used. In order to categorize the techniques more clearly, we use the following categorization:

(1) Holistic-based approaches: These methods use the whole face region as the raw input to a recognition system. One of the most widely used representations of the face region is the eigenface [3, 4], which is based on Principal Component Analysis.

(2) Feature-based approaches: Typically, in these methods, local features such as the eyes, nose and mouth are first extracted and their locations and local statistics (geometric and/or appearance) are fed into a structural classifier.

(3) Hybrid approaches: Just as the human perception system uses both local features and the whole face region to recognize a face, a successful face recognition system should use both. These methods should potentially offers a better performance level than the other two.

In this section, some examples of these three approaches will be presented.

2.3.1 Holistic-based Approaches

Eigenface

The eigenface [2] was proposed by Alex. P. Pentland and Matthew A. Turk of MIT in 1991. The main idea of the eigenface is to obtain the features in a mathematical sense instead of the physical face feature by using a mathematical transform for recognition.

There are two phases for face recognition using eigenfaces. The first phase is the training phase. In this phase, a large group of individual faces acts as the training set. These training images should be a good representation of all the faces that one might encounter. Their size, orientation and light intensity should be standardized. For example, all images are of size 128 × 128 pixels and all are frontal faces. Each of the images in the training set is represented by a vector of size $N \times N$, with N representing the size of the image. With the training images, a set of eigenvectors is found by using Principal Component Analysis (PCA).

The basic idea of PCA is to take advantage of the redundancy existing in the training set for representing the set in a more compact way. Using PCA, we can represent an image using M eigenvectors, where M is the number of eigenvectors used, ($M << N^2$). As M is much smaller than N^2, the comparing computation required for two feature vectors is greatly reduced.

PCA is done by first finding the average face ψ by averaging the training set images $\{T_1, T_2, \ldots\ldots T_M\}$ with T_i representing each of the vectors in the set. Then we form a matrix A = $\{\varphi_1, \varphi_2, \cdots\cdots\varphi_M\}$ with column vector $\varphi_i = T_i - \psi$, which is the difference vector of the training images and the average face. We can then compute the covariance matrix $C = AA^T$, as well as the eigenvectors and the associated eigenvalues of C.

After the eigenvectors have been calculated, the eigenvectors are sorted according to the magnitudes of their respective eigenvalues. These vectors are known as eigenfaces. M' ($M'<M$) eigenfaces with the largest eigenvalues are chosen, which are considered the best eigenvectors to represent a face. The span of the M' eigenfaces are called a face space.

The second phase of this algorithm is to recognize a face image. In this phase, a new or query face image is available. To recognize this image, we first subtract it by the average face ψ. Then, we calculate the dot products of the input vector and the eigenfaces. This makes a projection of the input image onto the face space. Similarly, we make projections of the training images onto the face space. The projection of an image onto the face space appears as a point in the plane. The Euclidean distances between the projection of the input face and that of each face in the database are then computed. The image in the database which has the minimum distance to the input should be the best match.

Fisherface

Fisherface [3] was proposed by Peter N. Belhumeur, Joao P. Hespanha and David J. Kriegman of Yale Univeristy in 1997. This approach is similar to the eigenface approach, making use of the projection of images into a face space, with improved insensitivity to large variation in lighting and facial expression.

The eigenface method uses PCA for dimensionality reduction, which yields projection directions that maximize the total scatter across all classes of images. The PCA is the best method for representing images from a low dimensional basis. However, this method does not consider the between-class scatter of the different subjects in a database. The projection may not be optimal in terms of discrimination in different classes. Let the total scatter matrix S_T be defined as follows:

$$S_T = \sum_{k=1}^{N} (T_k - \psi)(T_k - \psi)^T . \tag{2.10}$$

The projection W_{opt} is chosen to maximize the determinant of the total scatter matrix of the projection samples, i.e.

$$\begin{aligned} W_{opt} &= \arg\max_W \left| W^T S_T W \right| \\ &= [w_1, w_2, \cdots\cdots, w_m] \end{aligned} \tag{2.11}$$

where $\{w_i|\ i=1,2 \cdots \cdots ,m\}$ is the set of n–dimensional eigenvectors of S_T corresponding to the m largest eigenvalues.

The Fisherface method uses Fisher's Linear Discriminant (FLD), formulated by R.A. Fisher. This projection maximizes the ratio of between-class scatter to that of within-class scatter. The idea is that it tries to "shape" the scatter in order to make it more reliable for classification. Let the between-class scatter matrix be defined as

$$S_B = \sum_{l=1}^{C} N_l \left(\psi_l - \psi\right)\left(\psi_l - \psi\right)^T \tag{2.12}$$

where C is the total number of classes and N_l is the number of samples in l class.

and the within-class scatter matrix be defined as follows

$$S_W = \sum_{l=1}^{C} \sum_{T_k \in T_l} \left(T_k - \psi_l\right)\left(T_k - \psi_l\right)^T \tag{2.13}$$

where ψ_l is the mean image of class T_l. The optimal projection W_{opt} is chosen as the matrix with orthonormal columns, which maximizes the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples, i.e.

$$\begin{aligned} W_{opt} &= \arg\max_W \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|} \\ &= [w_1, w_2, \cdots\cdots, w_m] \end{aligned} \tag{2.14}$$

where $\{w_i|\ i=1,2 \cdots \cdots ,m\}$ is the set of n–dimensional eigenvectors of S_B/S_W corresponding to the m largest eigenvalues.

2.3.2 Feature-based Approaches

Many methods in the feature-based approaches have been proposed, including many early methods based on the geometry of local features [37, 38] as well as 1D [39] and pseudo-2D [40] HMM methods. One of the most successful of these approaches is Elastic Bunch Graph Matching (EBGM) [41], which is based on Dynamic Link Architecture (DLA) [42, 43]. This approach uses an elastic bunch graph to automatically locate the facial feature points on a face (eyes, nose, mouth, etc.) and to recognize the face according to these facial features.

The representation of facial features is based on the Gabor wavelet transform. Gabor wavelets are biologically motivated convolution kernels in the shape of plane waves restricted by a Gaussian envelope function. The family of Gabor kernels is given as follows:

$$\varphi_j\left(\vec{x}\right)=\frac{k_j^{\,2}}{\sigma^2}\exp\left(-\frac{k_j^{\,2}x^2}{2\sigma^2}\right)\left[\exp(i\vec{k}_j\vec{x})-\exp\left(\frac{-\sigma^2}{2}\right)\right] \tag{2.15}$$

This function is in the shape of plane waves with wave vector $\vec{k}_j$, restricted by a Guassian envelope. EBGM employs a discrete set of 5 different frequencies, index $v = 0, 1,\ldots,4$, and 8 orientations, with index $\mu = 0, 1,\ldots,7$, i.e.

$$\vec{k}_j=\begin{pmatrix}k_{jx}\\ k_{jy}\end{pmatrix}=\begin{pmatrix}k_v\cos\varphi_\mu\\ k_v\cos\varphi_\mu\end{pmatrix}, k_v=2^{-\frac{v+2}{2}}\pi, \varphi_\mu=\mu\frac{\pi}{8} \tag{2.16}$$

where index $j = \mu + 8v$ and $\sigma = 2\pi$.

Figure 2.5 shows the Gabor wavelet transformation, which is done by convolving the original image with the 40 Gabor filters. The set of 40 coefficients obtained for one image point is referred to as a “jet”. A collection of these jets, together with their relative location, form an image graph, as shown on the right of the Figure 2.5.

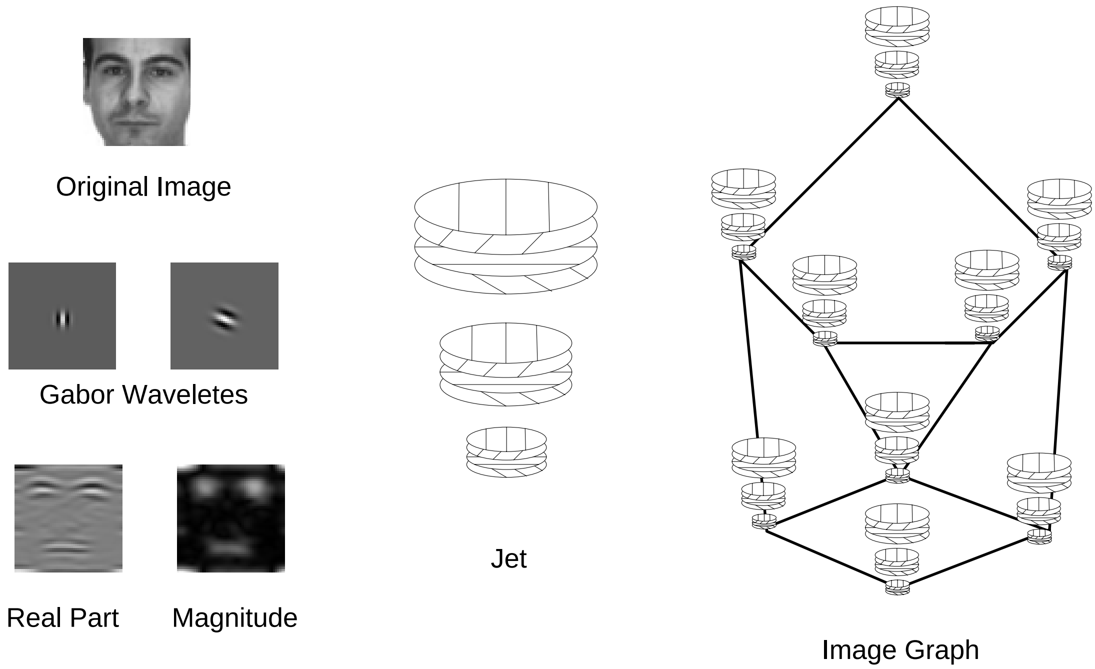


Figure 2.5 Convolution of an image and Gabor wavelets, jet of a point, image graph of the face.

To represent a face, we need to build an image graph from a set of feature points such as at the pupils, the mouth corners, the tip of the nose, the tops and bottoms of the ears, etc. A labeled graph *G* representing a face consists of *N* nodes at position $\vec{x}_n$, $n = 1, \cdots, N$ and *E* edges between them. An image graph is shown on the right side of Figure 2.5, which looks like a grid. For this image graph, 9 feature points are used as the nodes.

For an automatic face recognition system, the feature points must be located in order to automatically build an image graph for an input image. This can be done by matching the input image with a stack-like general representation of faces, called the Face Bunch Graph (FBG). A FBG consists of bunches, which are sets of jets, and can represent a wide range variation of face appearances. A face bunch graph, as shown in Figure 2.6, is a set of jets at the nodes representing the local appearance at the feature points, each with different variations. For example, the eye bunch may consist of jets of open eye, closed eye, male and female eye. With the variations, people with different facial expressions can be matched accordingly. Figure 2.7 shows the overall step of image graph matching for an image.

For the matching between an input graph and the FBG, a function called graph similarity is employed. This function measures the jet similarity and the distortion of the input image grid relative to the FBG grid. For an image graph g^I with nodes n = $1,\cdots,N$ and edges $e = 1,\cdots,E$, and an FBG B with model graphs $m = 1,\cdots,M$, the similarity is defined as

$$S_B(g^I, B) = \frac{1}{N}\sum_n \max_m (S_\phi(J_n^I, J_n^{B_m})) - \frac{\lambda}{E}\sum_e \frac{(\Delta\vec{x}_e^I - \Delta\vec{x}_e^B)^2}{(\Delta\vec{x}_e^B)^2} \qquad (2.17)$$

where λ determines the relative importance of the jets and the matrix structure, J_n represents the jets at nodes n, and $\Delta\vec{x}_e$ is the distance vector for edge e.

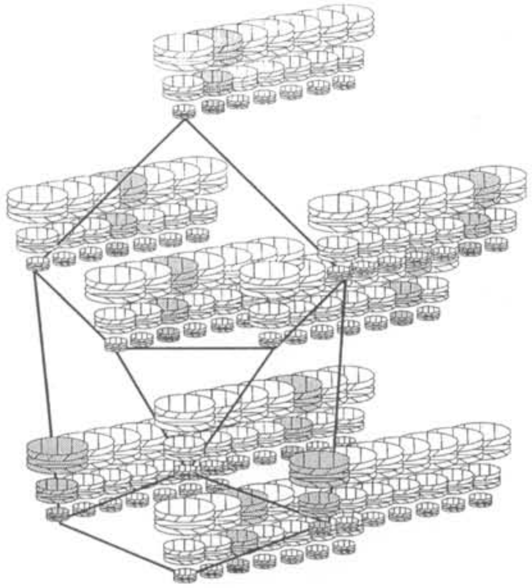


Figure 2.6 Face bunch graph.

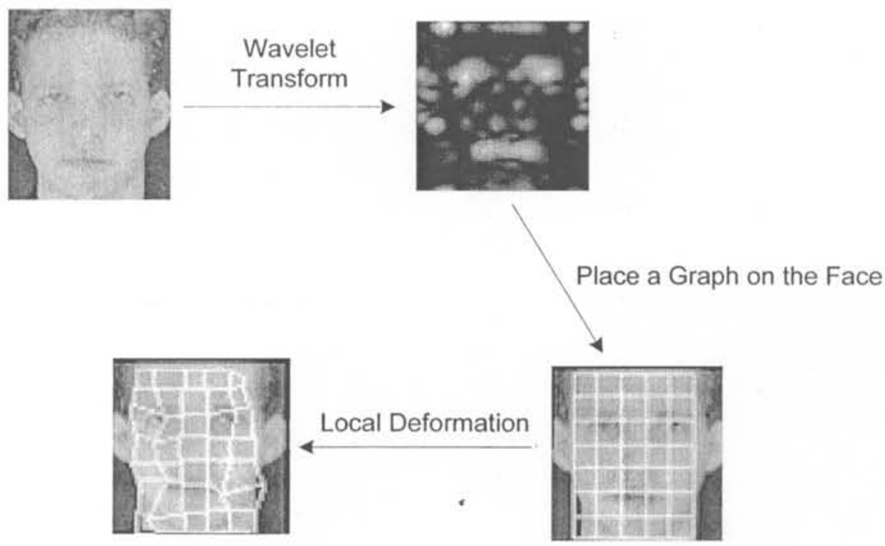


Figure 2.7 Overall steps for image graph matching.

To recognize an image, we simply compare the image graph to all the model graphs and select the one with the highest similarity value. The similarity function is an average over the similarities between pairs of corresponding jets. If g^I is the image graph, g^M is the modal graph, and node $n_{n'}$ is the modal graph corresponding to node n' in the image graph, the graph similarity is defined as

$$S_g(g^I, g^M) = \frac{1}{N'} \sum_{n'} S_a(J^I_{n'}, J^M_{n'}) \tag{2.18}$$

where the sum runs only over the N' nodes in the image graph with a corresponding node in the modal graph.

2.3.3 Hybrid Approaches

Hybrid approach uses both global and local features for face recognition. For example, the modular eigenface approach [44] uses both the global eigenfaces and the local eigenfeatures such as the eigeneyes, eigenmouth, etc. Different recognition approaches succeed and fail at different pose angle or illumination conditions. Due to this drawback, various individual recognition classifiers should be used in a recognition system. For example, if multiple views of a face image are considered in face recognition, we can employ two different approaches to handle the images. The first approach uses all the images and constructs a set of eigenfaces that represent all the images from all the views. The other approach uses separate eigenspaces for different views, so that the collection of images taken from each view has its own eigenspace. The second approach, known as the view-based eigenspaces [44], performs better.

Apart from the above approaches, Gordon [45] described a hybrid approach, which combines a frontal template and a profile template for face recognition in 1995. This approach extracts facial features to perform normalization and define template regions used for combined recognition of the frontal and profile regions in a classical template matching process (Figure 2.8).

After identifying head bounds in a frontal view, eye candidates are extracted using general eye templates, pupil detection and structural knowledge about the human head. A similar approach is used in the profile case by first extracting the profile line and then estimating the nose and chin tip. Overall template matching is subject to a scoring of five facial templates (left eye, right eye, nose, mouth and profile).

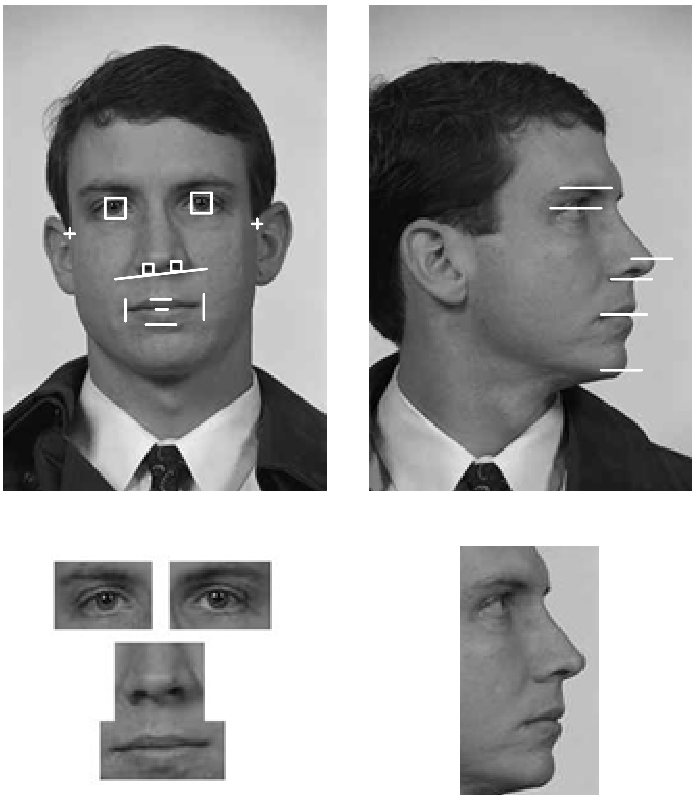


Figure 2.8 Frontal and profile templates for hybrid face recognition.

CHAPTER 3

An Accurate Active Shape Model for Facial Feature Extraction

3.1 Introduction

In Chapter 2, we presented the problem of facial feature extraction. Various approaches to human facial feature extraction have been described presented to solve the problem. In this chapter, we will introduce a modified shape model, which can adapt to face images under different orientations.

Modeling faces under different poses is a challenging problem, since the appearance of the facial features will significantly differ. The conventional active shape model (ASM) [26] uses a single face model to represent a face; this model cannot extract facial features in face images under different perspective variations accurately. The ASM is constructed based on a linear combination of a set of 2D face appearances, which are usually frontal view images. Consequently, if the input face is not of frontal view, the model cannot work properly. Therefore, in our approach, we model the face contour and the facial features separately. The shape of a face contour will be changed to a much lesser extent than those facial features under different perspective variations. For the facial features, three models are used to represent the features when the face is frontal view, turned left, and turned right, respectively.

To extract the facial features in an image, we need to fit the defined face model to a face image. To search for the best match between the model and the face image optimally, the genetic algorithm (GA) [46] is used in our algorithm. The GA operates on a population of possible solutions, which evolves by means of the crossover and mutation operations to search for the optimum solution. Each solution represents the parameters of a facial feature model for the representation of a face image. The GA has been successfully applied to many areas, such as motion estimation for video coding [47], object recognition [48], human face detection [49, 50], facial feature extraction [51, 52], etc. In our approach, we use the GA to search for the facial feature model (either frontal-, left-, or right-viewed) to be used and the parameters of the corresponding model to represent a face.

The organization of this chapter is as follows. Our new face models and the GA used in our algorithm are described in Section 3.2. Experimental results are given in Section 3.3. Finally, a conclusion is drawn in Section 3.4.

3.2 Our Proposed Algorithm

3.2.1 A New Face Model

The ASM encodes the *a priori* information about an object's shape using the Principal Component Analysis (PCA), and matches an object by applying geometrical transformation to the PDM. This method may not be able to match a new target object accurately if the shape to be represented is not present in the training set. Facial feature extraction is challenging because the human face is a three-dimensional (3-D) object, and the perspective variations of a face will affect the accuracy of extracting the facial features (e.g. eyes, nose and mouth). In our proposed approach, two shape models that are linked by a cost function are used to represent the shape of a human face. One of the shape models represents the facial features, including the two eyebrows, two eyes, nose, and mouth. The other one models the face contour only. The reason for using separate models for the facial features and the face contour is that they are affected by perspective variations to different degrees. The facial feature model is constructed by using 31 points, while 21 points are used for the face contour model. Figure 3.1 illustrates all the feature points used to represent a face, and Figure 3.2 shows the facial feature model and the face contour model. The shape of a face model can be changed using the shape parameters in (2.3), which are the weights of the respective eigenvectors or principal components generated from the corresponding training set. Figure 3.3 plots the point distribution models based on the first three principal shape parameters varied from its mean values to their corresponding maximum and minimum values ± 3 standard deviations. Varying the value of b_1 will elongate the entire facial feature. The second parameter b_2 will change the model from facing downward to facing upward. The third parameter b_3 will rotate the head to different perspectives by raising the eyebrows and opening/closing the mouth.

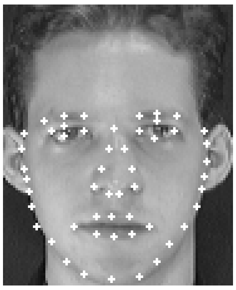
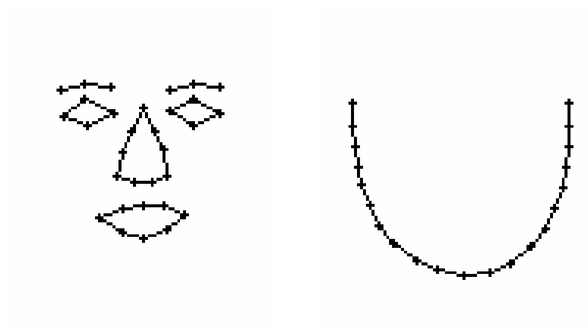


Figure 3.1 Locations of the points used to represent a face.



(a) (b)

Figure 3.2 The shape models for (a) the facial features and (b) the face contour.

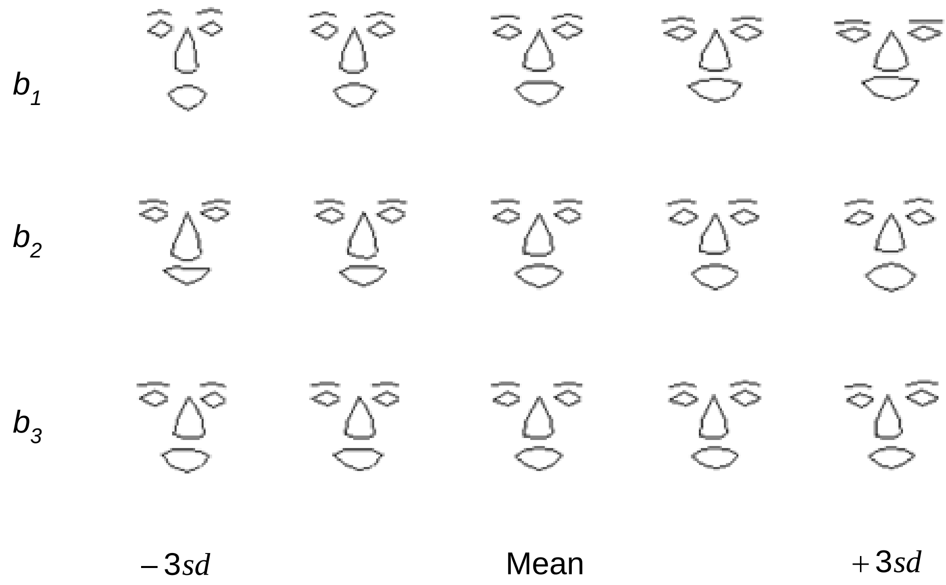


Figure 3.3 The effect of the first three shape parameters (b_1, b_2, and b_3) on shape variations.

The two individual feature models, i.e. the facial feature model and the face contour model, have their own shape variations. When they are combined, a more flexible representation of a face can be achieved. If a single face model is used to represent a whole face image, it may not be able to capture the local shape variations effectively under different facial expressions and perspective variations. Figure 3.4 illustrates some results based on the ASM, which show that it is difficult to locate the landmark points of the eyes and nose accurately if a whole-face model is used. The facial features in the whole-face shape model are related to each other in the ASM. The face model is fitted to the target object by changing the global variations of the point distribution model; the local shape variations may not be represented effectively.

For example, while changing the shape parameters to represent an open or close mouth, other facial features and the face contour will also be modified. This is the drawback of the ASM model. Moreover, ASM is restricted to a narrow view because the model can only represent a shape that is a linear combination of the 2D appearance in the training set. To solve this problem, nonlinear models or mixture models are adopted to model the face under perspective variations [33, 34]. Therefore, we use two separate shape models, one for the face contour and the other for the remaining facial features, with some constraints on the shapes and relative position of these two models.

In the optimization process, the facial feature model and face contour model have their own transformation parameters (or pose parameters), which include the scaling factor, the rotation angle, and the (x, y) translation. Consequently, this allows the respective shape models to deform according to the corresponding features, and the two shape models are combined to represent a face more accurately.

The ASM is restricted to representing faces form a narrow frontal view. Figure 3.5 illustrates that the ASM cannot effectively extract the face image under different viewing angles. To extract the facial features under different viewing angles during the searching process, three facial feature models are used for their respective angles. The left-viewed model describes faces with a viewing angle θ between 10° and 45°

to the left (i.e. -45°< θ <-10°), the frontal-viewed model for faces with an angle between -10° and 10° (i.e. -10°< θ <10°), and the right-viewed model for faces viewing between 10° and 45° to the right (i.e. 10°< θ <45°). Since the ASM is constructed based on a linear combination model, the feature model cannot capture a full range of pose change. These three models are shown in Figure 3.6. The aim is to use the facial feature model which best represents the local facial features under the corresponding viewing angle. The differences between these three models are in their representations of the nose and in their appearances under the different viewing angles. A closed contour is used for the nose in the frontal view model, while an open contour for the two side-viewed models. The number of points for a nose represented by the closed contour is more than that represented by the open contour. The PDMs for the frontal view and the two side views are illustrated in Figure 3.7. Each of these three models is trained based on a set of face images under a corresponding range of viewing angles, and the points for the PDMs are labeled manually. As the left-viewed and right-viewed models are the reflected version of each other, so these two models can be trained together.

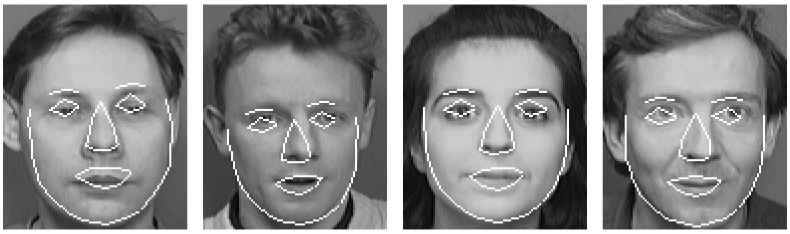


Figure 3.4 Some results of applying ASM to frontal face images.

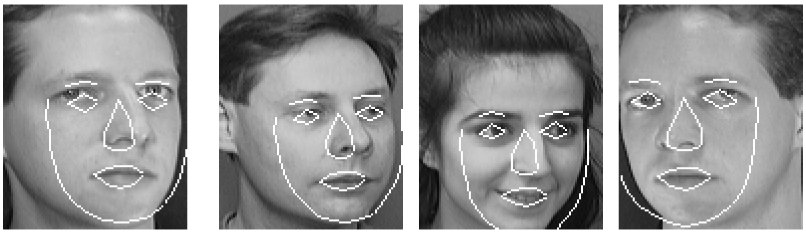
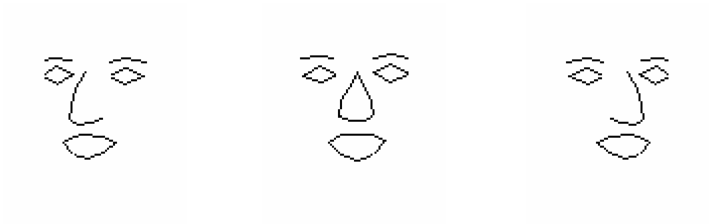
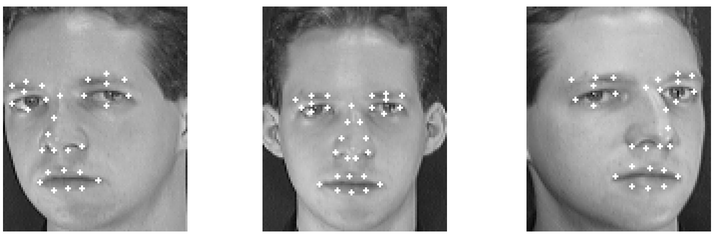


Figure 3.5 Some results of applying ASM to faces under perspective variations.



Left view	Frontal view	Right view
$-45° < \theta < -10°$	$-10° < \theta < 10°$	$10° < \theta < 45°$

Figure 3.6 The left-viewed, right-viewed and frontal-viewed face models, and their corresponding ranges of viewing angles.



Left Frontal Right

Figure 3.7 Examples of training images for the three different facial feature models.

3.2.2 Facial Feature Extraction Using the Genetic Algorithm

In this section, we present the use of the face shape models and the genetic algorithm (GA) to extract a face image in a gray-scale image. The three face shape models, as described in Section 3.2.1, are randomly generated and evenly distributed in the initial population. The fitness value of each candidate in a population is measured based on the gray-level appearance and the edge information. When the population evolves, the number of candidates with the correct face shape model will gradually dominate. The process will be stopped either when the average fitness value of the population does not change significantly over a number of iterations or after a certain number of iterations have been done. Finally, the parameters of the best candidate in the population are used to represent the face image.

The genetic algorithm is used to search the correct facial feature model and its corresponding optimal shape and pose parameters in representing a human face. This is an effective approach when the searching space is large. Each candidate in a population is associated with a fitness value, which measures how well the candidate can represent the required solution. The solution represented by a candidate is encoded as a chromosome, which is the basic element in the GA. In each iteration, a new population is generated based on the three genetic operators: selection, crossover and mutation. In our approach, candidates are selected for the new population based on their corresponding fitness values with the roulette wheel selection [46]. In other words, a candidate with a higher fitness value will have a higher probability of being selected for the next generation. The crossover and mutation operations are applied to the selected candidates to form the new population. For crossover, two selected parent chromosomes are cut at a random bit position and are then combined in a crossover manner to form two new children. For mutation, each bit in a chromosome may be mutated, i.e. changing from 0 to 1 and vice versa, with a certain probability.

Structure of a Chromosome

Each possible set of parameters of the face shape models is represented by a chromosome, which is in binary form. In our algorithm, a model is described by two components: the shape parameters and the pose parameters. The shape parameters are used to represent the variation of the shape model, while the pose parameters correspond to the scale, rotation, and position of the model. Both the facial feature model and the face contour model have the same number of pose parameters, i.e. scale, rotation, and the x and y translation of the model. However, we use 7 shape parameters for the facial feature model and 3 for the face contour model. This is because the facial features are more complicated and require more shape parameters in order to achieve an accurate representation. The structure of a chromosome and a summary of the parameters are illustrated in Figure 3.8 and Table 3.1, respectively. The 7 shape parameters used for the facial feature model, which are used to describe the eyes, nose, mouth, etc, can capture 95% of the variations in 80 training samples. These training samples are randomly selected from a database with a similar face scale but a different facial expression. Hence, 7 shape parameters (b_1, b_2, …, b_7) and 4 pose parameters (scaling s_1, rotation θ_1, Tx_1, and Ty_1) are defined for the facial feature model. The face contour is represented as a smooth curve, so 3 shape parameters are sufficient to represent most of its variations in the training set. Consequently, a total of 18 parameters (10 shape parameters and 8 pose parameters) are to be determined in the GA optimization process. The relationships between the two feature models are enforced by a cost function, which will be described in the next section. The allowed changes in scale on both the facial feature and the face contour models are within ±10% of their respective original sizes. In our approach, four bits are used to represent the change in scale. In addition, the difference in size between the two models is limited to 20%. The range of the in-plane rotation angle is within the range [-32°, 32°], which is represented by using 6 bits. The translation of a model in each of the directions (horizontal and vertical) is also represented by using 6 bits. The variations in each of the shape parameters, b_i, is within the corresponding three standard deviations from the mean. Each of these shape parameters is also represented by using 6 bits.

Chromosome:

7 shape parameters (b_1 - b_7)	4 pose parameters s_1, θ_1, Tx_1, Ty_1	3 shape parameters (b'_1 - b'_3)	4 pose parameters s_2, θ_2, Tx_2, Ty_2

Facial Feature Model **Face Contour Model**

Figure 3.8 Structure of a chromosome.

Parameters	Model	Number of bits	Ranges
b_1 to b_7	Facial Feature Model	6 for each b_i	64 intervals within ±3 standard deviations from the mean
Scale, s_1		4	change within ± 10%
Rotation Angle, θ_1		6	-32°< θ_1 <32°
Tx_1		6	±32
1.4.1.1.1 Ty_2		6	±32
b'_1 to b'_3	Face Contour Model	6 for each b_i	64 intervals within ±3 standard deviations from the mean
Scale, s_2		4	change within ± 10%
Rotation Angle, θ_2		6	-32°< θ_2<32°
Tx_2		6	±32
Ty_2		6	±32

Table 3.1 Summary of the model parameters.

With different combinations of the facial feature models and the face contour model, a population of possible face models with different locations, sizes, and orientations can be generated. The initial population is formed by selecting the three facial feature models randomly and in equal proportion. Therefore, if the population size is N, the numbers of left-viewed, right-viewed, and frontal-viewed face models are equal to $N/3$. The top n candidates with the highest fitness values for each of the facial feature models are retained and passed on to the next generation without performing the genetic operations. After a number of generations, one of the face models will dominate in the population. Then, all the candidates will employ this dominated facial feature model, and the final population size is reduced to $N/3$. The fitness of a face model is determined by means of the gray-level appearance model and edge information. Finally, the iteration will be terminated if there is no further improvement in the fitness value of the population or a maximum number of generations have been evolved. Then, the chromosome with the maximum fitness value is selected as the optimal solution. The corresponding parameters of the chromosome can provide a good representation of the face image under consideration. In our GA implementation, we have proposed two modifications. (1) The size of the population is reduced to one-third of the initial population when a dominant facial feature model has been identified. In addition, the population size is reduced by 10%

after each generation, until it becomes one-third of the initial population. This allows us to have a large population to make the search more effective at the beginning, and reduces the population and the required computation when potential candidates have been identified. In order to determine the optimal population size to be used, 270 testing face images were selected, and the initial population sizes for each of the three facial feature models were set at 50, 100, and 150. Table 3.2 tabulates the average pixel errors for the three different initial population sizes. It is found that the lowest average pixel error can be achieved when the population size for each of the facial feature models is 100. Therefore, in our following experiments, the initial population size is set at 300, which will gradually be reduced to 100. Furthermore, the maximum number of generations is set at 100. (2) A chromosome is composed of four components (two for the shape parameters and the other two for the pose parameters). The crossover operation is carried out within each of the components. The crossover process is illustrated in Fig. 3.9 Four sets of probabilities are employed, and are changed in two stages of the GA process. In the first stage, the crossover and mutation probabilities for the two sets of pose parameters are assigned with higher values, while smaller probabilities are adopted for the shape parameters. This is because, at the beginning of the search, the local optimal face location must be searched; otherwise the shape parameters will be meaningless. In our algorithm, the probabilities of crossover and mutation for the pose parameters are 0.8 and 0.05, respectively, and the corresponding probabilities for the shape parameters are 0.3 and 0.01, respectively. When one of the face models dominates the population (we set it at 80% of the total population), the GA will jump to the second stage. However, if an input face has a viewing angle in between of the frontal view and a side view, the population will not be dominated by a single face model. In this case, the population will consist mainly of faces represented by the frontal-view model and the side-view model. Consequently, the GA will not jump to the next stage. The iteration will stop under the same conditions and the population size also decreases to one-third of the initial population by a factor of 10% after each generation.

In second stage, the crossover and mutation probabilities for the pose parameters are reduced to 0.3 and 0.01, respectively. Therefore, the pose parameters for the face model will be subject to a smaller perturbation. However, the crossover and mutation probabilities for the shape parameters will be increased to 0.8 and 0.05, respectively. This will allow the algorithm to search for the optimal shape parameters more effectively. The above setting of the probabilities is obtained based on experiments that can achieve the best representation results for our algorithm.

Initial population size for each facial feature model	Total population size	Average pixel error
50	150	2.69
100	300	2.28
150	450	2.30

Table 3.2 Average pixel errors with different initial population sizes in the three view models.

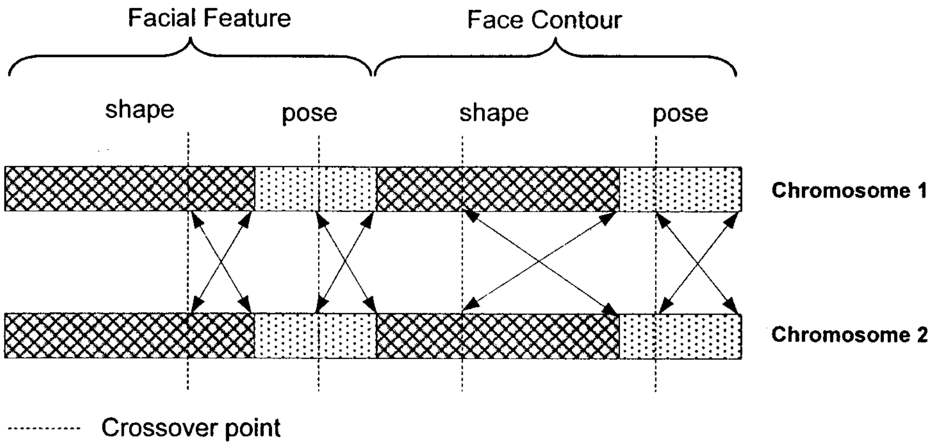


Figure 3.9 Four crossover operations carried out in a pair of selected chromosomes.

The Fitness Function

To measure the fitness of a face model described by a chromosome to represent a target face image, a fitness function is defined as follows:

$$F = w_1 f_{facial} + w_2 f_{contour} + w_3 f_{con}, \tag{3.1}$$

where f_{facial}, $f_{contour}$, and f_{con} are the functions used to evaluate the fitness of the facial feature model, the fitness of the contour model, and the fitness of combining these two models, respectively, and w_1, w_2, and w_3 are the corresponding weights for these three functions.

To improve the fitting performance of the face model to a face image, we employ not only the edge information but also the gray-level appearance of the image. Therefore, the fitness value of a chromosome is computed based on both the edge intensities and gray-level appearance of the image under consideration. The fitness functions for facial feature representation, f_{facial}, and face contour representation, $f_{contour}$, are defined as follows:

$$f_{facial} = \lambda_1 \frac{1}{S} \sum_C E(x,y) - \lambda_2 \left(\sum_{i=1}^{N_1} \mathrm{M}(\mathbf{y}_i, \quad \bar{\mathbf{y}}_i) \right) - \lambda_3 \sum_{i=1}^{N_1} \left| \frac{D_i}{Maxlength} \right| \text{, and} \tag{3.2}$$

$$f_{contour} = \lambda_4 \frac{1}{S} \sum_C E(x,y) - \lambda_5 \left(\sum_{i=1}^{N_2} \mathrm{M}(\mathbf{y}_i, \quad \bar{\mathbf{y}}_i) \right) - \lambda_6 \sum_{i=1}^{N_2} \left| \frac{D_i}{Maxlength} \right| , \tag{3.3}$$

where λ_i (i = 1, …, 6) represent the weighting factors, $E(x, y)$ is the edge intensities of the image, S is the number of pixels in the face model C, $\mathrm{M}(\mathbf{y}_i, \bar{\mathbf{y}}_i)$ is the Mahalanobis distance between a new profile, $\mathbf{y}_i$, and the mean profile , $\bar{\mathbf{y}}_i$, and N_1 and N_2 are the number of points on the facial feature model and the face contour model, respectively. *Maxlength* represents the maximum length of the profile on one side. For example, if the profile length is 19 pixels (9 pixels on both side, and itself), the *Maxlength* is equal to 9. Each of these cost functions is divided by their corresponding maximum value for normalization. Both (3.2) and (3.3) have the same form but with different weighting factors for the respective terms. The first term is the average edge intensity over the model used. The edges of the image are smoothed by means of a

Gaussian function so that the fitness of the model can also be computed even if it is not located exactly at the object to be searched. The second term is the Mahalanobis distance of the normalized derivative profile perpendicular to the object boundary at each of the landmark points. This term measures the local texture similarity between the local profile on each landmark point based on the face model and the corresponding profile from the training set. The third term is a measure of the relative distance from the landmark points to the strongest edge point along the profiles (see Figure 3.10). The distance, D_i, between the landmark point under consideration and the actual image boundary is to be minimized. In general, most of the feature points should be located at strong edges. If the relative distance D_i is equal to zero, we assume that the landmark point is best fitted to the boundary of the target image.

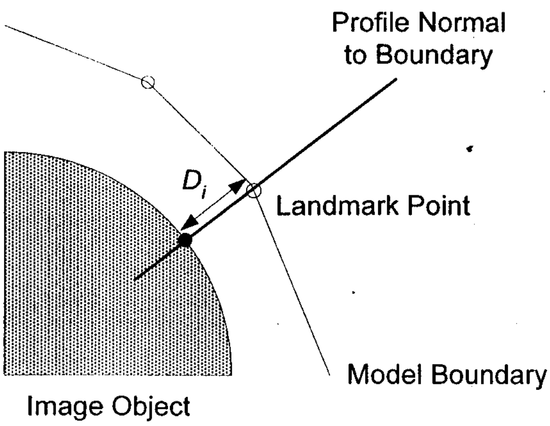


Figure 3.10 The distance measured between a landmark point and the actual image boundary.

The facial feature model and the face contour model are combined to form a face model in a chromosome. These two models are related to each other to form a valid face model. The validity of this combination is governed by the third term in (3.1), which is called the constraint fitness function. This term imposes constraints on possible combinations of the two models, and enforces the angles θ_{left} and θ_{right}, as shown in Figure 3.11, so they are as small as possible. The angle, θ_{left} or θ_{right}, is formed between the baseline of the two eyes and the line joining the corresponding eye center and the corresponding end point in the face contour model. The baseline is the line passing through the centers of the two eyes. Suppose that (x_1, y_1) and (x_2, y_2) are the centers of the right eye and left eye, then the equation of the baseline is $ax+by+c=0$, whose coefficients a, b, and c can be calculated as follows:

$$a = y_2 - y_1,\ b = x_1 - x_2, \text{ and } c = x_2y_1 - x_1y_2. \tag{3.4}$$

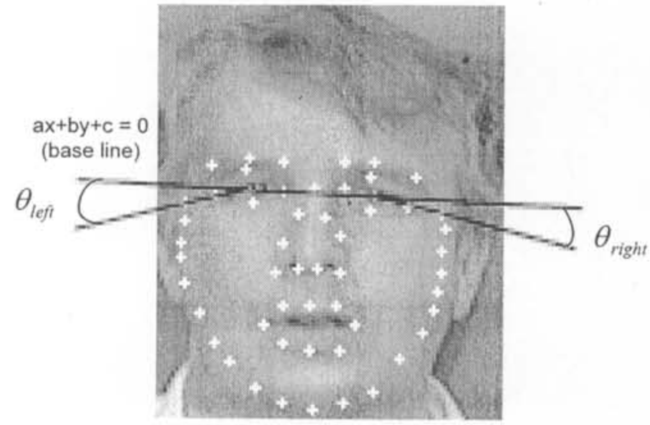


Figure 3.11 The angles θ_{left} and θ_{right}.

The two end points of the face contour model are connected to the corresponding eye center on its side to form the angles θ_{left} and θ_{right} with the baseline. To determine the angles θ_{left} and θ_{right}, let (x_e, y_e) and (x_c, y_c) be the coordinates of an end point of the face contour model and the eye center, respectively. Then, the distance d_1 between (x_e, y_e) and the baseline and the length d_2 between (x_e, y_e) and (x_c, y_c) are given by

$$d_1 = \frac{|ax_e + by_e - c|}{\sqrt{a^2 + b^2}} \quad \text{and} \quad d_2 = \sqrt{(x_e - x_c)^2 + (y_e - y_c)^2}\ . \tag{3.5}$$

The angle can therefore be calculated as follows:

$$\theta = \sin^{-1}\left(\frac{d_1}{d_2}\right). \tag{3.6}$$

The fitness function for this constraint model is:

$$f_{con} = \exp\{-(\theta_{Left} + \theta_{Right} + |\theta_1 - \theta_2|)\}, \tag{3.7}$$

where θ_1 and θ_2 are the rotation angles of the facial facture and face contour models. To maximize this fitness value, θ_{left}, θ_{right}, and $|\theta_1 - \theta_2|$ will be forced to be as small as possible. In additional, an associated penalty term will be added in the fitness evaluation to prevent overlapping the two models in the next generation. If any of the five points shown in Figure 3.12 is outside the face contour model, the weighting value w_3 will be set to a large negative number. Hence, with these constraints, the two models can be linked up to form a valid representation of a face in the optimization process.

Figure 3.12 Additional constraints on some points in the facial feature model.

3.3 Experimental Result

To evaluate the accuracy of our proposed algorithm, the average error, *e*, is defined as the average distance between the landmark points searched by the ASM or other algorithms and their actual positions. Therefore, *e* can be computed as follows:

$$e = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(a_{i,x} - b_{i,x})^2 + (a_{i,y} - b_{i,y})^2}\ , \tag{3.8}$$

where $\boldsymbol{b}_i = (b_{i,x}, b_{i,y})$ represents the correct position of the landmark points for a face image, $\boldsymbol{a}_i = (a_{i,x}, a_{i,y})$ the corresponding positions obtained by a search algorithm, and N the total number of points used in the face model.

The experiments were performed with the Olivetti Research Lab (ORL) face database. The ORL database contains 40 distinct subjects, with ten images per subject. The facial images of each subject are taken at different time instances, with varying perspective variations, facial expressions, facial details (glasses or no glasses), and each image is of size 92×112. All subjects are in the upright, frontal position, with up to 45 degrees of perspective change. The face images are divided into three categories according to the viewing angles (left, right, frontal), and the landmark points in the images are manually labeled. Fifty-two landmark points are used for frontal view images, and 50 for left- or right- side views. The number of face images in the respective categories, and the corresponding numbers of training images and testing images are tabulated in Table 3.3. For each of the left- and right-viewed categories, 25 labeled face images were randomly selected as the training samples, and the remaining 30 and 28 face images were used as the testing set. Therefore, a total of 50 training samples were used to form the left- and right-viewed model, as the left-viewed model is simply a reflection of the right-viewed model. For the frontal-view category, 80 faces were also selected randomly to form the training samples, and the remaining 212 faces form the testing set.

Category	Number of face images	Training set size	Testing set size
Left	55	25	30
Right	53	25	28
Frontal	292	80	212
Total	400	130	270

Table 3.3 The numbers of face images in each of the three categories for the ORL database.

Two sets of experiments were conducted. The first experiment compares the performance of both the ASM and our algorithm by using the frontal-view images only. The second experiment explores the performance of the ASM and our algorithm when the faces are under different perspective variations. In the first experiment, both the ASM and our algorithm were trained with the same 80 frontal-view face images, and the remaining 212 images were used for testing. For each testing image, 25 different initial positions – which are horizontally and vertically displaced from the true position by 0, 10 and 20 pixels – are considered in our experiment. The mean face model is used as the initial shape in the ASM search. Only the frontal-view facial feature model with different shape, scale, orientation and translation parameters were generated randomly for the initial population in our algorithm. The searching process will stop unless the results converge or the number of iterations for both ASM and our proposed algorithm is more than 100. By using the frontal-view images in the experiments, the overall average errors of ASM and our proposed approach are 3.97 and 2.03, respectively. Some results based on these two methods are illustrated and compared in Figure 3.13. These results show that the locations of the eyes and mouths achieved by our approach are more accurate than those achieved by ASM. Our algorithm employs the GA, which holds a population of possible solutions and performs the search in parallel. This can therefore provide a more effective search for the optimal solution. For ASM, the search will take a much longer time and the accuracy will become poorer if the initial candidate is not close to the target face image concerned.

With a 1.4GHz Pentium IV computer system, the average runtime for ASM is 210ms only. Our proposed algorithm, which requires 2.8s on average, is much more computationally expensive. This is because the GA requires a lot of overheads to encode a face model to a bit-string chromosome, and to decode the chromosome back to the face model in order to evaluate the fitness value.

Our algorithm can search for the correct face model (frontal, left, or right) to be used to represent a face, and can then use that model to represent the face accurately. In the second experiment, we used 130 training images and 270 testing images, and measured the accuracy of selecting the face models by means of our algorithm. The results are illustrated in Table 3.4, which shows that the respective accuracies of our algorithm are 93.3%, 96.4%, and 98.1%, for the left-, right- and frontal-view face images. On average, the accuracy is 95.9%, i.e. 263 out of 270 faces are represented with a correct face model. Of these 263 face images, the average error is 2.21. The results, based on both ASM and our algorithm, are shown in Figure 3.6. Figure 3.15 illustrates other results using our algorithm; the images are in the left- or right-views. For the purposes of comparison, the same setting, i.e. the number of training images and testing images, is used to evaluate the accuracy of the ASM under the three categories of viewing angles. The results are presented in Figure 3.5, which shows that the ASM fails to locate the face contour and nose for the left- and right-viewed faces. Table 3.5 also shows the average errors based on the ASM and our algorithm. It can be seen from the table that our algorithm is much more accurate than the ASM. In addition, the average errors based on the ASM and our algorithm for frontal-view images are higher than those in the first experiment. It is because the training set used in this experiment includes two extra face models with different viewing angles. Some failed results are shown in Figure 3.16, where it can be seen that the mouth and nose features cannot be extracted successfully. The failure of our algorithm is mainly due to the fact that the edge intensities over the lips are very weak or, in some cases, a mustache is present or covers most parts of the mouth region.

	Correct	Fail	Accuracy in %
Left	28	2	93.3%
Right	27	1	96.4%
Frontal	208	4	98.1%

Table 3.4 The accuracy of selecting a correct face model.

	Number of Test Images	ASM ($\mu \pm \sigma$)	Our algorithm ($\mu \pm \sigma$)
Left	30	5.07 ± 0.60	2.68 ± 0.35
Right	28	4.93 ± 0.65	2.48 ± 0.30
Frontal	212	4.46 ± 0.61	2.20 ± 0.32
Average		4.58	2.28

Table 3.5 The average errors based on ASM and our algorithm under the three categories of viewing angles.

(a) Results based on ASM.

(b) Results based on our algorithm.

Figure 3.13 Some matching results for frontal-view images based on the ASM (top row) and our proposed approach (bottom row).

(a) Results based on ASM.

(b) Results based on our algorithm.

Figure 3.14 Results for left-viewed and right-viewed face images based on the ASM (top row) and our approach (bottom row).

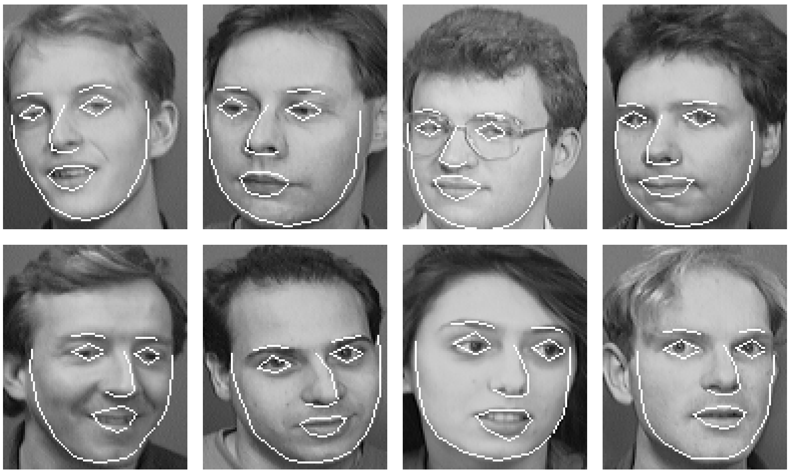


Figure 3.15 Other results for left-viewed (top row) and right-viewed (bottom row) face images based on our approach.

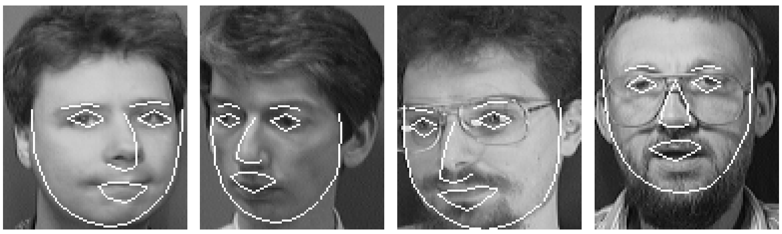


Figure 3.16 Some failed results.

Conclusion

The original ASM considers the frontal view face images in facial feature extraction. This method cannot perform well when the face image concerned is not of frontal view. In order to extract facial features accurately under different perspective variations, we have proposed a face model that consists of a facial feature model and a face contour model. Three individual views for the facial feature model (left-side, right-side and frontal model) are employed, and the GA is applied for searching for the face model to be adopted, so that the face image can be represented accurately. These three face models are also incorporated with the specific local texture features around each of their landmark points in the fitness function. Experimental results show that our proposed algorithm can search for the face model and the facial features of a face image accurately. The face model can also be used to estimate the head pose of the face image. The improvement achieved by our algorithm is mainly due to the use of separated models for facial features and face contour, a fitness function that considers the edges, texture, and the constraints on combining the two models, and the GA, which can search for the optimal solution accurately.

CHAPTER 4

Pose Estimation Based on Shape Model Parameters

4.1 Introduction

In Chapter 3, we presented an efficient and reliable approach for extracting the facial features such as eyes, nose, mouth and face contour. Based on the relative position of the facial features, we can estimate the head pose for face recognition. In general, face recognition will not achieve an accurate result when the pose of the input face is significantly different from those in the training set. Therefore, pose estimation should be performed, and the input image in the matching process should be changed accordingly. As we know, once the pose of a new input face is known, we can either modify the face in order to compensate the effect of the pose or compare the input to those faces in the database with the same pose. For the latter approach, the face images of a class must be divided according to their corresponding poses.

There are many existing methods for head pose estimation. These methods can be roughly divided into two categories: appearance-based approaches [53-55] and model-based approached [56, 57]. Appearance-based approaches treat the whole face as a feature vector in some statistical subspaces and do not require facial feature detection for preprocessing. It has recently become a popular method. Appearance-based approaches generally use multi-view classification [58] for head pose estimation. The multi-view classification methods divide the range of angle of a face into several intervals. A classifier is required to be built for each interval. Its accuracy depends on the classification performance for each interval and the angle range in each interval. T.F. Cootes and C.J. Taylor [57] described a model-based approach for locating a face outline. The model used is derived from a set of training face images. The shapes of the main features and the spatial relationships between them are represented by a Point Distribution Model (PDM). This provides a compact, parameterized description of shape for any instance of a face. Cootes and Taylor used the first and third parameters of the shape model, which are extracted from the PDM, for estimating the 3D pose of human faces. In this chapter, we propose a reliable method for estimating the pose of a human face. The idea of T.F. Cootes's model-based approach is employed. The 3D pose angle is further represented by using more shape model parameters. We first evaluate the shape variation of the face shape model with respect to each of the shape parameters, and then determine those reliable shape parameters which can provide useful information about the pose of a human head. Based on the analysis of these parameters, multiple regression is used to calculate the relationship between the range of pose angles and the shape parameters.

4.2 Building the Shape Variation Model

Firstly, principal component analysis (PCA) is applied to model the shape variation of human faces under different poses. We manually locate the facial feature points on 200 face images, which are selected randomly from the ORL face database. Since the ORL database contains face images of different poses, these images are useful for constructing the shape variation model. Some of the face images are shown in Figure 4.1. The coordinates of the facial feature points are collected to form a shape vector $\boldsymbol{x} = (x_0, y_0, x_1, y_1, \ldots, x_k, y_k, \ldots, x_{n-1}, y_{n-1})^T$, where ($x_k$, y_k) are the coordinates of the kth feature point. PCA can approximate any of the original points in the N-dimensional space with smaller dimension t, where $t \ll N$. The vector $\boldsymbol{b}$ is defined as a set of shape model parameters as follows:

$$\boldsymbol{b} = \boldsymbol{P}(\boldsymbol{x} - \overline{\boldsymbol{x}}) \tag{4.1}$$

where $\overline{\boldsymbol{x}}$ is the mean shape vector and $\boldsymbol{P}$ contains the t eigenvectors corresponding to the largest eigenvalues.

By varying the elements of $\boldsymbol{b}$**,** we can generate new examples of the face shape model. The resultant shapes obtained by varying the first six shape parameters independently are shown in Figure 4.2. As illustrated in the first row, varying the first parameter b_1 rotates the head in the yaw direction and changes the eyes, nose and face contour vertically. Varying the second parameter b_2 changes the length of the face.

Varying the third parameter b_3 moves the head in the tilt direction. Varying the fourth, fifth and sixth parameters controls the mouth (open/close), the shape of the chin contour, and the direction of the nose bridge, respectively. These 3D pose variations in the shape model can therefore be found in the first several shape model parameters. By changing these six parameters at the same time, we can construct different 3D pose variations such that the faces have different yaw and tilt angles and different appearance at the eyes, nose, mouth and chin. Figure 4.3 illustrates some examples of the 3D poses generated. The effect of the remaining parameters on pose variation is much smaller then the first several parameters from the principal component analysis. These remaining shape parameters also produce similar kinds of effect, but the effect is implicit.

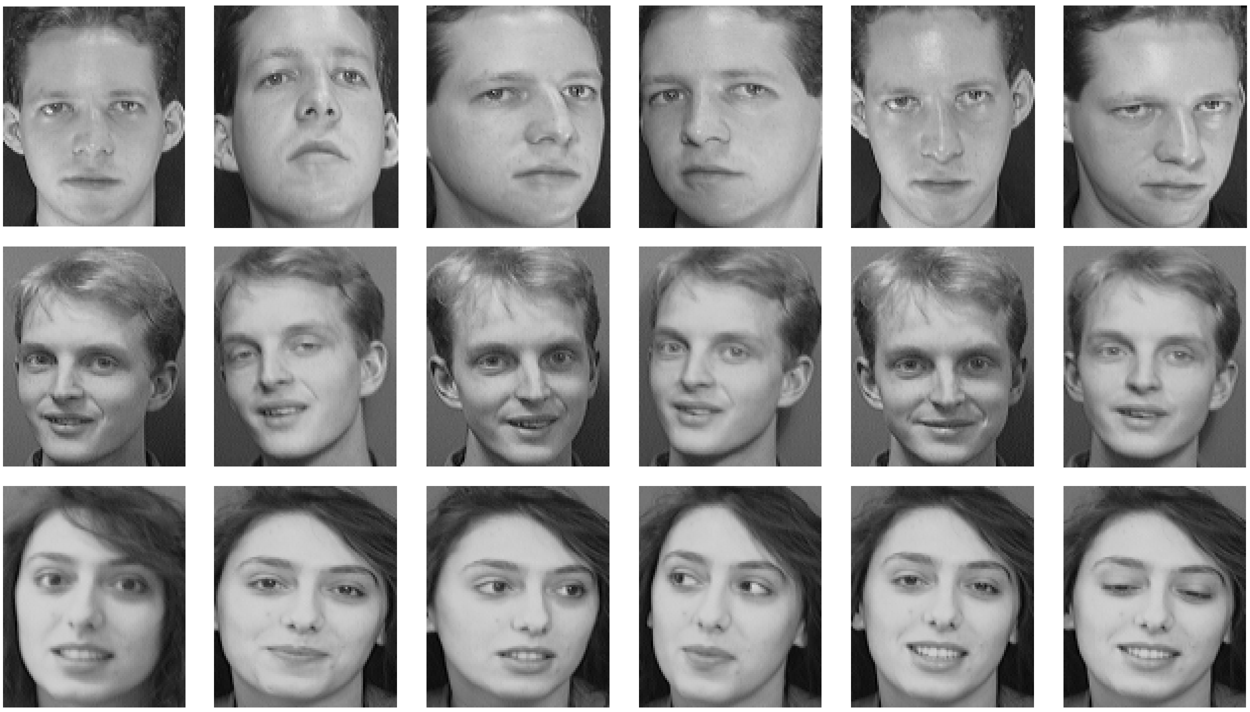


Figure 4.1 Sample face images from the ORL database.

Figure 4.2 Shape variation by varying the first six shape model parameters.

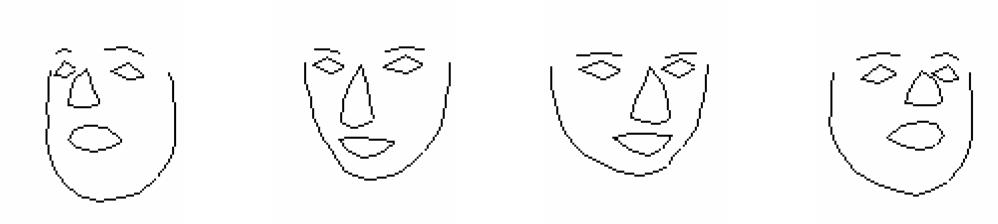


Figure 4.3 Different 3D pose variations.

In order to understand the relationships between the pose angle and the first shape parameter, Figure 4.4 plots b_1 against the yaw angle from – 40 to + 40 degrees. In this figure, the face images of ten different subjects under different perspective variations are used. Some of the sample images are shown in Figure 4.5. Linear regression is employed to understand the relationship between the first shape model parameter b_1 and the pose angles. Figure 4.4 shows that the angle increases linearly as b_1 decreases. Hence, a simple linear equation, $y = a + bx$, having a negative slope, is calculated from the data set.

From Figure 4.2, we can observe that the face shape variations caused by the sixth shape parameter b_6 include the viewing angle of a face, the direction of the nose bridge, and the direction of the mouth contour. Hence, we assign an appropriate yaw angle for each value of b_6, and plot the graph b_6 against the yaw angle ranging from – 40 to + 40 degrees, as shown in Figure 4.6. In our approach, we extend the original pose estimation, which is based on the linear relationship between the angle and a single parameter, by employing more than one shape model parameter. Hence, the two shape parameters b_1 and b_6 are used to estimate the yaw angle. They are combined to form a multiple linear regression model as follows:

$$y = \beta_0 + \beta_1 b_1 + \beta_2 b_6 \tag{4.2}$$

where y is the estimated angle, and β_0, β_1 and β_2 are the regression coefficients.

When a new face image is presented, the coordinates of the facial feature points are extracted by means of the active shape model described in the previous chapter. Then, the shape parameters $\boldsymbol{b}$ are calculated by projecting the shape vector $\boldsymbol{x}$ onto the principal components. The resulting value of b_1 and b_6 are recorded for pose angle estimation. The pose estimation of yaw are analyzed in our investigation, the tilt variation can be approximated with the same way as the yaw variation.

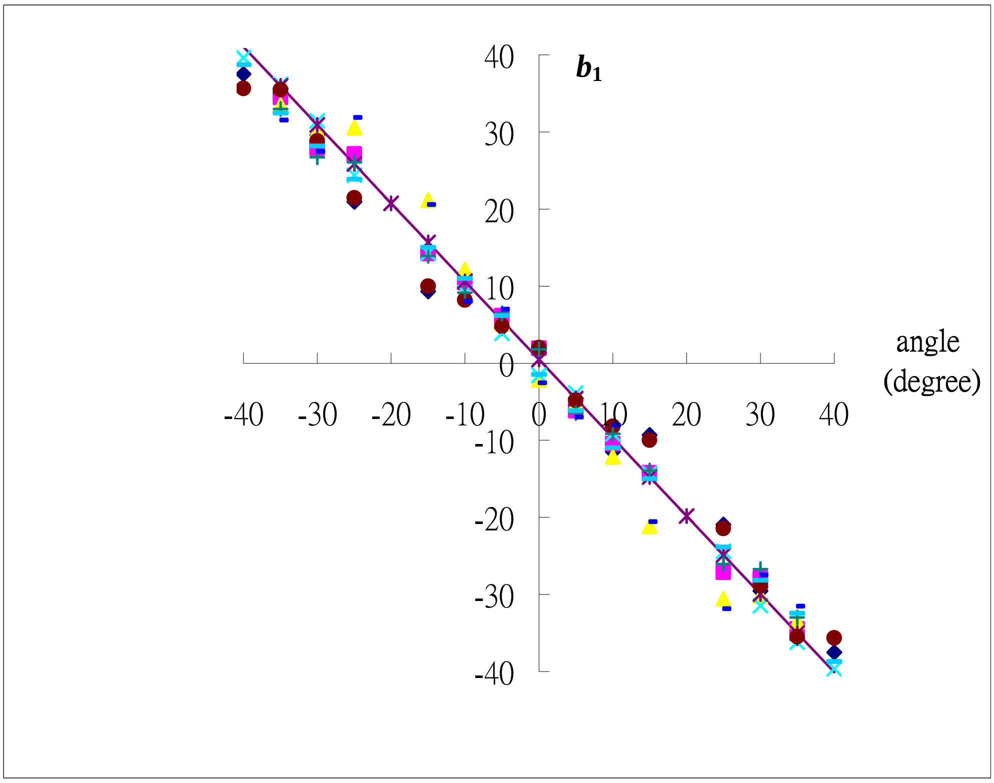


Figure 4.4 Graph of b_1 against angle range from – 40 to + 40 degrees in yaw.

Figure 4.5 Face image samples (angle range from – 40 to + 40 degrees in yaw).

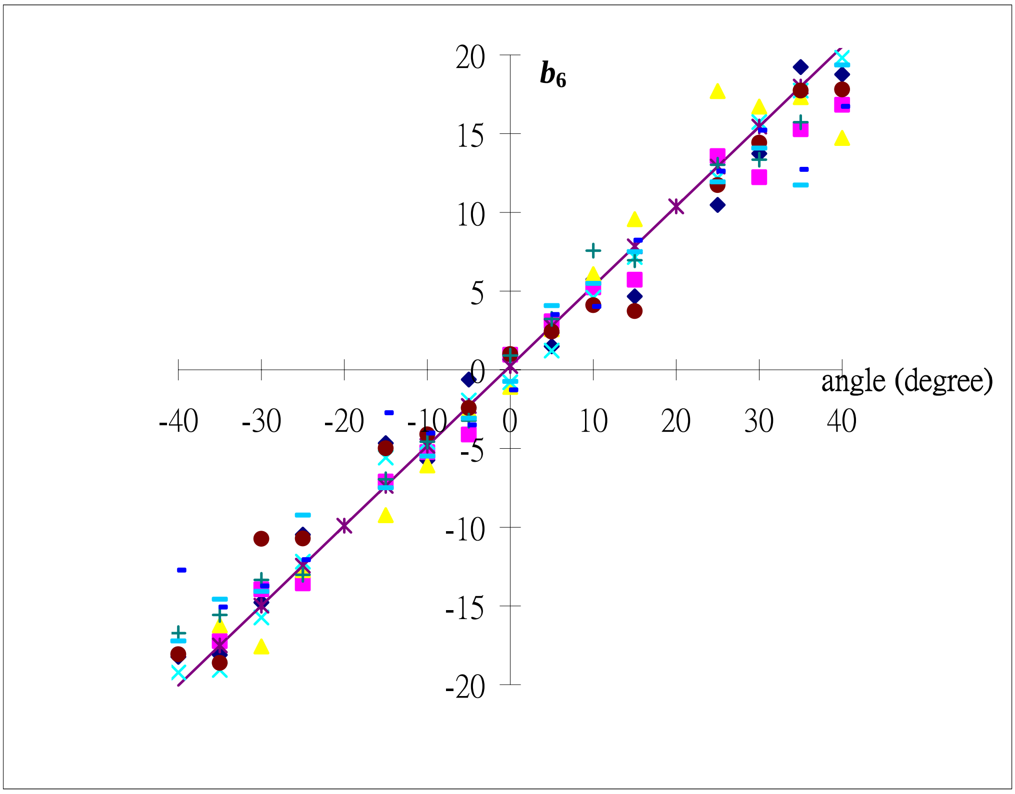


Figure 4.6 Graph of b_6 against angle range from – 40 to + 40 degrees in yaw.

4.3 Experimental Results

The accuracy of pose angle estimation is tested based on 70 test images, which were selected from the FERET face database. The FERET database is chosen because it consists of 200 subjects, each of which has a series of images with slightly different facial expressions and poses in a range of ± 40 degrees from the frontal pose (see Figure 4.5). Experimental results based on using b_1 only, b_6 only, and our approach (i.e. combining the two parameters) are tabulated in Table 4.1. From the results, we can obverse that our approach is robust to pose angles. This estimated pose can be employed for face recognition such that either compensation for the pose can be performed, or faces with the same pose in the database can be used.

Angle error within	±5°	±10°	±15°
Accuracy using b_1	77%	89%	98%
Accuracy using b_6	71%	85%	89%
Accuracy using b_1 and b_6	80%	92%	100%

Table 4.1 Pose estimation results.

4.4 Conclusion

In this chapter, a model based approach for estimating the head pose has been described. The shape variation of the face model based on the first several shape parameters has been investigated. Besides using the first shape parameter b_1 for estimating the head pose, another appropriate shape parameter b_6, which also provides useful information about the pose of a human head, is added. Each of these two parameters can form a linear model to estimate the pose. In our approach, we combine these two parameters to construct a multiple linear model. Experimental results have proven that the combined model outperforms that based on individual parameters, and a more accurate estimation can be achieved.

CHAPTER 5

Face Recognition using the Gabor Feature under Pose Variation

5.1 Introduction

Research on face recognition has attracted significant interests in the last 20 years. Various approaches on face recognition have been presented in Chapter 2. In this Chapter, we mainly focus on two types of feature representation for face recognition. The first type is the geometric position of a set of facial feature points, which are automatically extracted by our proposed algorithm, as described in Chapter 3. Since head pose variation can be considered as a geometric problem, we simply use the geometric information to determine the pose angle of an input test face image. The second type is a set of multi-scale, multi-orientation Gabor wavelet coefficients extracted at the respective facial feature points. A Gabor-based face recognition technique with a weighting feature similarity measure is proposed to enhance the face recognition performance when the faces are at different poses. Our approach is illustrated in Figure 5.1.

This chapter is organized as follows. The facial feature representation is introduced in Section 5.2. The weighting feature distance measure based on class discriminability of different poses is proposed and defined in Section 5.3. Section 5.4 presents the experimental results in terms of the recognition rates based on different similarity measures. Finally, a conclusion is given in Section 5.5.

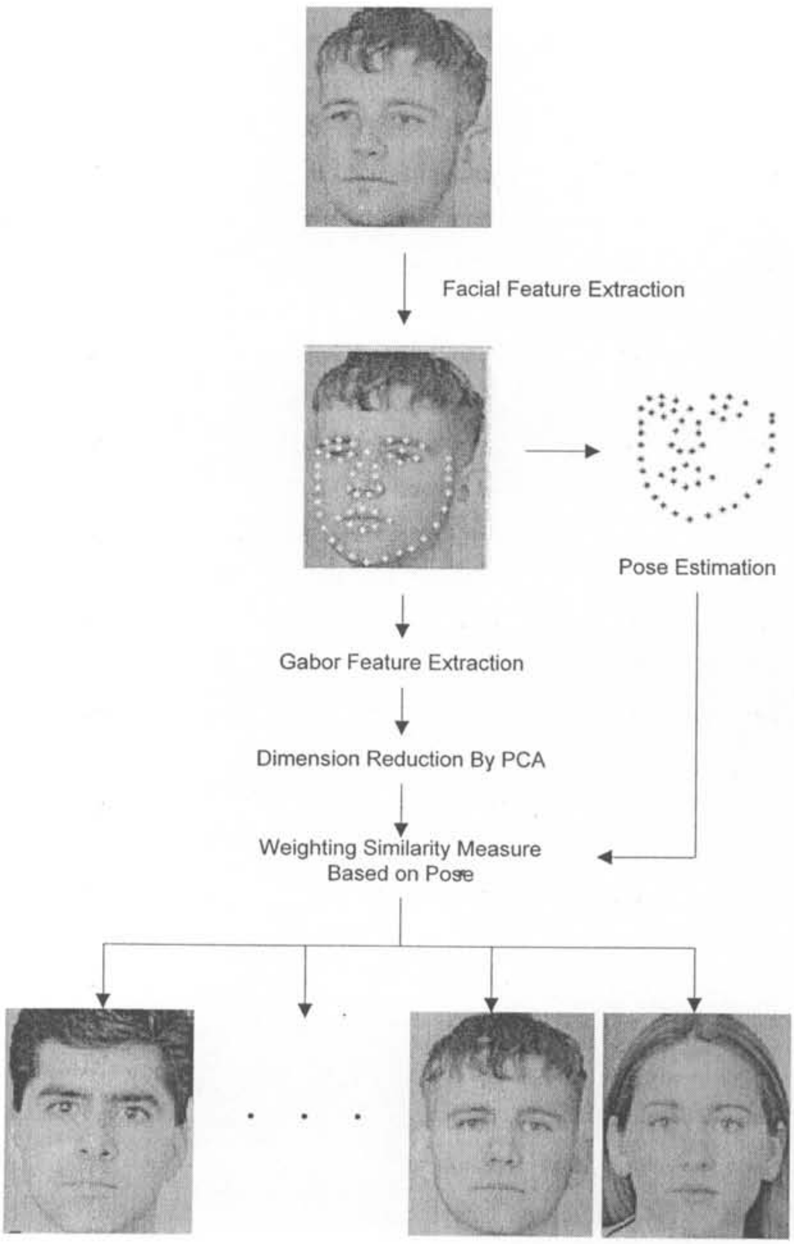


Figure 5.1 System architecture for face recognition with faces under different poses.

5.2 Facial Feature Representation

Extracting useful data from a face image is the main process for a successful face recognition system. These kinds of data are called facial features. A lot of research has been conducted into feature extraction. In general, pixel intensity and geometric information about a face image are the basic facial features. However, face images vary with a change of head position, size, expression, and illumination. Therefore, facial features based on the pixel intensity are insufficient to identify the individual. A more effective form of representation is based on the Gabor wavelets [59, 60], which have been used for texture detection [61] and facial feature extraction [62, 63]. The responses of a Gabor filter have some useful characteristics. First, it provides robustness against varying brightness and facial expression in the image. Second, it can represent the characteristics of the local face region effectively. In other words, the Gabor feature is more effective than using the original face image directly. As Gabor filers of different scales and orientations are employed, the dimension of the Gabor feature is much larger than the original image; therefore principal component analysis (PCA) is applied for dimensional reduction. Finally, this low dimensional feature representation is used in the face classification process.

5.2.1 Gabor Filter Response

The processing of facial images using Gabor filters is based on their biological relevance and computational properties [64-66]. The Gabor filter kernels are similar to the responses of the receptive fields of simple cells in the primary visual cortex. In other words, they are multi-scale and multi-orientation kernels. The response of a Gabor filter describes a small patch of gray values in an image $I(\boldsymbol{x})$ around a given pixel $\boldsymbol{x} = (x, y)$. The responses of the Gabor wavelet to a facial feature point are collectively called a 'Gabor jet', which is obtained by convolution between the image and the Gabor wavelet functions, as shown below:

$$J_j(\boldsymbol{x}) = \int (\boldsymbol{x}')\psi_j(\boldsymbol{x}'-\boldsymbol{x})d^2\boldsymbol{x}' \ , \tag{5.1}$$

where $\psi_j(\boldsymbol{x})$ represents the Gabor kernels as follows:

$$\psi_j(\boldsymbol{x}) = \frac{\boldsymbol{k}_j^{\,2}}{\sigma^2}\exp\left(-\frac{\boldsymbol{k}_j^{\,2}\boldsymbol{x}^2}{2\sigma^2}\right)\left[\exp(i\boldsymbol{k}_j\cdot\boldsymbol{x}) - \exp\left(-\frac{\sigma^2}{2}\right)\right] \ , \tag{5.2}$$

Gabor kernel is a two-dimensional plane wave with wavelet vector $\boldsymbol{k}_j$ restricted by a Gaussian envelope function.

$$\boldsymbol{k}_j = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_v \cos\varphi_\mu \\ k_v \cos\varphi_\mu \end{pmatrix}, \tag{5.3}$$

where $k_v = \frac{0.5\pi}{(\sqrt{2})v}$ and $\varphi_v = \mu\frac{\pi}{8}$.

In our algorithm, we employ 5 different scales, i.e. v = 0, …, 4, and 8 orientations, i.e. μ = 0, …, 7. The corresponding index j for $\boldsymbol{k}_j$ is $j = \mu + 8v$. Thus, there are 40 features in each Gabor jet. The width σ / k of the Gaussian is controlled by the parameter $\sigma = 2\pi$. The second term in the bracket of Eq. (5.2) makes the kernels DC-free. The Gabor kernels and the corresponding Gabor filter responses are shown in Figures 5.2 and 5.3. In our approach, only the magnitudes are used, since they are insensitive to the position while the phases are very sensitive to position.

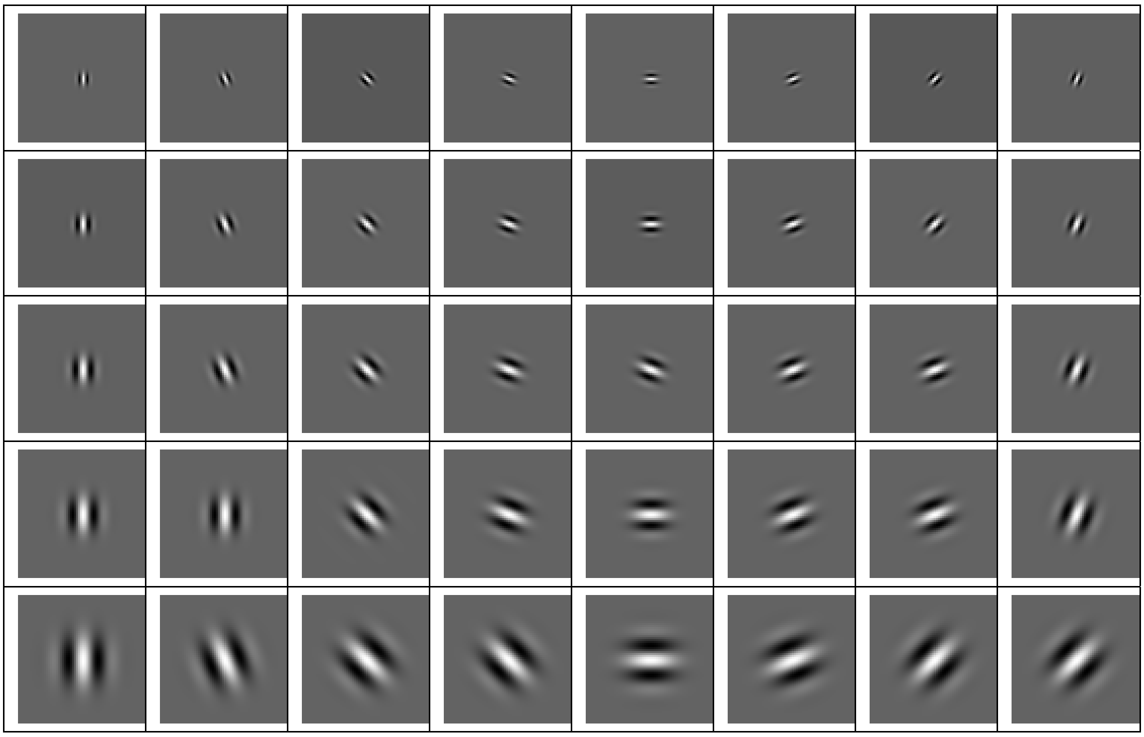
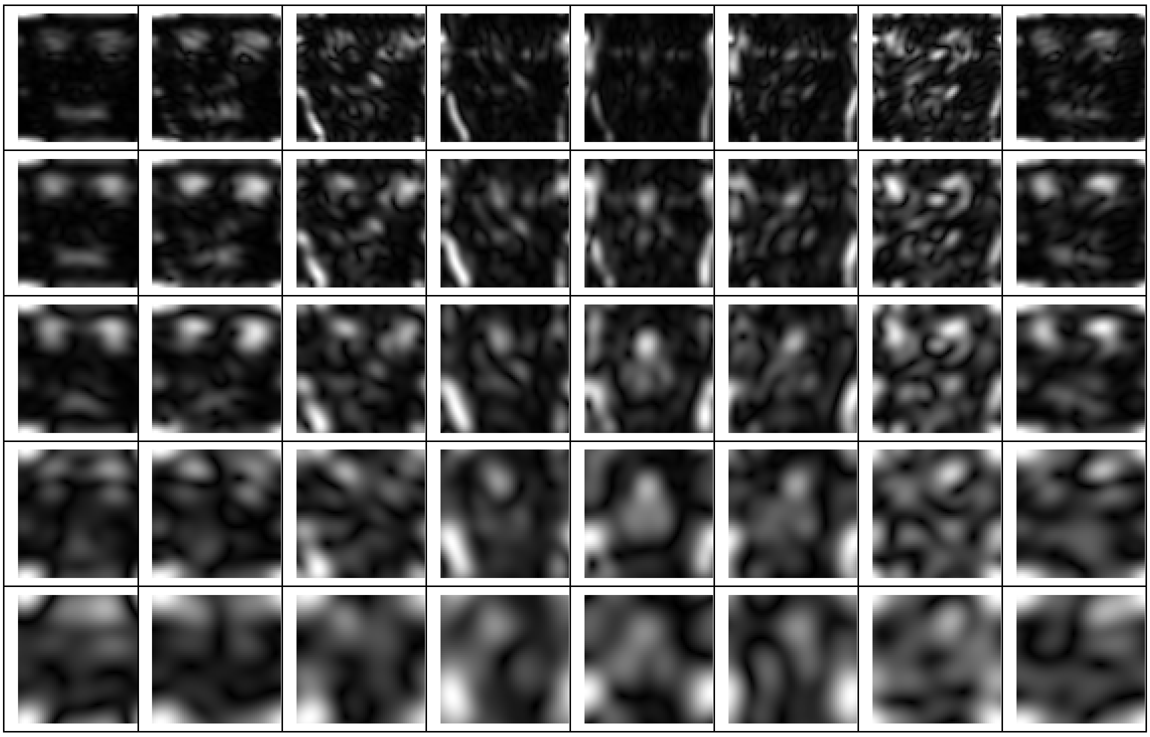


Figure 5.2 The real part of the Gabor kernels with 5 scales and 8 orientations.

(a)



(b)

Figure 5.3 (a) Original face image and (b) the magnitudes of the Gabor filter responses.

5.2.2 Dimensional Reduction of the Gabor Feature

Each image is represented by a Gabor feature vector, which is formed by concatenating the Gabor jets at 52 important facial feature points. These facial feature points are automatically located using our proposed approach, as described in Chapter 3. The Gabor feature vector is represented as

$$\boldsymbol{G} = [\boldsymbol{J}_1, \boldsymbol{J}_2, \ldots \boldsymbol{J}_k]^T , \tag{5.4}$$

$$\boldsymbol{J}_k = [J_{k1}, J_{k1}, \ldots J_{kj}] , \tag{5.5}$$

where k is 52 and j is 40, and where k and j are the number of feature points and the number of Gabor wavelet kernels used, respectively.

In summary, each image is represented by a vector of dimension 2,080 (40×52). After computing the Gabor feature vector, the high dimensional feature vectors are projected onto the principal components. Since the high-dimensional Gabor feature vector requires more bytes in its representation and much more computation in the classification process for face recognition, dimensional reduction should be performed by means of PCA before being utilized for classification and database storage. If N training images is used, the corresponding Gabor feature $\boldsymbol{G}_i$, where i = 1,…, N is computed. The covariance matrix of $\boldsymbol{G}$, i.e. $\boldsymbol{G}\boldsymbol{G}^T$ is of dimension 2,080 × 2,080. To reduce the computation required to solve this large matrix, we consider the covariance matrix $\boldsymbol{G}^T\boldsymbol{G}$ first, which is of dimension N×N only. If the eigenvectors of $\boldsymbol{G}^T\boldsymbol{G}$ are denoted as $\boldsymbol{u}_i$, then the corresponding eigenvectors of $\boldsymbol{GG}^T$ are $\boldsymbol{Gu_i}$. Let P be the projection matrix whose column vectors are the leading eigenvectors of the covariance matrix. The number of eigenvectors used is much smaller than the dimension 2080. Hence, the Gabor feature vector of a face image can be represented in the reduced subspace as follows:

$$\boldsymbol{x} = \boldsymbol{P}(\boldsymbol{G}_i - \overline{\boldsymbol{G}}), \qquad (5.6)$$

where $\mathbf{G}_i$ is the input Gabor feature vector and $\overline{\boldsymbol{G}}$ is the mean of the Gabor feature vectors in the training set. This dimensional reduction is important as it can reduce the computations required when face recognition is performed.

5.3 The Weighting Feature Similarity Measure Based on Poses

PCA can provide the best representation of data in a low-dimensional space, but this does not necessarily imply the best representation results in the best discrimination of the data. In this section, we propose a weighting function for different similarity measures, which can emphasize those with greater discrimination power. Suppose that each of the feature parameters extracted from the PCA has a different degree of discriminating characteristic for the different pose variations. Hence, a weighting function based on the pose angle is defined according to the class discriminability of each of the feature parameters, as follows:

$$\beta = \frac{\sigma_B}{\sigma_W}, \tag{5.7}$$

where $\sigma_B = \sum_{i=1}^{C} n_i \left(m_i - m\right)^2$

and $\sigma_W = \sum_{i=1}^{C} \sum_{x_j \in X_i} \left(x_j - m_i\right)^2$,

where σ_B and σ_W are the between-class and within-class distance values, respectively, X_i represents feature vectors belonging to the *i*th class, m_i and m denote the mean of the *i*th class and the mean of all the training samples, and n_i is the number of samples in the *i*th class. In order to measure class discriminability on different poses with respect to the frontal view, each class contains two samples only — one frontal view and one side view with a particular pose. To obtain a better discrimination of a feature parameter for a particular pose, the larger weighting factor for that feature parameter is used. The feature parameters with the best discrimination are chosen to maximize the ratio of inter-person variance to intra-person variance. Therefore the distance between vectors corresponding to images of different people should be large compared to the distance between vectors corresponding to images of the same person.

In our approach, the weighting value is determined by the experiment, which is set proportional to the discriminability value. Following is a definition of the weighting feature similarity measure:

Given two face images $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$, which are represented by Gabor feature vectors $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$, respectively. The corresponding feature vectors after performing PCA are denoted as $\boldsymbol{x}_1$ and $\boldsymbol{x}_2 \in \mathbb{R}^M$, respectively. The similarity measure of these two vectors is:

$$Sim_{tot}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_{k=1}^{M} \beta_k^{\theta} Sim_k(x_{1k}, x_{2k}), \tag{5.8}$$

where β_k^{θ} is the kth weighting function based on the pose θ with the restrictions that $\beta_k^{\theta} > 0$ *and* $\sum_k \beta_k^{\theta} = 1$, and M is the dimension of the reduced Gabor feature vector. $Sim_k(x_{1k}, x_{2k})$ can be of any similarity measure between the kth feature of x_1

and x_2. If the feature k is more useful for face recognition with faces at a pose θ, we expect β_k^θ to be high, and vice versa.

In order to evaluate the efficiency of our proposed similarity measure, we compare its performance to the L_1, L_2 and cosine similarity measures based on the nearest neighbor classification rule for face recognition. The nearest neighbor classification rule is defined as follows:

$$Sim(\boldsymbol{x}, \boldsymbol{x}_k) = \min_j Sim(\boldsymbol{x}, \boldsymbol{x}_j) \tag{5.9}$$

The feature vector $\boldsymbol{x}$ is classified to the closest class k based on the similarity measure. The similarity measures compared in our experiments include the L_1 distance, L_2 distance, cosine similarity measure, weighted L_1 distance, weighted L_2 distance, and weighted cosine similarity measure, which are defined as follows:

$$Sim_{L_1} = \sum_{k=1}^{M} \left| x_{1k} - x_{2k} \right| \tag{5.10}$$

$$Sim_{L_2} = \left(\boldsymbol{x}_1 - \boldsymbol{x}_2 \right)^t \left(\boldsymbol{x}_1 - \boldsymbol{x}_2 \right) \tag{5.11}$$

$$Sim_{COS} = \frac{-\boldsymbol{x}_1^t \boldsymbol{x}_2}{\left\| \boldsymbol{x}_1 \right\| \left\| \boldsymbol{x}_2 \right\|} \tag{5.12}$$

$$Sim_{WL_1} = \sum_{k=1}^{M} \beta_k^\theta \left| x_{1k} - x_{2k} \right| \tag{5.13}$$

$$Sim_{WL_2} = \sum_{k=1}^{M} \beta_k^\theta \left(x_{1k} - x_{2k} \right)^2 \tag{5.14}$$

$$Sim_{WCOS} = -\frac{1}{\|\boldsymbol{x}_1\|\|\boldsymbol{x}_2\|}\sum_{k=1}^{M}\beta_k^{\theta}x_{1k}x_{2k} \tag{5.15}$$

where $\|\bullet\|$ denotes the norm operator. Note that the cosine similarity measure includes a minus sign in (5.12) because the nearest neighbor rule applies minimum distance measure rather than maximum similarity measure.

5.4 Experimental Results

To evaluate the performance of our face recognition algorithm based on the different weighting feature similarity measures, the FERET face database [67] was used in the experiment. It contains 200 subjects, and each subject has a series of images with different poses. In our experiments, 100 subjects are used as the training samples in the principal component analysis. The remaining of 100 subjects from the database were selected to evaluate the recognition performance. Each of the subjects has one upright frontal view and six different poses. The size of each image is 256 × 384, and some examples are shown in Figure 5.5.

The experiment set-up was as follows: the upright frontal view of each subject was chosen to form our database. The remaining 6 images of different poses were used as testing images, so we have a total of 600 testing images. Each of the testing images was then compared to each face in the database. The 52 facial feature points in each of the face images are located manually. To construct the subspace for dimensional reduction, 50 eigenvectors, which represent 98% of the shape variations, are selected from the projection matrix. For comparison purposes, the different normal similarity measures and our proposed weighted similarity measures were evaluated in the experiments. These experiments were performed in the dimensionality reduced subspace by PCA. Figure 5.4 shows the face recognition performance by using the Sim_{L_1} (L1), Sim_{WL_1} (WL1), Sim_{L_2} (L2), Sim_{WL_2} (WL2), Sim_{COS} (Cos), and Sim_{WCOS} (WCos) similarity measures. The results show that face recognition rates based on the three normal similarity measures improve when our proposed weighting function is applied. Table 5.1 tabulates the best recognition rate of each similarity measure, and the corresponding results under different pose angles are listed in Table 5.2. From these two tables, we can find that (1) the performance of our proposed approach is better than that of the normal similarity measures, and (2) our approach achieves 100% recognition accuracy when the pose angle is within 15° of the left or right and 50 features are used.

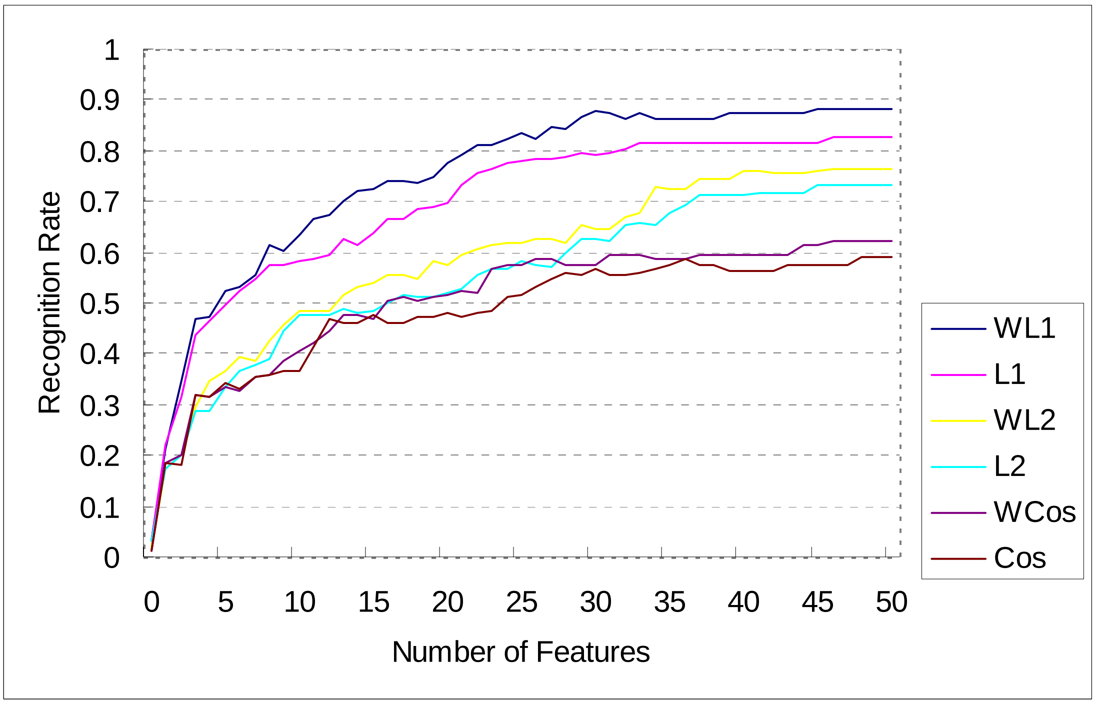


Figure 5.4 Face recognition rate using L1, L2 and cosine similarity measures.

	Normal	Weight
L_1 distance	81%	88%
L_2 distance	73%	76%
Cosine distance	59%	62%

Table 5.1 The best results of each similarity measure.

Pose angle	-40˚	-25˚	-15˚	15˚	25˚	40˚
L_1 distance	0.67	0.83	0.92	0.94	0.79	0.68
Weighted L_1 distance	0.74	0.92	1.00	1.00	0.89	0.76

Table 5.2 Recognition rates using 50 features under different pose angles.

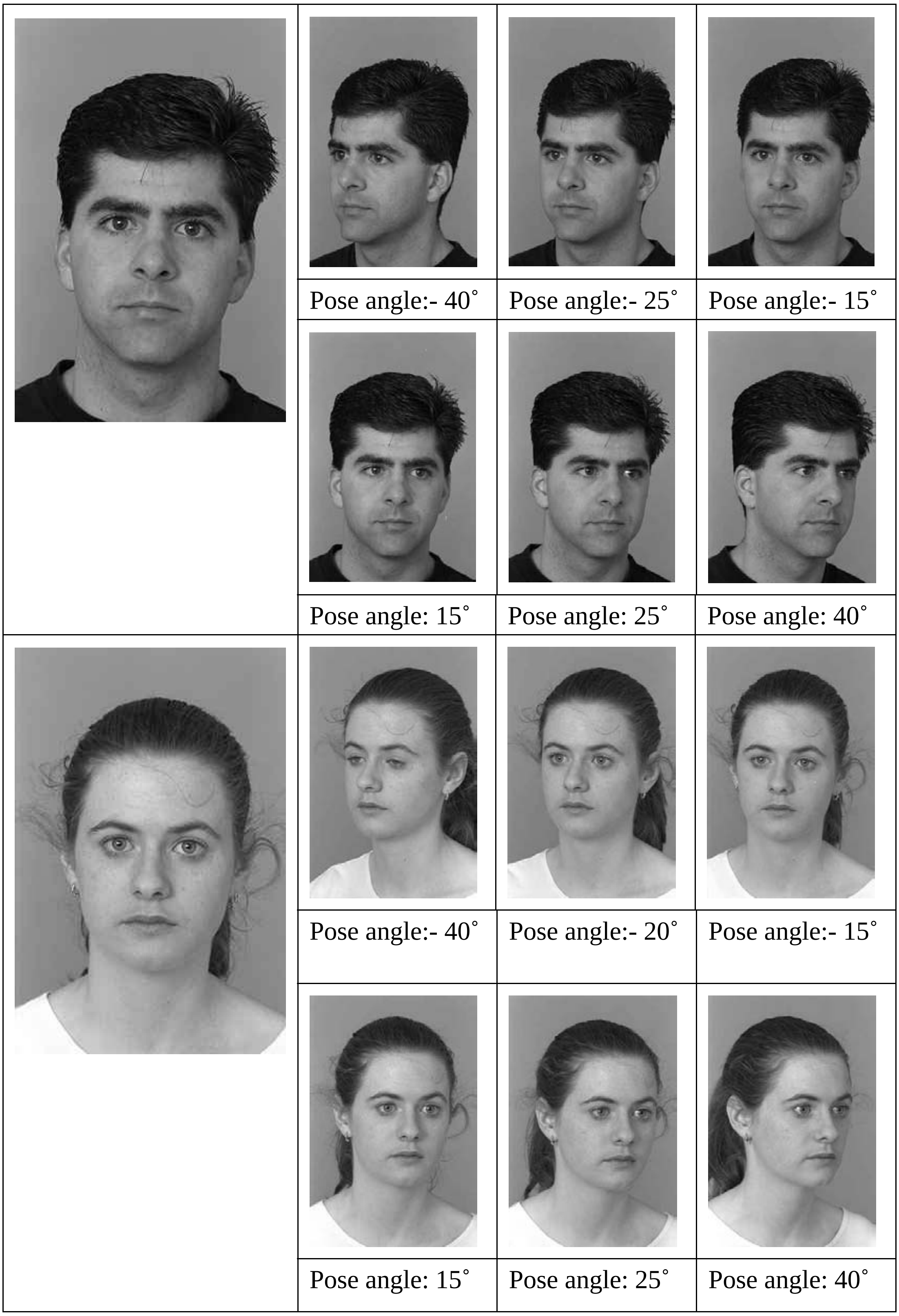


Figure 5.5 Some examples of the images in the FERET face database.

5.5 Conclusion

In this chapter, a new weighting feature similarity measure based on pose estimation is proposed for face recognition. The discrimination power of each feature parameter for a range of pose angles is different. Therefore, it is necessary to conduct a quantitative evolution in different poses. Our new weighting function can emphasize those feature parameters with a higher discriminate power for a particular pose. To achieve a high recognition performance level, Gabor features are extracted at 52 predefined facial feature points instead of a whole face region. This can help to alleviate the effect of variations in both illumination and small facial expression. In order to classify the faces more effectively, the high dimensional Gabor feature vector is reduced by PCA. Experimental results based on the FERET database show that this new approach outperforms the normal similarity measures in terms of recognition rate. In particular, the weighted L_1 similarity measure achieves 100% recognition accuracy under $\pm 15^\circ$ pose angles when 50 features are used.

CHAPTER 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we have provided an overview of a human face recognition system and introduced some existing techniques for facial feature extraction and face recognition. For facial feature extraction, the active contour model (snake), deformable template and active shape model (ASM) have been reviewed. The active contour model and the deformable template have been widely adopted; they can achieve a good performance level in facial feature extraction, such as the mouth, the eyes and the face contour. However, these methods are computationally intensive due to the fact that a large number of parameters are involved during the optimization process. In the active shape model, points are used to describe the details of a face shape object and are controlled by a several main modes of shape variation derived from a training data set. The major advantage of using ASM is that no heuristic assumptions are made as to the legal shape. However, the ASM can not be able to provide a good fit to those face shapes that are quite different from the training data set.

For face recognition, three approaches, namely the holistic-based, feature-based and hybrid approaches, have also been presented. The holistic-based approach considers the global properties of a human face as the raw input to the recognition system. The performance of this approach will be degraded if the face to be recognized is not aligned well. In the feature-based approach, local features such as the eyes, nose and mouth are first extracted, and their locations (geometric) and local statistics (appearance) are fed into the classification system. Its success relies mainly on the accuracy of the facial feature extraction. In order to suppress the weaknesses of those two approaches, a combination of different techniques based on the global and local features of a human face is employed in the hybrid approach. As the hybrid approach combines all the strengths of the different techniques, it should potentially offer a better performance than other the two approaches work independently.

In our research, we have proposed efficient methods for facial feature extraction and human face recognition. To extract the location of facial feature such as the eyes, nose, mouth, and face contour under perspective variations, a more accurate approach based on the genetic algorithm and active shape model has been proposed. In order to make the model represent a face more flexibly, the representations of the important facial features, i.e. the eyes, nose and mouth, and the face contour are separated. An energy function is defined that links these two representations of a human face. To represent a face image under different poses, three models are employed to represent the important facial features: the left-viewed, right-viewed and frontal-viewed models. In additional, the genetic algorithm is applied to search for the best representation of face images. Experimental results show that the facial features can be extracted more reliably and accurately under different perspective variations.

In this research, pose estimation based on the shape model parameters has also been investigated to determine the approximate pose angles of an input face image. We have improved the performance of pose estimation by investigating the relationships between the first few shape model parameters. We have also modeled the relations between two of the shape parameters by the multiple linear regression model. Experimental results show that this approach can provide a better performance level than using the first shape parameter for pose estimation.

For face recognition, we have employed the Gabor filters to extract the facial features at the predefined feature points, instead of directly using pixel gray values and applying the PCA to reduce the dimension of the Gabor feature vectors. Since the discrimination power of each feature parameter for a range of pose angles is different, we conducted a quantitative discriminability evaluation for different poses. Hence, a new weighting function that can emphasize the significance of the feature parameters in a particular pose is proposed. We have conducted different similarity measures for comparison purposes. Experimental results show that this new weighting similarity measure can achieve a higher recognition rate than the normal similarity measure.

In practice, our proposed facial feature extraction approach can be used as the first stage of an automatic face recognition system. Then, the pose estimation is the second stage, which estimates the pose based on the extracted facial feature points. Finally, with the estimated pose angle, we can select a set of appropriate weights on the similarity measure for face classification. This overall system architecture is presented in Figure 5.1. Since we cannot guarantee that the testing face image is always in an upright frontal view, our proposed automatic face recognition system can reduce the effect of variations in the pose angle, so only a single frontal view image of a person needs to be stored in a database.

In our research, the facial feature extraction process is more computationally intensive than the face recognition process because the genetic algorithm is employed, which is computational in the encoding/decoding of the chromosomes and measuring the fitness of candidate in application. Nevertheless, this computational process has to be performed once for a query input, and the runtime is in the order of 2 to 3 seconds. Although the time required to capture two faces is very short, it will take much longer if the database concerned contains thousands of faces. As a consequence, the time for facial feature detection will become relatively short. In addition, the accuracy of the facial feature detection will directly affect the accuracy of the face recognition. To create a real-time face recognition system, parallel processing or dedicate hardware should be employed to process the genetic algorithms. The encoding/decoding process for each chromosome and he fitness evaluations for each candidate solution can be calculated independently. This means that all the candidates can be computed in parallel.

6.2 Future Work

In our research, face detection and facial feature extraction are the first step to be performed for face recognition. As we know, a robust and successful face detection step will increase the recognition performance. Implementing such a face detection method is an important future task for successful face recognition applications. Moreover, only local facial features are considered in our proposed face recognition algorithm. In order to achieve a more robust face recognition system, we can also employ the global texture features. In general, the global texture features of a face image vary much more than the local features due to changes in head pose, facial expression and environment illumination. One possible way to overcome this problem is to use shape free features. The shape-free feature is a normalized shape-free gray-level patch that is enclosed by the mean face shape warped from the original images. The global texture image has the same edges and contours and is insensitive to shape variation due to the differences in head pose and facial expressions. Therefore, our feature work should consider the combination of the global texture and local features in face recognition system.

References

[1] G. Ann, "Authentication: From password to biometrics", www.sis.uncc.edu/~gahn, University of North Carolina, December 2000.

[2] M. Turk and A. Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, pp. 711–720, 1997.

[4] A. Martinez, "Recognizing imprecisely localized, partially occluded and expression variant faces from a single sample per class", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 748–763, 2002.

[5] M. H. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: A survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 34–58, 2002.

[6] S. H. Jeng, H. Y. M. Liao, C. C. Han, M. Y. Chern and Y. T. Liu, "Facial Feature Detection using Geometrical Face Model: an Efficient Approach", Pattern Recognition, vol. 31, no. 3, pp. 273-282, 1998.

[7] W. Haiyuan, C. Qian and M. Yachida, “Facial Feature Extraction and Face Verification”, Proc. of the 13th International Conference on Pattern Recognition, vol. 3, pp. 484-488, 1996.

[8] F. C. Wu, T. J. Yang and M. Ouhyoung, “Automatic Feature Extraction and Face Synthesis in Facial Image Coding”, Sixth Pacific Conference on Computer Graphics and Applications, pp. 218-219, 1998.

[9] A. M. Alattar and S.A. Rajala, “Facial Features Localization in Front View Head-and -Shoulders Images”, Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 6, pp. 3557-3560, 1999.

[10] G. B. De Natale, D. Giusto and F. Maccioni, “A Symmetry-Based Approach to Facial Features Extraction”, Proc. of the 13th International Conference on Digital Signal Processing, vol. 2, pp. 521-525, 1997.

[11] F. Mu, H. Li and R. Forchheimer, “Automatic Extraction of Human Facial Features”, Signal Processing: Image Communication, vol. 8, no. 4, pp. 309-326, May 1996.

[12] D. Pramadihanto, Y. Iwai and M. Yachida, “A Flexible Feature Matching for Automatic Face and Facial Feature Points Detection”, Proc. of the 14th International Conference on Pattern Recognition, vol. 1, pp. 92-95, 1998.

[13] S. J. McKenna, S. Gong, R. P. Wurtz, J. Tanner and D. Banin, "Tracking Facial Feature Points with Gabor Wavelet and Shape Models", Proc. of the 1st International Conference on Audio- and Video-based Biometric Person Authentication, vol. 1206, pp. 35-42, 1997.

[14] A. L. Yuille, "Deformable templates for face recognition", Journal of Cognitive Neuroscience. vol. 3 no. 1, pp. 59-70, 1991.

[15] K. M. Lam and Hong Yan, "An Improved Method for Locating and Extracting the Eye in Human Face Images", Proc. of the 13th International Conference on Pattern Recognition, vol. 3, pp. 411-415, 1996.

[16] K. M. Lam and Hong Yan, "Locating and Extracting the Eye in Human Face Images", Pattern Recognition, vol.29, no. 5, pp.771-779, 1996.

[17] A. R. Mirhosseini, H. Yan and K.M. Lam, "Adaptive Deformable Model for Mouth Boundary Detection", Optical Engineering, vol. 3, no. 27, pp. 869-875, March 1998.

[18] A. R. Mirhosseini, C. Chen, K. M. Lam and H. Yan, "A Hierarchical and Adaptive Deformable Model for Mouth Boundary Detection", Proc. International Conference on Image Processing, vol. 2, pp. 756-759, 1997.

[19] S. Bernogger, L. Yin, A. Basu and A.Pinz, “Eye Tracking and Animation for MPEG-4 Coding”, Proc. of the 14th International Conference on Pattern Recognition, vol. 2, pp. 1281-1284, 1998.

[20] G. C. Feng and P. C. Yuen, “Variance Projection Function and its application to Eye Detection for Human Face Recognition”, Pattern Recognition Letters, vol. 19, no. 9, pp. 899-906, July 1998.

[21] T. Hamada, K. Kato and K. Kawakami, “Extracting Facial Features as in infants”, Pattern Recognition Letters, vol. 21, pp. 407-412, 2000.

[22] M. Kass, A. Witkin and D. Terzopoulos, “Snakes: Active Contour Models”, International Journal of Computer Vision, vol. 9, pp. 321-331, 1988.

[23] K. M. Lam and H. Yan, “Fast Algorithm for Locating Head Boundaries”, Journal of Electronic Imaging, vol. 3, no. 4, pp.351-359, Oct. 1994.

[24] K. M. Lam. and H. Yan, “Fast Greedy Algorithm for Active Contours”, Electronic Letter, vol.30, no. 5, pp.21-22, Jan. 1994.

[25] W. P. Choi, K. M. Lam and W. C. Siu, “An Adaptive Active Contour Model for Highly Irregular Boundaries”, Pattern Recognition, vol. 34, no. 2, pp. 323-33, 2001.

[26] T. F. Cootes, C. J. Taylor, “Active Shape Model”, Third British Machine Vision Conference, pp. 266-275, 1992.

[27] T.F. Cootes, A. Hill, C. J. Taylor, J. Haslam, “The Use of Active Shape Models for Locating Structures in Medical Images”, Image and Vision Computing, vol. 12 no. 6, pp.355-366, 1994.

[28] A. Lanitis, C. J. Taylor, T. F. Cootes, A. Ahmad, “Automatic Interpretation of Human Faces and Hand Gestures using Flexible Models”, Image and Vision Computing, vol. 13, no. 5, pp. 98-103, 1995.

[29] T. F. Cootes, C. J. Taylor, A. Lanitis, D. H. Cooper and J. Graham, “Building and Using Flexible Models Incorporating Grey-Level Information”, Fourth International Conference on Computer Vision, pp. 242-246, 1993.

[30] S. C. Yan, C. Liu, S. Z. Li, H. J. Zhang, H. Shum, Q. S. Cheng, “Face alignment using texture-constrained active shape models”, Image and Vision Computing, vol. 21, no. 1, pp. 69-75, 2003.

[31] B. Froba, T. Kastner, W. Zink, C. Kublbeck, “Real-time active shape models for face segmentation”, International Conference on Image Processing, pp. 205-208, 2001.

[32] L. Yong, C. S. Zhang, L. Xiaoguang, D. Zhang, “Face contour extraction with active shape models embedded knowledge”, International Conference on Signal Processing, vol. 2, pp. 1347-1350, 2000.

[33] B. Ginneken, A. Frangi, J. Staal, B. Romeny and M. Viergever, “A nonlinear gray-level appearance model improves active shape model segmentation”, Proceedings of the Mathematical Methods in Biomedical Image Analysis, pp. 205–212, 2001.

[34] S. Romdhani, S. Gong and A. Psarrou, “A Multi-View Nonlinear Active Shape Model Using Kernel PCA”, 10th British Machine Vision Conference, vol. 2, pp. 483-492, 1999.

[35] R. Chellappa, C. L. Wilson and S. Sirohey, “Human and Machine Recognition of Faces: A Survey”, Proc. of the IEEE, vol. 83, no. 5, pp. 705-740, May. 1995.

[36] M. A. Grudin, “On Internal Representations in Face Recognition systems”, Pattern Recognition, vol. 33, no. 7, pp. 1161-1177, 2000.

[37] T. Kanade, “Computer recognition of human faces”, Interdisciplinary Systems Research, vol. 47, 1977.

[38] M. D. Kelly, “Visual identification of people by computer”, Technical Report. AI-130, Stanford AI Project, Stanford, CA, 1970.

[39] F. Samaria, “Face recognition using hidden markov models”, Ph.D. dissertation. University of Cambridge, Cambridge, U.K. 1994.

[40] F. Samaria and S. Young, “HMM based architecture for face identification”, Image and Vision Computing, vol. 12, no.8, pp. 537–583, 1994.

[41] L. Wiskott, J. M. Fellous, and C. Von Der Malsburg, “Face recognition by elastic bunch graph matching”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 775–779, July 1997.

[42] J. Buhmann, M. Lades, and C. Von Der Malsburg, “Size and distortion invariant object recognition by hierarchical graph matching”, In Proceedings, International Joint Conference on Neural Networks, vol. 2, pp. 411–416, June 1990.

[43] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. Von Der Malsburg, R. Wurtz, and W. Konen, “Distortion invariant object recognition in the dynamic link architecture”, IEEE Transactions on Computers, vol. 42, no. 3. pp. 300–311, March 1993.

[44] A. Pentland, B. Moghaddam, and T. Starner, “View-based and modular eigenspaces for face recognition”, In Proceedings IEEE Conference on Computer Vision and Pattern Recognition, 1994.

[45] G. Gordon, “Face Recognition from Frontal and Profile Views”, In Proceedings of the International Workshop on Face and Gesture Recognition, pp.47-52, June 1995.

[46] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1989.

[47] C. H. Lin and J. L. Wu, "Genetic block matching algorithm for video coding", Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems, pp. 544-547, June 1996.

[48] D. L. Swets, B. Punch and J. Weng, "Genetic algorithms for object recognition in a complex scene", Proceedings of the International Conference on Image Processing, vol. 2, pp. 595-598, 1995.

[49] K. W. Wong, K. M. Lam and W. C. Siu, "An efficient algorithm for human face detection and facial feature extraction under different conditions", Pattern Recognition, vol. 34, pp. 1993-2004, 2001.

[50] Y. Yokoo and M. Hagiwara, "Human faces detection method using genetic algorithm", Proceedings of IEEE International Conference on Evolutionary Computation, pp. 113-118, May 1996.

[51] C. H. Lin and J. L. Wu, "Automatic facial feature extraction by genetic algorithms", IEEE Trans, IEEE Transactions on Image Processing vol. 8, no. 6, pp. 834-845, June 1999.

[52] A. Lanitis, A. Hill, T. F. Cootes and C. J. Taylor, "Locating Facial Features using Genetic Algorithms", Proc. International Conference on Digital Signal Processing, pp. 520-525, 1995.

[53] S. Z. Li, X. G. Lv and H. J. Zhang, “View-Based Clustering of Object Appearances Based on Independent Subspace Analysis”, Proc. Eighth IEEE International Conference on Computer Vision, vol. 2pp. 295-300, July 9-12, 2001.

[54] Y. Li, S. Gong and H. Liddell, “Support Vector Regression and Classification Based Multi-View Face Detection and Recognition”, Proc. Fourth IEEE International Conference on Face and Gesture Recognition, pp.300-305, March 2000.

[55] Y. Li, S. Gong and H. Liddell, “Multi-view face detection using support vector machines and eigenspace modeling”. In Proceedings of International Conference on Knowledge-based Intelligent Engineering Systems and Allied Technologies, pp. 241-245, September 2000.

[56] T. F. Cootes, K. Walker and C. J. Taylor, “View-Based Active Appearance Models”, Proc. Fourth International Conference on Automatic Face and Gesture Recognition, pp. 227-232, March 2000.

[57] A. Lanitis, C. J. Taylor and T. F. Cootes, "Automatic Interpretation and Coding of Face Images Using Flexible Models”. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.19, no.7, pp.743-756, July 1997.

[58] S. Z. Li, Q. D. Fu, Li. B. Scholkopf, Y. M. Cheung and H. J. Zhang, “Kernel Machine Based Learning for Multi-View Face Detection and Pose Estimation”, In

Proceedings of 8^{th} IEEE International Conference on Computer Vision, ICCV2001, Vancouver, Canada, vol. 2, pp. 674-679, July 9-12, 2001.

[59] C. K. Chui, An Introduction to Wavelets. Academic Press. 19992.

[60] T. S. Lee, "Image representation using 2D Gabor wavelets", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 10, pp. 959-971, 1996.

[61] B. S. Manjunath, W. Y. Ma,, "Texture feature for browsing and retrieval of image data", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 8, pp. 837-842, 1996.

[62] Lades. M., Vorbruggen, J.C. Buhmann, J., Lange, J., Malsburg, C.V.D., Wurtz, R.P., Konen, W., "Distortion invariant object recognition in the dynamic link architechure". IEEE Trans. Comput, vol. 42, no.3, pp. 300-311, 1993.

[63] C. J. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition image processing", IEEE Transactions on Image Processing, vol. 11, no. 4, pp. 467-476, April 2002.

[64] J.G. Daugman, "Uncertainty for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-Dimensional Cortical Filter", Journal of the Optical Society America, vol. 2, no. 7, pp. 1160-1169, July 1998.

[65] J.G. Daugman, “Complete Discrete 2D Gabor Transforms by Neural Networks for Image Analysis and Machine Intelligence”, vol. 36, no. 7, pp. 1169-1179, July 1988.

[66] J. P. Jones and L. A. Palmer, “An Evaluation of the Two-Dimensional Gabor Filter Model of Simple Receptive Fields in Cat Striate Cortex”, In Journal of Neurophysiology, vol. 58, no. 6, pp. 1212-1232, December 1987.

[67] P. J. Philips, H. Wechsler, J. Huang, and P. Rauss, “The FERET Database and Evaluation Procedure for Face-Recognition Algorithms”, Image and Vision Computing, vol. 16, pp. 295-306, 1998.