THE HONG KONG POLYTECHNIC UNIVERSITY

DEPARTMENT OF

ELECTRONIC AND INFORMATION ENGINEERING

Efficient Schemes for Indexing and Retrieval from Large Face Databases

(A thesis submitted in partial fulfillment of the requirements for the Degree of Master of Philosophy)

Student Name:	**KOO Hei Sheung**
Student ID:	**04900877R**
Supervisor:	**Dr. Kenneth K.M. Lam**
Initial Submittion Date:	**25 January 2007**

CERTIFICATE OF ORIGINALITY

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

_______________ (Signed)

Koo Hei Sheung _______ (Name of student)

Abstract

The aim of this research is to develop efficient techniques for face recognition with a large face database. In practice, the number of faces in a database may range from hundreds to several tens of thousands. As a result, many problems need to be considered when developing a practical human face recognition system. One of these problems is the required search time for a human face in the large database. To reduce the search time, an efficient indexing and retrieval algorithm is required. In this thesis, efficient and accurate face recognition techniques based on a 3-D face structure and the optimal selection of Gabor features will be investigated for a large human face database.

In this thesis, a new algorithm is proposed to derive the 3-D structure of a human face from a group of face images under different poses. Based on the corresponding 2-D feature points of the respective images, their respective poses and the depths of the feature points can be estimated based on measurements using the similarity transform. To accurately estimate the pose of and the 3-D information about a human face, the genetic algorithm (GA) is applied. Our algorithm does not require any prior knowledge of camera calibration, and has no limitation on the possible poses or the scale of the face images. It also provides a means to evaluate the accuracy of the constructed 3-D face model based on the similarity transform of the 2-D feature point sets. We have shown that our approach can be applied to face recognition such that the effect of pose variations can be alleviated. Experimental results show that our proposed algorithm can construct a 3-D face structure reliably and efficiently.

Another approach to enhance the performance of face recognition in a large face database is the use of selective Gabor features. Gabor features have a good performance level for face recognition. However, the extraction of Gabor features at different centre frequencies and orientations is computationally intensive. An algorithm to extract and select the Gabor features for face recognition has been proposed. The Gabor features of the images are extracted using a simplified version of the Gabor wavelet; this can reduce the extraction runtime by 30% compared to using original Gabor wavelets. As the responses of the Gabor wavelets are strongly related to those edges that are perpendicular to the wave vectors, edge detectors with different orientations are employed. To further reduce the recognition runtime, the size of the database can be decreased so that the number of comparisons between the query image and each model image can be reduced. Experimental results show that the recognition rate can be maintained with a faster processing time.

Author's Publications

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

Journal Paper (Revised and Submitted)

1. H. S. Koo and K. M. Lam, “Recovering the 3-D Shape and Poses of face images based on the Similarity Transform,” submitted to the *Pattern Recognition Letter*.

Conference Papers (Accepted)

1. H. S. Koo and K. M. Lam, “An Efficient Scheme for 3-D Face Construction based on the Similarity Transformation,” *International Workshop on Advanced Image Technology*, Thailand, 2007. (you should also give the page numbers.)

Acknowledgements

I would like to take this opportunity to express my sincere gratitude to my Chief Supervisor, Dr. Kenneth K. M. Lam, from the Department of Electronic and Information Engineering of the Hong Kong Polytechnic University. Without his support, this research work would not have been completed. He offered me many valuable ideas and professional advice for my research work as well as my future career.

I would also like to thank all members of the DSP Research Laboratory, especially for Professor W. C. Siu, K. W. Wong, Thomas Tse, W. P. Choi, Billy Chow, C. M. Lai, Danny Sze, X. D. Xie, C. Cai, H. Fu and many more. Their supportive and contributive comments that help me overcome a lot of difficulties during my research studies. The countless discussions I had with them have proved to be both fruitful and inspiring. I am also grateful to the other colleagues: M. C. Cheung, S. R. Chen, K. C. Tam, K. C. Liu, K. H. Chun, K. O. Cheng and C. P. Wu for making my life so much simpler and more delightful.

I owe the most to my family for their love and support. Without their patience and forbearance, my research work would have not been completed.

Finally, I am also thankful to the Centre for Multimedia Signal Processing of the Department of Electronic and Information Engineering for generous support over the past two years.

Table of Contents

List of Figures

List of Tables

Chapter 1：Introduction

The objective of this chapter is to introduce the issues to be considered for face recognition with a large human face database. In this research, the structures of 3-D face models, the selection of Gabor features and an indexing scheme will be investigated. Therefore, a general concept of human face modeling, the Gabor features which are used for face recognition will first be introduced. We will also address the originality and the organization of this thesis.

1.1 Problem Statement

Some work was done on facial profile-based biometrics in 1888 [27]. However, the earliest work on face recognition can be traced back at least to the 1950s in psychology [34] and to the 1960s in the engineering literature [109]. In the 1970s, research on automatic machine recognition of faces was started [88], [64] and [100]. However, research on face recognition technology has grown significantly since the early 1990s. There were several reasons for this, including an increase in emphasis on civilian/commercial research projects, the availability of real-time hardware, and the increasing need for surveillance-related applications, etc. Nowadays, an automated face recognition system that performs the functions of face detection, verification and recognition has a wide rage of applications, such as identity verification, access control, face-based video indexing/browsing, security surveillance, and human-computer interaction [88], [97], [2].

In practice, the size of a face database used may range from hundreds to several tens of thousands. Many problems have therefore to be considered when developing a practical human face recognition system. This thesis focuses on indexing and retrieval from a large human face database. An efficient indexing structure can greatly reduce the search time for a human face in the large database. The efficient and accurate face recognition techniques in the large human face database investigated in this thesis are based on the 3-D face structure and the optimal selection of Gabor features. However, the generation of a realistic and accuracy 3-D human face model is one of the most challenging and difficult problems in computer graphics. On the other hand, the Gabor features have a good performance level for face recognition, but extracting Gabor features at different centre frequencies and orientations is computationally intensive. For face recognition with large face databases, the number of Gabor wavelets (GWs) to be used should be as few as possible so as to reduce the feature dimensionality while maintains their discriminant power.

1.2 Motivation

1.2.1 3-D Face Model Reconstruction

Many face recognition methods have been developed over the past few decades. Most of those based on frontal-view images without expression and under controlled lighting can achieve a reasonably high performance level [88], [113], [110], [94]. However, face recognition techniques based on 2-D images are strongly affected by the variation in pose, which are the primary source of difficulties in face recognition. The performance of face recognition algorithms suffers dramatically when a large variation in pose is present in a query image, especially when training data have few non-frontal images. A sensible way to improve the recognition performance for the face images under arbitrary poses is to use multiple training images under different poses. However, face images under different poses may not be available in some applications, and the use of multiple face images will greatly increase the size of a database and the computation required for matching. Various 3-D deformable models [74], [1], [22] have therefore been applied for pose-invariant face recognition.

An accurate way to construct a 3-D face model is by the laser scanners. When the 3-D structure of a face is available, the process of reconstructing a

3-D scene and the recovery of the depth information given a set of human face images has been investigated. Some of these systems, such as the C3D^{tm1} or the Cyberware^{tm} scanners, are commercially available. However, these laser scanners are expensive, and the data is usually noisy. In addition, the scanning time of a face and a body is approximately 5-20s, and the subject must remain stationary.

Since the cost of computers, digital cameras and video cameras is relatively low, producing the 3-D human face models directly from a sequence of 2-D images are become of great interest. Some research has been conducted in this area. For example, structure-from-stereo [80], [107], [32], [41], [124], [43], [59] and structure-from-motion [76], [102], [103], [13], [44], [111], [15], [90] are two common existing methods to recover 3-D coordinates from multiple 2-D images. However, the accuracy of the constructed 3-D face model cannot be evaluated because the 3-D human face structure is usually unknown. In order to perform human face modeling, efficient algorithms that can construct the corresponding 3-D face model are indispensable. The 3-D information about a human face can be used for face recognition so that a more accurate recognition can be achieved.

1.2.2 Gabor Features Extraction and Selection

A face recognition methodology includes representation and classification issues. A good representation method should require minimum manual annotations. GWs are similar to the 2-D receptive field profiles of the mammalian cortical simple cells, and exhibit desirable characteristics of spatial locality and orientation selectivity. The visual neurons optimize the general uncertainty for resolution in space, spatial frequency, and orientation. Since the Gabor filters have similar characteristics to the visual neurons of the human visual system, they allow for a good image feature representation. The biological relevance and computational properties of the GWs for image analysis have been described in [36], [37], [38]. To extract the Gabor features, the GWs are applied to the whole image through a convolution, resulting in Gabor filtered images [1], [51]. The effect of filtering an image is to break down image content into different scales, locations, and orientations to allow discriminative features to be extracted for classification. Okajima [48] derived Gabor functions as solutions for a certain mutual-information maximization problem. It shows that the Gabor receptive field can extract the maximum information from local image regions. In [47], the top two performers in the 2004 International Face Verification Competition used GWs to extract features from images. This showed that GWs play an important role in face recognition.

Although the GWs have a good performance record for face recognition, the feature extraction of all possible orientations and scales at every pixel in an image is computational intensive. For face recognition applications, 40 Gabor filters (5 scales and 8 orientations) are usually used to convolve a face image, and therefore the dimension of the feature vectors extracted is incredibly large. For example, an image of size 64×64 will result in a feature vector of dimension 163,840 when 40 Gabor filters are used. As a result, when GWs are used for face recognition, the number of wavelets should be as small as possible. A GWs can give a maximal response at an image location which has a similar orientation and scale. In order to find an efficient and effective representation, a subset of image locations will be selected. In each selected image location, only some specific GWs will be chosen as the Gabor features for classification.

1.2.3 Indexing and Retrieval from Large Face Databases

The storage requirement of digital information becomes more and more since more applications using the digital information have been developed. Human face recognition is one kind of these applications. In practice, the number of face images in a database may range from hundreds to several tens of thousands. Many problems have been considered when developing a practical human face recognition system. Thus, an efficient indexing and retrieval scheme is required to make this application practically.

In a face recognition system, the features for each human face have already been extracted and store them in the face database. When a query image inputs, the same type of features are also extracted and then compared with the features in the face database. To recognize the human face, the feature distances between the query image and the database images are computed. The database image corresponding to the minimum distance is the same subject as the query image. However, the computation of all the feature distances between the query image and the database images is too expensive. If the number of face images in the database can be reduced, the runtime required for face recognition can also be reduced.

1.3 Statements of Originality

The following contributions reported in this thesis are claimed to be original.

1. An efficient 3-D face reconstruction method is proposed. In our approach, a multiple of face images under different poses of the same subject are used to construct a 3-D face model. The structure of the 3-D face model is derived by minimizing the feature-point distance between the projected

3-D face model and a 2-D training face image which is measured by the similarity transform.

2. The measurement of the accuracy of the constructed 3-D face model based on the similarity transform is also suggested. After constructing a 3-D face model, it can be compared to their corresponding training face images by using the Levenberg-Marquardt method to optimize the similarity distance between the face model and the respective face images.
3. Having constructed a group of face models, the 3-D face database based on the 3-D face structure can be developed. Hence, in addition to the conventional 2-D face recognition techniques, the 3-D information can also be considered. The face recognition system can extend the 2-D-to-2-D point set matching technique to the 2-D-to-3-D point set matching. This provides additional discriminant information that does not exist in the 2-D images.
4. A novel Gabor features selection and extraction method is proposed. Since the responses of the Gabor wavelets are strongly related to these edges which are perpendicular to the wave vectors, the oriented edges images are used to select the Gabor features. In the meantime, the Gabor features of the images are extracted using a simplified version of the Gabor wavelet;

this can save about 30% extraction time compared to using the original Gabor wavelet.

5. The method to construct a condensed database based on the vantage object and the Gabor features is also proposed. Therefore, the runtime required for face recognition can be further reduced because the number of the face images considered in the database is decreased.

1.4 Organization of the Thesis

The thesis is organized as follows:

There are two sections in Chapter 2. The first section gives a brief review of the current 3-D reconstruction and 3-D face modeling techniques for face recognition. In this part, multiple view geometry and some well-known 3-D construction techniques - structure-from-stereo and structure-from-motion - are described. After that, some 3-D face reconstruction and 3-D face recognition techniques are reviewed. The second section is a review of the face recognition techniques based on Gabor features. In this section, some methods for Gabor features selection and extraction are presented.

Our proposed 3-D face reconstruction approach will be described in Chapter 3. In our proposed algorithm, three or more face images of the same subject under different poses are used to construct a 3-D face model using the similarity transform. In the measurement, the 3-D face model is adjusted to the poses of the 2-D face image to be compared by the genetic algorithm (GA). In the meantime, the depths of some facial feature points are computed. The similarity transform can also be used to measure the accuracy of the constructed 3-D face model and 3-D face recognition.

Chapter 4 describes our proposed Gabor features selection method for face recognition in large face databases. The Gabor features are selected based on the oriented edges images. In the meantime, the Gabor features of the images are extracted by using the simplified Gabor wavelet; this can reduce the extraction runtime by 30% compared to using the original Gabor wavelets. The runtime required for face recognition can be further reduced by constructing a condensed database based the vantage object structure and the Gabor features.

Finally, the conclusion of our work is given in Chapter 5, and some suggestions have been provided for further development.

Chapter 2 : Literature Review

In this chapter, we will first give a brief survey of the current literature in 3-D reconstruction and 3-D face modeling for face recognition. Multiple-view geometry and some well-known 3-D construction techniques will be introduced. In addition, some 3-D face recognition techniques will be described. Then, a review of face recognition techniques based on Gabor features will be presented. Methods for the selection and extraction of Gabor features for face recognition are also reviewed.

2.1 3-D Reconstruction

3-D reconstruction refers to the 3-D structure inferred from 2-D images. Because of the image formation process of a camera, the 3-D structure of the objects in an image is lost. Structure-from-stereo [80], [107], [32], [41], [43], [124] and structure-from-motion [76], [102], [103], [13], [44], [15], [90], [121], [71] are two popular approaches for recovering 3-D structure from digital images since they do not need any special hardware for 3-D reconstruction. Before reviewing these two common approaches and 3-D face recognition, some basic concepts such as camera models, coordinate systems and camera calibration are illustrated first.

2.1.1 Camera Models

A camera is a device which maps the 3-D world to a 2-D image. The camera models can be categorized into two major classes: models with a centre “at infinity” and cameras with a finite centre. These two camera models correspond to two projection models: the orthographic model and the perspective model. The perspective projection model is a realistic model of the imaging process, whereas orthographic projection is far easy-to-solve models that are applicable in some simple cases. The orthographic projection model is employed primarily because it uses mathematically tractable equations. It is a reasonable approximation of objects that have a small field of view and whose distance does not change dramatically.

Figure 2.1 shows the orthographic projection model, in which the projection lines are parallel to each other and the principal ray. In order to determine how 3-D objects in the world appear in 2-D camera images geometrically, three different coordinate systems - the world coordinate system, the camera coordinate system and the image coordinate system - are defined to represent these objects. The world coordinate system is a fixed, 3-D frame of reference for representing 3-D objects and scenes in the world. The camera coordinate system is another 3-D coordinate system but it corresponds to the camera's location and orientation. The third coordinate system is the image coordinate system, which is a frame of reference for the pixel coordinates of a 2-D camera image. Orthographic projection considers the mapping between a point in space with coordinates $\mathbf{X_c}=(X_c, Y_c, Z_c)^T$ and the point $\mathbf{x_p}=(x_p, y_p)^T$ on the image plane. The projection equation can be rewritten as:

$$\mathbf{x_p} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{X_C} . \tag{2.1}$$

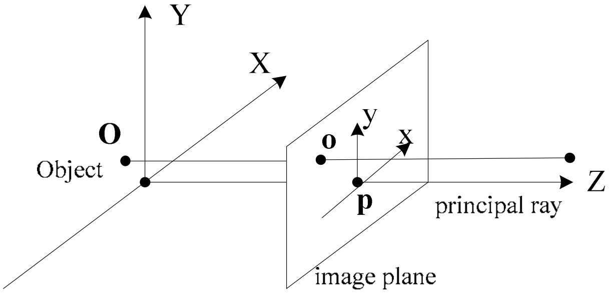


Figure 2.1: Orthographic projection.

Figure 2.2 shows the pinhole camera model [26] which corresponds to the perspective projective. The intensity of an object is formed on the camera's image plane through perspective projection. The camera is represented as a small hole through which light travels. In other words, all projection rays from the camera intersect at the camera centre.

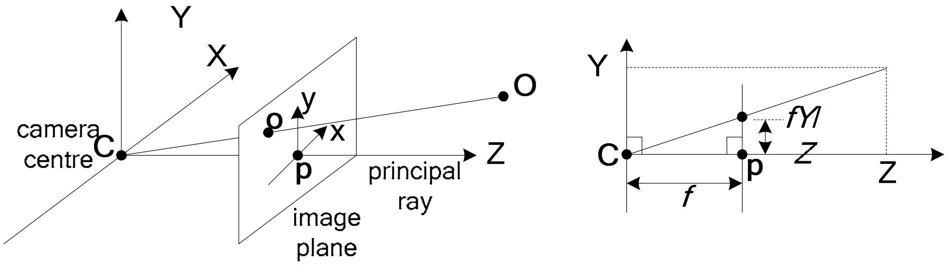


Figure 2.2: The pinhole camera model.

Let the centre of projection **C** be the origin of the Euclidean coordinate system, and the image plane Z=*f*, where *f* is the focal length of the camera. The pinhole camera model considers the mapping between a point in space with coordinates $\mathbf{X_c}=(X_c, Y_c, Z_c)^T$ and the point on the image plane. The line joins the point **O** to the centre of projection, which meets the image plane. By using similar triangles, the mapping between the point $(X_c, Y_c, Z_c)^T$ and the 2-D point $(fX_c/Z_c,\ fY_c/Z_c,\ f)^T$ on the image plane can be computed. The line from the camera centre perpendicular to the image plane is called the principal ray, which intersects the image plane at the principal point **p**.

In general, points in space are expressed in world coordinate system. Let $\mathbf{X_W}$ be a point of an inhomogeneous 3-D vector in the world coordinate system, and the corresponding point in the camera coordinate system is denoted as $\mathbf{X_C}$, then the mapping can be written as $\mathbf{X}_\mathrm{C} = R(\mathbf{X}_\mathrm{W} - \tilde{\mathbf{C}})$, where $\tilde{\mathbf{C}}$ is the camera centre location in the world coordinate system, and *R* is a 3×3 orthogonal rotation matrix representing the orientation of the camera coordinate system.

The inhomogeneous coordinates can be written as:

$$\mathbf{X_C} = R\mathbf{X_W} - R\tilde{\mathbf{C}} . \tag{2.2}$$

The parameters of R and $\tilde{\mathbf{C}}$ which are related to the camera orientation and position in a world coordinate system are called the extrinsic camera parameters.

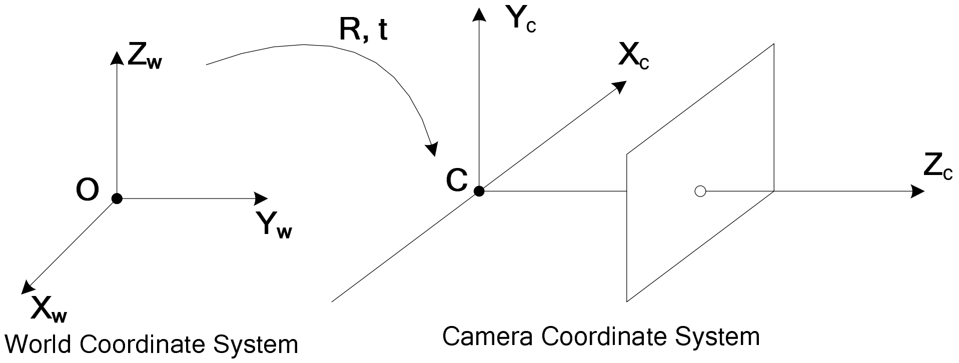


Figure 2.3: The Euclidean transformation between the world and camera coordinate frames.

If vectors are used to represent the points in the camera coordinate system and the image coordinate system, then central projection between linear mapping and their inhomogeneous coordinates can be expressed in matrix form as:

$$\begin{pmatrix} fX_C \\ fY_C \\ 1 \end{pmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{X_C} . \tag{2.3}$$

Generally, the principal point is located at (p_x, p_y), as shown in Figure 2.4.

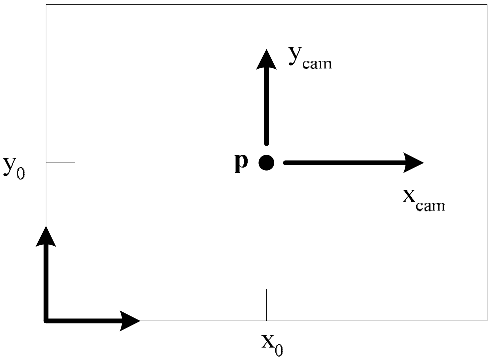


Figure 2.4: Image and camera coordinate system.

The mapping between a 3-D location $(X_C, Y_C, Z_C)^T$ and the 2-D image plane is

$$(X_C, Y_C, Z_C)^T = (f\frac{X_C}{Z_C} + p_x, f\frac{Y_C}{Z_C} + p_y)^T . \tag{2.4}$$

This equation can then be further expressed as

$$\begin{pmatrix} fX_C + Z_C p_x \\ fY_C + Z_C p_y \\ Z_C \end{pmatrix} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \mathbf{X_C} . \tag{2.5}$$

Let K be the matrix

$$K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} . \tag{2.6}$$

The projection equation can be rewritten as

$$\mathbf{x_p} = K\mathbf{X_C} . \tag{2.7}$$

The parameters in K are called the intrinsic camera parameters, which contain the internal orientation information. Combining (2.2) and (2.7), the following is obtained:

$$\mathbf{x_p} = KR\left[I \mid \tilde{\mathbf{C}}\right]\mathbf{X_W} . \tag{2.8}$$

The term $KR\left[I \mid \tilde{\mathbf{C}}\right]$ is the camera projection matrix. However, it is more common to transform the coordinate system as a rotation followed by a translation rather than the camera centre. (2.8) can be rewritten as

$$\mathbf{x_p} = K[R \mid t]\mathbf{X_W}, \tag{2.9}$$

where *t* is the translation matrix.

2.1.2 Camera Calibration

The computation of the camera projection matrix is a pre-processing step for the 3-D reconstruction under perspective projection. Many methods have been proposed for camera calibration [16], [84], [122]. However, these methods generally depend on *a priori* information such as the focal length and the skew. An alternative method [23] is to compute calibrations based on the imaged objects (planar object). The camera projection matrix was recovered from a set of images obtained either from multi-view, i.e. using a number of cameras, or from a video sequence where the object is in motion.

2.1.3 Structure-from-stereo

Structure-from-stereo [80], [107], [32], [41], [43], [124] infers depth information from images captured from different viewpoints. Therefore, a single pair of images of the same object is taken by two cameras at different locations and orientations. Depth information of this object is inferred by some computational techniques using the location offset between two images. The separation between the optical centres of the left and right cameras is called the baseline, and is usually created by a translation between the cameras' optical centres horizontally.

2.1.3.1 Stereo Geometry

The geometrical relationship between two images is already known due to the fixed configuration of the stereo system. If both the intrinsic and extrinsic parameters of the cameras are pre-determined by camera calibration, the problem of the structure estimation can be solved using a simple procedure known as triangulation.

In general, an object in the scene is represented with respect to a fixed world coordinate system or a fixed camera coordinate system, in which case the representation would differ from one camera to another if they have different positions or orientations. Therefore, the relationship between two camera coordinate systems in different cameras must be defined.

Let $\mathbf{X_{CL}}$ and $\mathbf{X_{CR}}$ be the left and right camera coordinates of the same point $\mathbf{X_W}$ in space, and R_L, R_R, t_R and t_L be the extrinsic parameters of the left and right cameras, respectively, such that

$$\mathbf{X_{CL}} = R_L \mathbf{X_W} + t_L \quad \text{and} \quad \mathbf{X_{CR}} = R_R \mathbf{X_W} + t_R \,. \tag{2.10}$$

$\mathbf{X_{CR}}$ can be expressed as

$$\mathbf{X_{CR}} = R\mathbf{X_{CL}} + t \tag{2.11}$$

where $R = R_R R_L^{-1}$ and $t = -R_R R_L^{-1} t_L + t_R$.

2.1.3.2 Triangulation and Recovery of 3-D Coordinates

As shown in Figure 2.5, let $\mathbf{X_{CL'}}$ and $\mathbf{X_{CR'}}$ be the unit vectors passing through $\mathbf{x_{pL}}$ and $\mathbf{x_{pR}}$ from the optical centers of two cameras, respectively. The objective of triangulation is to find the intersection between two vectors extrapolated from $\mathbf{X_{CL'}}$ and $\mathbf{X_{CR'}}$. Let K_L and K_R be the projective matrices for the left and right cameras. By applying the reverse of the projection on the homogeneous coordinates of $\mathbf{x_{pL}}$ and $\mathbf{x_{pR}}$, then two vectors are:

$$\mathbf{X_{CL'}} = K_L^{-1}\begin{bmatrix}\mathbf{x_{pL}}\\ 1\end{bmatrix} \text{ and } \mathbf{X_{CR'}} = K_L^{-1}\begin{bmatrix}\mathbf{x_{pR}}\\ 1\end{bmatrix}. \tag{2.12}$$

The extrapolated vectors may not intersect exactly because there may be some errors in feature extraction and camera calibration. As a result, $\mathbf{X_w}$ is estimated as the mid-point of the two lines orthogonal to both $\mathbf{X_{CL'}}$ and $\mathbf{X_{CR'}}$.

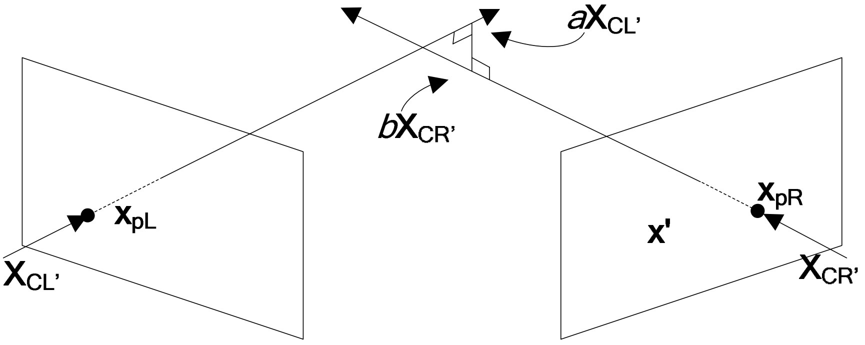


Figure 2.5: 3D reconstruction by triangulation.

Figure 2.5 can be expressed in the left camera coordinate system as follows:

$$a\mathbf{X_{CL'}} - R^{\mathbf{T}}(b\mathbf{X_{CR'}} - t) = c[\mathbf{X_{CL'}} \times R^{\mathbf{T}}(\mathbf{X_{CR'}} - t)]\,. \qquad (2.13)$$

(2.13) can be written as

$$\begin{bmatrix} \mathbf{X_{CL'}} & -R^{\mathbf{T}}\mathbf{X_{CR'}} & \mathbf{X_{CL'}} \times R^{\mathbf{T}}(\mathbf{X_{CR'}} - t) \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = R^{\mathbf{T}}t\,. \qquad (2.14)$$

a, b and *c* are determined by solving (2.14).

Let $\mathbf{X_{CL''}}$ be the estimation of $\mathbf{X_{CL}}$ which is the mid-point of $a\mathbf{X_{CL'}}$ and $R^{\mathrm{T}}(b\mathbf{X_{CR'}}\text{-}t)$. Therefore,

$$\mathbf{X_{CL''}} = \frac{a\mathbf{X_{CL'}} + R^{\mathbf{T}}(b\mathbf{X_{CR'}} - t)}{2}\,. \qquad (2.15)$$

2.1.3.3 Epipolar Geometry

Calibrating two views involves computing the epipolar geometry [59], [123]. As shown in Figure 2.6, suppose a point **X** in 3-D was imaged in two views, with point **x** in the first view and point **x'** in the second view. A typical stereo matching problem is to find the correspondence between **x** in one image and **x'** in another image. Both camera centres **C** and **C'**, the points **x**, **x'**, and **X** are coplanar, and this plane is called an epipolar plane. The line that connects the camera centres is called a baseline. The points **e** and **e'** where the baseline intersect the two views are called the epipoles. The lines connecting **x**, **e** and **x'**, **e'** are the epipolar lines. Theoretically, every feature in the right image is a potential match candidate for every feature in the left image; this makes feature matching be a large 2-D search problem. In order to solve the problem efficiently, the epipolar constraint can reduce the search problem to one dimension. From the definition of perspective projection, points **C**, **x**, and **X** are collinear and any point on this line between **x** and **X** projects as **x** in the first image. Therefore, the correspondence of **x** must lie on the projection of the line from **x** to **X** in the second image. Consequently, given the location of any feature point *i* on the left image, the search for the point's correspondence can be narrowed along the epipolar line.

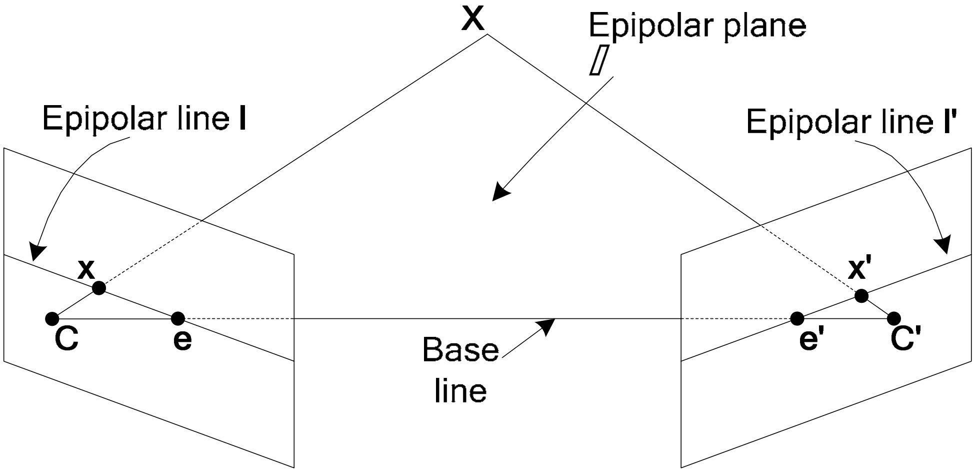


Figure 2.6: Epipolar geometry.

Epipolar geometry is the intrinsic projective geometry between two views. It is independent of scene structure. The only dependence is the internal camera parameters and the relative poses. Intrinsic geometry is encapsulated in the fundamental matrix, **F**, which is a 3×3 matrix of rank 2. A mapping is performed and the point **X** is imaged as **x** in the first image, and a correspondence **x'** in the second image. Then, the image points satisfy the epipolar constraint, $\mathbf{x'}^T\mathbf{F}\mathbf{x} = 0$. Armangué and Salvi [111], Zhang [123] have given an overview up to 19 of the most widely used techniques for computing the fundamental matrix. Shapiro et al. [59] considered the affine epipolar line properties and solved the affine epipolar line equation, and then determined all the unknown camera motion parameters if the feature correspondences between two images were available.

2.1.3.4 Advantages and Disadvantages

In the approach of structure-from-stereo, if the geometry of the camera system is known and the error of the feature correspondences is small, the calculation will be quite simple. However, the calibration is usually fastidious and not very reliable. The results of triangulation are reasonably insensitive to errors when the baseline in a stereo system is large. However, a large baseline causes geometric distortion and occlusion.

2.1.4 Structure-from-motion

Structure-from-motion [76], [102], [103], [13], [44], [121], [15], [90], [71] infers the depth information from images captured at different time. Typically, the camera motion and the movement of the scene are two types of motion in an image sequence. In this approach, the 3-D information about a collection of discrete structures, such as lines, curves and points, is recovered from a 2-D collection of such lines, curves and points. 2-D images are formed by projections from the 3-D world. Structure-from-motion recovers the original 3-D information by inverting the effect of the projection process. Two well-known projection models are the perspective projection model and the orthographic projection model, which have been described in 2.1.1.

2.1.4.1 Structure-from-motion under Orthographic Projection

Lots of research has been conducted on determining the motion and structure of rigid moving objects under orthographic projection. Ullman [98] has proved that four point correspondences over three views yielded a unique solution to motion and structure. The 3-D shape and motion of an object can be recovered by linear methods of factorization methods. Classic linear methods are mainly based on least-squares minimization and eigen-values minimization.

2.1.4.1.1 Linear method

Let $\mathbf{X_{1i}}$=(X_{1i}, Y_{1i}, Z_{1i}) and $\mathbf{X_{2i}}$= (X_{2i}, Y_{2i}, Z_{2i}) be the 3-D coordinates of the i^{th} point in an object at time T_1 and T_2, respectively. Also, let $\mathbf{x_{1i}}$=(x_{1i}, y_{1i}) and $\mathbf{x_{2i}}$=(x_{2i}, y_{2i}) be the image coordinates of the i^{th} point in an object at time T_1 and

T₂, respectively. Then,

$$\mathbf{X_{2i}} = R\mathbf{X_{1i}} + t \,, \tag{2.16}$$

where $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ is a 3×3 rotation matrix and t is a 3-D translation vector. Note that, with orthographic projection,

$x_{1i} = X_{1i}$, $y_{1i} = Y_{1i}$, $x_{2i}= X_{2i}$ and $y_{2i}= Y_{2i}$,

Rewriting (2.16), we have

$$\begin{bmatrix} x_{2i} \\ y_{2i} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} x_{1i} \\ y_{1i} \end{bmatrix} + \begin{bmatrix} r_{13} \\ r_{23} \end{bmatrix} Z_{1i} \,. \tag{2.17}$$

Then, eliminating Z_i by multiplying $[r_{23}\ –r_{13}]$ to both sides, (2.17) becomes

$$r_{23}x_{2i} - r_{13}y_{2i} + r_{32}x_{1i} - r_{31}y_{1i} = 0 \,. \tag{2.18}$$

(2.18) shows that it is impossible to determine the motion and structure uniquely from two orthographic views no matter how many point correspondences are available. In fact, the number of solutions is uncountable. Huang and Lee [103] presented a linear algorithm to obtain the 3-D motion and structure parameters by using three images under different views. However, Hu and Ahuja [116] argued that when the object rotates around the optical axis, the motion can be uniquely determined in a two-view problem. However, this method does not consider structure estimation. In a three-view problem, Hu and Ahuja [116] refined the results that were followed by Huang and Lee [103]. Xirouhakis and Delopoulos [121] have proposed to extract the motion and shape parameters of a rigid 3-D object by computing the rotation matrices via the eigenvalues and eigenvectors of appropriate defined 2×2 matrices, where the eigenvalues are the expressions of four motion vectors in two successive transitions. If more than four motion vectors per transition are available, a least squares estimation of the rotation matrices involved can be performed.

2.1.4.1.2 Factorization method

[52], [15], [63], [101] can calculate the structure and motion of an object from a sequence of tracked feature points using the factorization method. The input of this method is the 2-D coordinates of the tracked feature points, while the output is the 3-D coordinates of the points and the base vectors of the camera planes in all frames. Tomasi and Kanade [15] used the singular value decomposition (SVD) technique to factorize the measurement matrix into two matrices, which represent the object shape and the camera motion, respectively. Let N be the number of feature points in an object across F frames, $\mathbf{x_{if}}=[x_{if}, y_{if}]^T$ be the 2-D image coordinates of the i^{th} feature point at frame f, and $\mathbf{X_i}=[X_i, Y_i, Z_i]^T$ be the 3-D coordinates of the i^{th} feature point of an object. Rewriting (2.16), the 2-D image coordinates can be written as follows:

$$\mathbf{x_{if}} = R_f \mathbf{X_i} + t_f \ , \qquad (2.19)$$

where R_f represents the first two rows of the rotation matrix with respect to frame *f*, and t_f be the 2-D offset.

For all points in all the images, (2.19) can be written as

$$W = M \cdot S\,, \tag{2.20}$$

where $W = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ y_{11} & \cdots & y_{1N} \\ & \vdots & \\ x_{F1} & \cdots & x_{FN} \\ y_{F1} & \cdots & y_{FN} \end{bmatrix}$ is a measurement matrix, $M = \begin{bmatrix} R_1 \\ \vdots \\ R_f \end{bmatrix}$ is the motion matrix and $S = \begin{bmatrix} X_1 & \cdots & X_N \\ Y_1 & \cdots & Y_N \\ Z_1 & \cdots & Z_N \end{bmatrix}$ is the shape matrix.

The measurement matrix *W* is factorized to obtain the shape matrix *S*. The rank of *W* is reduced to three by the SVD. The rank of *W* is at most three because it is the product of the 2*F*×3 motion matrix *M* and the 3×*P* shape matrix *S*. This factorization is only determined up to a 3×3 linear transformation. The 3×3 non-singular matrix *G* can be inserted so that $W = \tilde{M}GG^{-1}\tilde{S}$. The estimated rotational matrix can be written as $\tilde{R} = \tilde{M}G$, where $\tilde{R} = \begin{bmatrix} \tilde{R}_1 \\ \vdots \\ \tilde{R}_f \end{bmatrix}$ and $\tilde{M} = \begin{bmatrix} \tilde{M}_1 \\ \vdots \\ \tilde{M}_f \end{bmatrix}$.

A closed-from solution for *G* under orthography has been originally proposed by Morita and Kanade [101]. For orthographic projection, the base vectors in each frame are orthonormal. Therefore, there are three rotational constraints per frame.

$$\tilde{R}_{if}\tilde{R}_{jf}^{T} = \tilde{M}_{if}G^{T}G\tilde{M}_{jf}^{T}\begin{cases}1 & \textbf{if } i = j \\ 0 & \textbf{if } i \neq j\end{cases} \tag{2.21}$$

where $\tilde{R}_{if}$ is the i^{th} row of $\tilde{R}_f$, $\tilde{M}_{if}$ is the i^{th} row of $\tilde{M}_f$ and i=1, 2.

Let the symmetric matrix $L = G^T G = \begin{bmatrix} l_1 & l_2 & l_3 \\ l_2 & l_4 & l_5 \\ l_3 & l_5 & l_6 \end{bmatrix}$. The elements of L can be calculated by the least squares solution of the over-determined system:

$$\begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{bmatrix} = \begin{bmatrix} p(\tilde{M}_{11}^{T}, \tilde{M}_{11}^{T}) \\ p(\tilde{M}_{21}^{T}, \tilde{M}_{21}^{T}) \\ p(\tilde{M}_{11}^{T}, \tilde{M}_{21}^{T}) \\ \vdots \\ p(\tilde{M}_{1F}^{T}, \tilde{M}_{1F}^{T}) \\ p(\tilde{M}_{2F}^{T}, \tilde{M}_{2F}^{T}) \\ p(\tilde{M}_{1F}^{T}, \tilde{M}_{2F}^{T}) \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \\ 0 \\ \vdots \\ 1 \\ 1 \\ 0 \end{bmatrix}, \tag{2.22}$$

or briefly l=$P^{-1}c$, where P^{-1} is the pseudo-inverse of P and $p(a, b)$ is

$$[a_1b_1, a_1b_2 + a_2b_1, a_1b_3 + a_3b_1, a_2b_2, a_2b_3 + a_3b_2, a_3b_3].$$

The matrix G can be obtained by the eigen-decomposition of L.

Morita and Kanade [101] modified [15] by using a covariance-like matrix instead of feature positions so that the size remains constant when the number of frame increases. The SVD is replaced by an updated computation of only three dominant eigenvectors so that the computational time can be reduced. Xi in [63] and Hajder in [52] improved the outliers by computing SVD using linear l_1-norm regression and an iterative method, respectively.

2.1.4.1.3 Advantages and disadvantages

The computation is quite simple for the 3-D reconstruction based on orthographic projection because intrinsic parameters are not considered in this case. However, the approximation of the affine camera model is not proper when the scene is close to the camera. The reconstruction methods based on this camera model yield distorted shapes due to the perspective effect. Therefore, perspective reconstruction of 3-D structure and motion has been considered.

2.1.4.2 Structure-from-motion under Perspective Projection

2.1.4.2.1 Linear method

[13], [44], [90], [102] show a linear algorithm to obtain the 3-D motion and structure of a rigid body by observing the corresponding projected features at two different instants of time. Let $\mathbf{X_{1i}}=(X_{1i}, Y_{1i}, Z_{1i})$ and $\mathbf{X_{2i}}=(X_{2i}, Y_{2i}, Z_{2i})$ be the i^{th} point of an object at time T_1 and T_2, and $\mathbf{x_{1i}}=(x_{1i}, y_{1i})$ and $\mathbf{x_{2i}}=(x_{2i}, y_{2i})$ represent the perspective coordinates of $\mathbf{X_{1i}}$ and $\mathbf{X_{2i}}$ onto the image plane. The rigid body motion equation is given as

$$\mathbf{X_{2i}} = R\mathbf{X_{1i}} + t \,, \tag{2.23}$$

where R is a 3×3 rotation matrix and t is 3×1 translation vector.

Let t' be a non-zero vector which is collinear with t, taking its cross-product with both sides of (2.23), and then taking the inner product of both sides with (x_{2i}, y_{2i}, 1),

$$(x_{2i}, y_{2i}, 1)(t' \times R)(x_{1i}, y_{1i}, 1)^T = 0\,, \tag{2.24}$$

where $t' \times R = [t' \times r_1,\ t' \times r_2,\ t' \times r_3]$ and r_1, r_2 and r_3 are the columns of R. Let $E = t' \times R$, then (2.24) becomes

$$(x_{2i}, y_{2i}, 1)E(x_{1i}, y_{1i}, 1)^T = 0\,. \tag{2.25}$$

Suppose there are N correspondences. Let

$$A = \begin{bmatrix} x_{21}x_{11} & x_{21}y_{11} & x_{21} & y_{21}x_{11} & y_{21}y_{11} & y_{21} & x_{11} & y_{11} & 1 \\ x_{22}x_{12} & x_{22}y_{12} & x_{22} & y_{22}x_{12} & y_{22}y_{12} & y_{22} & x_{12} & y_{12} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{2n}x_{1n} & x_{2n}y_{1n} & x_{2n} & y_{2n}x_{1n} & y_{2n}y_{1n} & y_{2n} & x_{1n} & y_{1n} & 1 \end{bmatrix},$$

$$E = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \text{ and}$$

$$h = [h_1 \quad h_2 \quad h_3 \quad h_4 \quad h_5 \quad h_6 \quad h_7 \quad h_8 \quad h_9]^T\,.$$

Then, (2.25) can be transformed into the over constraint linear equation for h

$$Ah = 0\,. \tag{2.26}$$

The nine components vector h is found by using the least-squares method.

E has two decompositions, $t' \times R$ and $(-t') \times R$. Note that $E = [t' \times r_1,\ t' \times r_2,\ t' \times r_3]$. Therefore, three columns span a 2-D space and three constraints are shown as follows:

$$rank(E) = 0\,,$$

$$\|E\| = 2\|t'\|\,, \text{ and}$$

$$E^T t' = 0\,. \tag{2.27}$$

(2.27) can be solved by using the least-squares method for *t'*, and the value of the vector *t'* can be obtained using the other constraints. Taking a cross-product with both sides of (2.23) by (x_{2i}, y_{2i}, 1),

$$Z_{1i}(x_{2i}, y_{2i}, 1)^T \times \left[R(x_{1i}, y_{1i}, 1)^T\right] + (x_{2i}, y_{2i}, 1)^T \times t' = 0\ . \tag{2.28}$$

When Z_{1i}<0, *t* has the same orientation as *t'* or (-*t'*) if and only if $(x_{2i}, y_{2i}, 1)^T \times \left[R(x_{1i}, y_{1i}, 1)^T\right]$ has the same orientation as $(x_{2i}, y_{2i}, 1)^T \times t'$ or $-(x_{2i}, y_{2i}, 1)^T \times t'$. It has the same orientation if and only if

$$\sum_{i=1}^{n} (x_{2i}, y_{2i}, 1)^T \times \left[R(x_{1i}, y_{1i}, 1)^T\right] + (x_{2i}, y_{2i}, 1)^T \times t' \geq 0 \textbf{ or} \leq 0\ . \tag{2.29}$$

Once the correct *t'* is determined, the true *R* can be uniquely determined through *E=t'*×*R*,

$$R = \left[E_2 \times E_3 \quad E_3 \times E_1 \quad E_1 \times E_2\right] - (t' \times E), \tag{2.30}$$

where *E*=[E_1 E_2 E_3].

2.1.4.2.2 Factorization method

Perspective reconstruction of the 3-D structure and motion can be obtained using a factorization algorithm [7], [40], [65], [73], [83], [91], [95]. The factorization algorithm is used to recover the projective structure and motion from the image measurements. The factorization technique for structure-from-motion was introduced by Tomasi and Kanade [15] for orthographic reconstruction and Poelman and Kanade [8] for the weak-perspective reconstruction of rigid structures. Sturm [83] and Triggs [7] showed that by scaling the projective depths in the image measurements, full perspective reconstruction could be achieved using the factorization technique. The projective depths can be recovered using the pair-wise constraints among images. [95], [65] used the subspace constraints in the entire set of measurements to recover the projective depths more reliably.

Let N be the number of feature points in an object across F frames, and P_f be the unknown 3×4 image projection matrices, $\mathbf{x_{fi}}=[x_{fi}, y_{fi}, 1]^T$ be the 2-D homogeneous coordinates of the i^{th} feature point at frame f, and $\mathbf{X_i}=[X_i, Y_i, Z_i, 1]^T$ be 3-D homogeneous coordinates of the i^{th} feature point of an object. The basic image projection equation is

$$\lambda_{fi}\mathbf{x_{fi}} = P_f\mathbf{X_i}\,. \tag{2.31}$$

λ_{fi} are the unknown scaling factors called projective depths. The complete set of image projections can be written as

$$W = \begin{bmatrix} \lambda_{11}\mathbf{x_{11}} & \lambda_{21}\mathbf{x_{21}} & \cdots & \lambda_{F1}\mathbf{x_{F1}} \\ \lambda_{12}\mathbf{x_{12}} & \lambda_{22}\mathbf{x_{22}} & \cdots & \lambda_{F2}\mathbf{x_{F2}} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{1N}\mathbf{x_{1N}} & \lambda_{2N}\mathbf{x_{2N}} & \cdots & \lambda_{FN}\mathbf{x_{FN}} \end{bmatrix}^T = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_F \end{bmatrix} \begin{bmatrix} X_1 & X_2 & \cdots & X_N \end{bmatrix}. \tag{2.32}$$

The $3N\times F$ rescaled measurement matrix W has a rank of at most 4. The projective depths can be recovered from fundamental matrices and epipoles. Once the projective depths are obtained, the rescaled measurement matrix W

can be factorized by the SVD. Since *W* has a rank of at most 4, only the first four columns of **U** and first four rows of **V** are needed.

$$W = U'DV', \tag{2.33}$$

where *U'* is 3*F*×4 and *V'* is 4×*N*. Any factorization of *D* into two 4×4 matrices *D'* and *D"* and *D*=*D'D"*,

$$W = U'D'D''V' = \begin{bmatrix} \hat{P}_1 \\ \hat{P}_2 \\ \vdots \\ \hat{P}_F \end{bmatrix} \begin{bmatrix} \hat{X}_1 & \hat{X}_2 & \cdots & \hat{X}_N \end{bmatrix}, \tag{2.34}$$

where $U'D' = \begin{bmatrix} \hat{P}_1 \\ \hat{P}_2 \\ \vdots \\ \hat{P}_F \end{bmatrix}$ and $D''V' = \begin{bmatrix} \hat{X}_1 & \hat{X}_2 & \cdots & \hat{X}_N \end{bmatrix}$.

2.1.4.2.3 Advantages and disadvantages

The structure estimated using two frames is very sensitive to noise. Using sequence of images can combat this problem. In addition, using a 3-D reconstruction from the motion without knowing the magnitude of the relative translation between the camera and the scene, the depths of the object points cannot be determined exactly, but only within a global scale factor. For example, if one object is twice as far away as another, but twice as big, the resulting images of the two objects will be exactly the same.

2.1.5 3-D Face Modeling for Face Recognition

The previous sub-sections have reviewed some common 3-D reconstruction techniques. In this section, we will focus on 3-D face model reconstruction and face recognition using 3-D data.

There have been many existing methods for 3-D face model reconstruction based on structure-from-motion [125], [93], [119], [3] and structure-from-stereo [89]. Zhang et al. [119], [125] refined the face model and the head pose estimated by model-based bundle adjustment, which directly searches in the face model space to optimize the head motion and 3-D coordinates. Kang and Jones [93] proposed a method called appearance-based constrained structure from motion (AbCSfm) to refine the objects whose general structure is known, such as faces, but little discernable texture appeard in significant parts of the surfaces. As a result, these refinement techniques are useful for textured 3-D animated face models. Lengagne et al. [89] proposed using the *a priori* information in a reconstruction process from a sequence of calibrated face images so that the accuracy of the 3-D reconstruction using the conventional stereo algorithms can be enhanced. A 3-D mesh modeling the face is iteratively deformed, and the optimal face structure depends on an energy function.

Excluding structure-from-motion and structure-from-stereo, Sengupta and Ko [49], Sengupta and Burman [50] presented a method for 3-D face modeling from a monocular image sequence. They used a group of image frame pairs from the affine camera projection geometry, and then the spline-fitting technique was adopted for the model to express the depth at each point on the face. The spline-fitting technique is based on a modified nonparametric regression method which can estimate the curve, so that it can be used to represent the face structure. Another face modeling technique was based on a morphable model. Blanz and Vetter [105] derived a morphable face model based on the statistics of a large dataset of 3-D face scans (geometric and textural data, *Cyberware*^{TM}). This morphable face model was based on the linear combination of a large number of 3-D face scans. To model different 2-D input face images, a probability distribution was imposed on the morphing function so that shape and texture vary for different face images.

Nandy and Ben-Arie [25], Malassiotis and Strintzis [96] proposed 3-D face modeling methods that can address the problem of face recognition under varying pose and illumination. Malassiotis and Strintzis [96] used a pair of 2-D and 3-D images to produce a pair of normalized images for frontal view and illumination. A combination of 2-D color, 3-D image and geometry of the face was used to cope with the pose and illumination variations. Nandy and Ben-Arie [25] recovered the 3-D surfaces of the faces from a single image. The principle of this method was to recover the 3-D structure of a specific part of a face by specialized back propagation based neural networks. These parts of a face were represented based on principal components, and were then mapped to another set of principal component coefficients that represented the corresponding 3-D surface shape.

The 3-D face model is an effective tool for face recognition. Heseltine et al. [92] and Lu et al. [108] used the 3-D information of the faces, which was captured by the laser scanner, for face recognition. However, 3-D information captured by laser scanner is not commonly used. As a result, a group of 2-D face images was used to construct the face model for face recognition. [1], [23], [25], [9] used the 3-D facial feature point sets for face recognition. Ansari and Abdel-Mottaleb [1] proposed a method for automatic 3-D face modeling using the facial features. These features are detected from a pair of orthogonal frontal and profile view images of a person because the frontal-view image provides the 2-D face information, and the profile-view images provides the depth information about the human face. For face recognition, the query face model was aligned to the face models in the database and then the distances were computed. Gordon [23] and Tanaka et al. [25] presented the depth and curvature features for face recognition. Chua et al. [9] proposed a face recognition technique based on the Point Signature – a representation for free-form surfaces. This algorithm can handle various facial expressions for face recognition. Lee [69] proposed a pose-invariant face recognition system based on a 3-D face model. This face model was constructed from a composite of an edge model, a color region model of a face image. The query image was classified by finding the most similar synthesized face using a least-squares strategy.

Blanz and Vetter [99] presented a method for face recognition using a morphable model reconstructed from a single image. The 3-D shape and texture of this morphable model are estimated from a single image, and face recognition can be performed across the variations in pose and a wide range of illuminations. A query image is represented by the morphable model and is then compared to the shape and texture parameters of the models in the database. Jiang et al. [15] also used a single image to morph the face model. Then, the face model with different poses, illuminations and expressions was synthesized based on the personalized 3-D face to characterize the face subspace. Finally, face recognition was conducted based on these face models using the nearest neighbors for classification. Another 3-D face recognition method using the morphable models was suggested by Zhang and Cohen [10]. However, this 3-D face model has to be constructed from multiple face images under different views in order to estimate the *a priori* unknown poses. A cubic explicit polynomial in 3-D was used to morph a generic face into the specific face structure iteratively until the distance map metric was minimized. This distance map residual error and the image intensity residual error were used for face recognition.

2.2 Gabor Features Selection and Extraction

In this section, a Gabor wavelet function is first described. Then, the dimensionality of the Gabor features reduced by a linear subspace method and a kernel-based method are introduced. In addition, methods that select the optimal positions in a face image for the Gabor features used for face recognition are also reviewed.

2.2.1 Gabor Wavelets

In the spatial domain, a Gabor wavelet (GW) is a complex exponential modulated by a Gaussian function [68], [104], which is shown in the following:

$$\psi_{\omega,\theta}(u,v) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{(u\cos\theta + v\sin\theta)^2 + (-u\sin\theta + v\cos\theta)^2}{2\sigma^2}\right)} \cdot \left[e^{i(\omega u\cos\theta + \omega v\sin\theta)} - e^{-\frac{\omega^2\sigma^2}{2}} \right], \qquad (2.35)$$

where *u*, *v* denote the pixel position in the spatial domain, ω is the radial center frequency of the complex exponential, θ is the orientation of the GW, and σ is the standard deviation of the Gaussian function. The relationship between σ and ω can be derived as,

$$\sigma = \frac{\kappa}{\omega} \text{ where } \kappa = \sqrt{2\ln 2}\left(\frac{2^{\phi}+1}{2^{\phi}-1}\right), \tag{2.36}$$

where *ø* is the bandwidth in octaves.

For face recognition applications, the number of Gabor filters used to convolve face images varies with applications, but usually 40 filters (5 scales and 8 orientations) are used [68], [60], [1]. Figure 2.7 shows the real part of the Gabor kernels at five different scales and eight different orientations, while Figure 2.8 shows the magnitudes of the Gabor kernels at the five different scales.

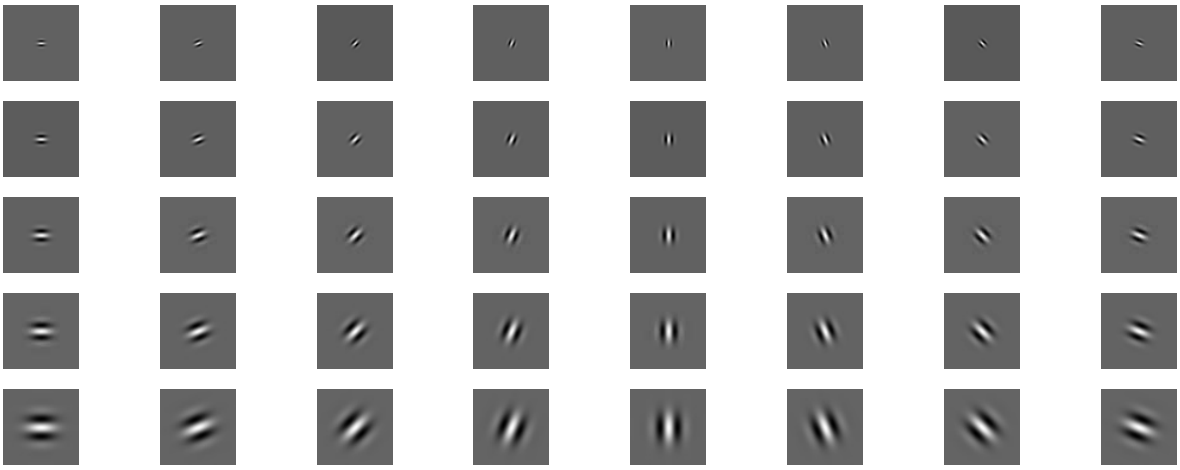


Figure 2.7: Real part of the Gabor kernels at five different scales and eight different orientations.

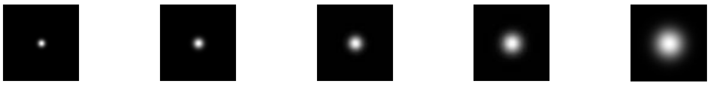


Figure 2.8: Magnitudes of the Gabor kernels at five different scales.

The Gabor wavelet representation of an image is the convolution of the image with a family of Gabor kernels. Let $I(u, v)$ be the gray level distribution of an image. The convolution output of the image I and the Gabor kernel $\psi_{\omega,\theta}(u, v)$ is defined as follows:

$$O_{\omega,\theta}(u,v) = I(u,v) * \psi_{\omega,\theta}(u,v), \tag{2.37}$$

where * denotes the convolution operator.

Applying the convolution theorem, the convolution output $O_{\omega,\theta}(u, v)$ can be computed using the fast Fourier transform (FFT) as follows:

$$\Im\{O_{\omega,\theta}(u,v)\} = \Im\{I(u,v)\}\Im\{\psi_{\omega,\theta}(u,v)\} \text{ and } O_{\omega,\theta}(u,v) = \Im^{-1}\Im\{I(u,v)\}\Im\{\psi_{\omega,\theta}(u,v)\}, \tag{2.39}$$

where $\Im$ and $\Im^{-1}$ denote the Fourier and the inverse Fourier transform, respectively.

Figure 2.10 shows the convolution outputs of the sample image shown in Figure 2.9. The outputs display strong characteristics of spatial locality, and scale and orientation selectivity corresponding to those displayed by the GWs. Such characteristics produce salient local features, such as the eyes, nose and mouth, that are suitable for visual event recognition. Only the magnitude - not the phase - is applied because magnitude can provide a more Gabor representation [68], [30]. A 1-D Gabor representation of an image can be obtained by concatenating the convolution output row by row, and is denoted as follows:

$$\mathbf{O}_{\omega,\theta} = \left[O_{\omega,\theta}(0,0), O_{\omega,\theta}(0,1), \ldots, O_{\omega,\theta}(0, N_r), O_{\omega,\theta}(0,1), \ldots, O_{\omega,\theta}(N_c, N_r)\right]^T , \qquad (2.40)$$

where N_c and N_r are the numbers of columns and rows in an image, respectively.

Figure 2.9: The face image of size 64×64.

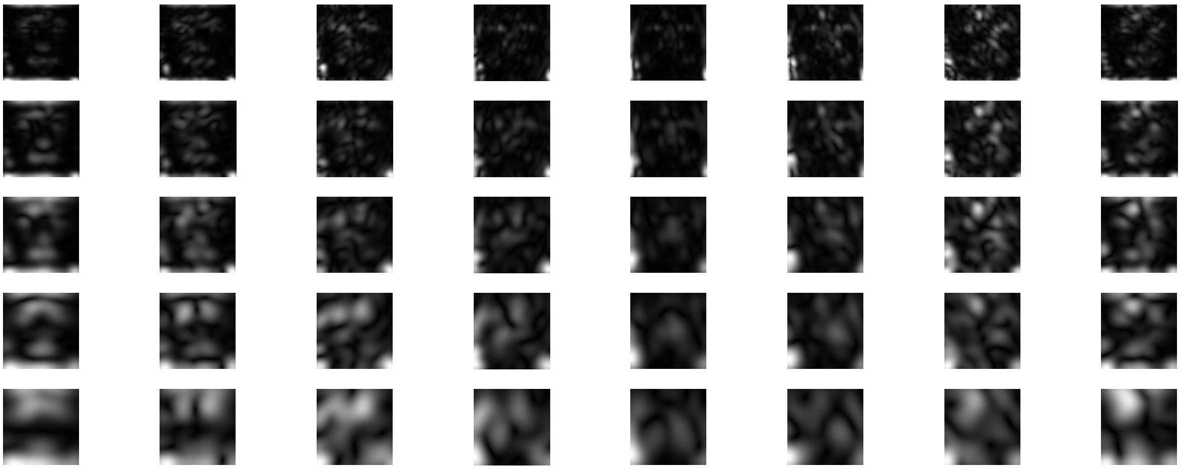


Figure 2.10: Magnitude of the convolution outputs of a sample image.

To encompass different local, scale, and orientation features of and image, the convolution outputs of the various Gabor wavelets are concatenated to derive a feature vector **Y**. Before the concatenation, the feature vector $\mathbf{O}_{\omega,\theta}$ is normalized to zero mean and unit variance. The augmented Gabor feature vector is then defined as follows:

$$\mathbf{Y} = \left[\mathbf{O}_{0,0}{}^T, \mathbf{O}_{0,1}{}^T, \ldots, \mathbf{O}_{4,7}{}^T\right]^T . \qquad (2.41)$$

2.2.2 Gabor Wavelets + Linear Subspace Analysis

Linear subspace analysis considers a feature space as a linear combination of a set of basis function. Principal Component Analysis (PCA) decomposes a Gabor feature vector as a combination of a sequence of basis vectors, which is an effective method to represent a feature vector with the number of basis vectors used being much lower than its dimension. Linear Discriminant Analysis (LDA) not only maximizes the between-class scatters of different Gabor feature vector of a subject, but also minimizes the within-class scatters of the same person. Independent Component Analysis (ICA) can be considered as a generalization of PCA. ICA searches for a linear transformation to express a set of random variables as the linear combinations of statistically independent source variables. While PCA considers the second-order moments only, and it un-correlates the data, ICA can further reduce the higher-order statistical dependencies.

2.2.2.1 Principal Component Analysis + Gabor Features

[48], [4], [21], [117] used the PCA to reduce the dimensionality of the Gabor features. Suppose that *m* Gabor features extracted from different subjects are used for training. Let the training Gabor features be $\mathbf{Y}_1$, $\mathbf{Y}_2$,..., $\mathbf{Y}_m$ and the

mean of these Gabor features is Ψ. Each training Gabor feature differs from the average Gabor feature by $\Phi_i = \mathbf{Y}_i - \Psi$.

The vectors $\mathbf{u}_k$ and the scalars λ_k are the eigenvectors and eigenvalues, respectively, of the covariance matrix of the demeaned face images Φ_i.

$$\mathrm{C} = \frac{1}{m}\sum_{n=1}^{m} \Phi_m{}^T \Phi_m = \mathrm{AA}^T, \tag{2.42}$$

where A = [Φ_1, Φ_2 ... Φ_m]. The eigenvalues, λ are selected such that

$$\lambda_k = \frac{1}{m}\sum_{n=1}^{m}\left(\mathbf{u}_k^T \Phi_n\right)^2 \tag{2.43}$$

is maximized, where $\mathbf{u}_l^T \mathbf{u}_k = \delta_{lk} = \begin{cases} 1\ \mathbf{if}\ l = k \\ 0\ \mathbf{if}\ l \neq k \end{cases}$.

The Gabor feature **Y** of a face is projected into the face space, and the projection is onto $\mathbf{u}_\mathrm{k}$, $\omega_\mathrm{k} = \mathbf{u}_\mathrm{k}^\mathrm{T}(\mathbf{Y} - \Psi)$. The weights form a vector $\Omega^\mathrm{T} = [\omega_1, \omega_2\ ...,\ \omega_m]$, which can be used for face recognition. However, the major problem with the use of PCA is that PCA is effective in representing a feature vector, but not in distinguishing the difference between feature vectors. Hence, this method has limited performance ability for classification. As a result, other subspace methods are more commonly used for classification.

2.2.2.2 Linear Discriminant Analysis + Gabor Features

[21], [120], [66], [118], [29] employed the LDA to reduce the dimensionality and enhance the discriminability of the Gabor features. Mathematically, there are two measures - within-class scatter matrix $\mathbf{S}_w$ and

between-class scatter matrix **S**$_b$ – describing the distances between the samples with classes and between the classes, respectively. **S**$_w$ and **S**$_b$ can be written as follow:

$$\mathrm{S}_w = \sum_{i=1}^{c}\sum_{j=1}^{N_i}\left(\mathbf{Y}_{ij} - \mu_i\right)\left(\mathbf{Y}_{ij} - \mu_i\right)^T , \tag{2.44}$$

where $\mathbf{Y}_{ij}$ is the j^{th} Gabor feature of class i, μ_i is the sample mean of class i, c is the number of classes, and N_i is the number of Gabor features in class i.

$$\mathrm{S}_b = \sum_{i=1}^{c}(\mu_i - \mu)(\mu_i - \mu)^T , \tag{2.45}$$

where μ represents the mean of all classes.

The optimal projection W_{opt} is chosen as the matrix which maximizes the ratio of the determinant of the between-class scatter matrix to the determinant of the within-class scatter matrix and whose columns are orthonormal to each other, i.e.,

$$\begin{aligned} W_{opt} &= \arg,\max_{W} \frac{\left|W^T S_b W\right|}{\left|W^T S_w W\right|} \\ &= \left[\mathbf{w}_1 \quad \mathbf{w}_2 \quad \dots \quad \mathbf{w_m}\right] \end{aligned} \tag{2.46}$$

where $\{\mathbf{w}_i \,|\, i = 1, 2, \dots , m\}$ is the set of generalized eigenvectors of $\mathbf{S}_\mathrm{B}$ and $\mathbf{S}_\mathrm{W}$ corresponding to the m largest generalized eigenvalues $\{\lambda_i \,|\, i = 1, 2, \dots , m\}$, i.e.,

$$S_B \,\mathrm{w}_i = \lambda_i S_W \,\mathrm{w}_i, \qquad i = 1,2,\cdots,m . \tag{2.47}$$

If $\mathbf{S}_\mathrm{W}$ is a nonsingular matrix, the optimal discriminant vectors can be solved with the following equation:

$$\left(S_W^{-1}S_B\right)\mathrm{w}_i = \lambda_i\, \mathrm{w}_i, \qquad i = 1,2,\cdots,m\,. \tag{2.48}$$

One drawback of the LDA is that the within-class scatter matrix $\mathbf{S}_{\mathrm{W}}$ is always singular due to the small sample size problem. Usually, the feature vector dimension of a face is very large, but only a few training examples per face are available. Liu and Wechsler [10] used the Enhanced Fisher Linear Discriminant Model (EFM) to enhance the performance of LDA. The EFM decomposes the LDA into two steps, whitening the within-class scatter matrix, and then applying the PCA on the between-class scatter matrix using the transformed data. Since those eigenvectors with small eigenvalues tend to capture noise, a proper balance between the selected eigenvalues of the eigenvectors accounts for most of the spectral energy of the raw data, and the requirement that the eigenvalues of the within-class scatter matrix in the reduced PCA space be not too small is preserved to enhance the performance of the EFM.

2.2.2.3 Independent Component Analysis + Gabor Features

Liu and Wechsler [11] proposed reducing the dimensionality of the Gabor feature vectors by mean of PCA first, and then defined the independent Gabor features based on the ICA. Let **X** be an *n* dimensional random vector corresponding to the PCA output. The covariance matrix of **X** is defined as follows:

$$\Sigma_{\mathbf{X}} = \varepsilon\left\{[\mathbf{X}-\varepsilon(\mathbf{X})][\mathbf{X}-\varepsilon(\mathbf{X})]^T\right\}, \tag{2.49}$$

where ε(•) is the expectation operator. The ICA of **X** factorizes the covariance matrix $\Sigma_{\mathbf{X}}$ into the following form:

$$\Sigma_{\mathbf{X}} = F\Lambda F^T, \tag{2.50}$$

where Λ is a positive diagonal matrix and *F* transforms the original random vector **X** to a new one **Z**, where **X** = *F***Z**, such that the *m* components ($m \leqq n$) of the new random vector **Z** are independent. Both *F* and **Z** are estimated by using the observable random vector **x** and some statistical assumption. The goal of ICA is to find the separation matrix W such that **Z** = **WX**. The search criterion of **W** involves the minimization of the mutual information of the new random vector **Z**.

The role of ICA is to find sparse feature code analogs to detect redundant features and to form a representation in which these redundancies are reduced and the independent features and Gabor feature vectors are represented explicitly. Thus, ICA provides a more powerful data representation than does PCA [39], [72].

2.2.3 Gabor Wavelets + Kernel-Based Method

In the kernel methods, the sample data is mapped to a higher dimensional feature space. A nonlinear problem in the original feature space is turned into a linear problem in the high-dimensional feature space, and it has been proved that kernel method can solve pattern recognition problems successfully. Kernel Principal Component Analysis (KPCA) and General Discriminant Analysis (GDA) are the kernel versions of PCA and LDA, respectively.

The support vector machine (SVM) approach is another example of the kernel method. SVM finds the hyperplane that separates the largest possible fraction of points of the same class on the same side while maximizing the distance from either class to the hyperplane for a two-class classification problem.

Three classes of kernel functions have been widely used for face recognition are polynomial kernels, Gaussian kernels and sigmoid kernels. Let $\mathbf{Y}_1$, $\mathbf{Y}_2,\ldots,\mathbf{Y}_M \in \mathbf{R}^N$ be the Gabor features in the input space where M is the number of Gabor features and N is the dimension of the feature vectors. Then, the three kernel functions are given as follows:

$$k\left(\mathbf{Y}_i,\mathbf{Y}_j\right)=\left(\mathbf{Y}_i,\mathbf{Y}_j\right)^d, \qquad (2.51)$$

$$k(\mathbf{Y}_i, \mathbf{Y}_j) = e^{\left(-\frac{\|\mathbf{Y}_i - \mathbf{Y}_j\|^2}{2\sigma^2}\right)} \text{, and} \tag{2.52}$$

$$k(\mathbf{Y}_i, \mathbf{Y}_j) = \tanh(\kappa(\mathbf{Y}_i, \mathbf{Y}_j) + \vartheta) . \tag{2.53}$$

where $d \in N$, $\sigma>0$, $\kappa>0$ and $\vartheta<0$. When $0<d<1$, this polynomial kernels include fractional power polynomial models.

2.2.3.1 Kernel Principal Component Analysis + Gabor Features

[12], [112], [6] proposed to use Gabor-based KPCA for face recognition. Let Φ be a nonlinear mapping between the input space and the higher-dimension feature space **F**:

$$\Phi : \mathbf{R}^N \to \mathbf{F} . \tag{2.54}$$

Assume the mapped data is centered and let K be a $M\times M$ Gram matrix,

$$K_{ij} = (\Phi(\mathbf{Y}_i) \cdot \Phi(\mathbf{Y}_j)) . \tag{2.55}$$

The orthonormal eigenvectors $\mathbf{v}_1$, $\mathbf{v}_2$ …,$\mathbf{v}_m$ of **K** corresponding to the m largest positive eigenvalues $\lambda_1>\lambda_2>\ldots>\lambda_m$ are computed. After that, the corresponding eigenvectors $\mathbf{b}_1$, $\mathbf{b}_2$ …,$\mathbf{b}_m$ for the KPCA can be written as

$$\mathbf{b}_j = \frac{1}{\sqrt{\lambda_j}} \mathbf{P}\mathbf{v}_j , \qquad j = 1,\ldots,m. \tag{2.56}$$

where $\mathbf{P}=[\Phi(\mathbf{Y}_1), \Phi(\mathbf{Y}_2) ..., \Phi(\mathbf{Y}_M)]$ is the mapped data matrix in the high-dimensional feature space. For a testing feature vector **Y**, **Y** is mapped to Φ(**Y**) and is then projected onto the eigenvector system $\mathbf{Q}=[\mathbf{b}_1, \mathbf{b}_2 \ldots,\mathbf{b}_m]$. The projected vector is:

$$\mathbf{w} = \left(w_1, w_{2,\ldots,} w_m\right)^T = \mathbf{Q}^T \Phi(\mathbf{Y})\,. \qquad (2.57)$$

2.2.3.2 General Discriminant Analysis + Gabor Features

Shen et al. [57] proposed using GWs and GDA for face identification and verification. Similar to KPCA, let Φ be a nonlinear mapping between the input space and the higher dimension feature space **F**. The idea behind GDA is to perform a classic LDA, which has been described in Section 2.2.2, but the feature space **F** replaces of the input space $\mathbf{R}^N$.

In face recognition tasks, the number of training samples is much smaller than the dimensionality of **F,** so the within-class scatter matrix may be degenerated. To solve this problem, either pseudo inverse or PCA has been used to remove the null space of the within-class scatter matrix. However, the null space may contain the most significant discriminant information. Lu et al. [42] suggested using Kernel Direct Discriminant Analysis (KDDA) to solve this problem. The basic idea is that the null space of the within-class scatter matrix may contain significant discriminant information if the projection of the between-class scatter matrix is not zero in that direction. The null space of the between-class scatter matrix can be discarded without a significant loses of information. Shen and Bai [54] have used the KDDA and GWs for face recognition. The experimental results showed that KDDA and Gabor wavelets have a better performance than KPCA and GWs, as well as GDA and GWs.

2.2.4 Graph Matching

Lades et al. [69] presented the dynamic link architecture (DLA) for distortion invariant object recognition. DLA first computed the Gabor jets of face images, and then elastic graph matching was used to compare the resulting image decomposition. Wiskott et al. [62] presented the elastic bunch graph matching, which extended the method proposed in [69].

2.2.4.1 Elastic Graph Matching

Elastic graph matching [69] recognized an object by using sparse graphs. The vertices or nodes are labeled with collections of features that describe the gray-level distribution locally with high precision, and globally with lower precision. The responses of the GWs to a facial feature point are called a Gabor jet. In elastic graph matching, a set of feature vectors is formed over a dense grid of image points. Also, sparse model graphs are formed and are labeled with jets from a rectangular sub-grid. The jets $J_n^I = a_n^I \exp(i ø_n^I)$ and $J_n^M = a_n^M \exp(i ø_n^M)$ in an image domain and a model domain, respectively, have magnitudes of a_n^I and a_n^M, and phases of $ø_n^I$ and $ø_n^M$. As part of the elastic graph matching, the similarity of pairs of vertex labels defined as

$$S_a\left(J^I, J^M\right) = \frac{\sum_n a_n{}^I a_n{}^M}{\sqrt{\sum_n a_n{}^{I\,2} \sum_n a_n{}^{M\,2}}} . \tag{2.58}$$

In the matching process, the preservation of topology between the image graph and the model graph is imposed by the constraint between the matching of the neighboring vertices. Edge labels between vertices x_i and x_j with the Euclidean distance vector is defined as:

$$\Delta_{ij} = x_j - x_i \qquad (i, j) \in E , \tag{2.59}$$

where *E* is the set of edges in the image or the model graph. The edge labels of the image graph are compared to the corresponding ones in the model graph by a quadratic comparison function

$$S_e\left(\Delta_{ij}^I, \Delta_{ij}^M\right) = \left(\Delta_{ij}^I - \Delta_{ij}^M\right)^2 . \tag{2.60}$$

Elastic matching of a model graph following a cost function is a minimum,

$$C_{total}\left(\left\{x_i^I\right\}\right) = \lambda \sum_{(i,j)\in E} S_e\left(\Delta_{ij}^I, \Delta_{ij}^M\right) - \sum_{n\in V} S_a\left(J_n{}^I\left(x_n^I\right), J_n^M\right), \tag{2.61}$$

which is a linear combination of an edge term and a vertex term. The coefficient λ controls the rigidity of the image graph, with large values penalizing distortion of the graph *I* with respect to the graph *M*. The process is repeated for every stored model graph, and the match with the lowest cost is identified as the model recognized.

Wiskott [61] used elastic graph matching for face recognition while Krüger et al. [71] and Lyons et al. [67] employed this method matching for pose estimation and facial expression recognition. Kotropoulos et al. [9] proposed to use multi-scale morphological operations instead of a set of Gabor jets in order to reduce the computational time for the elastic graph matching.

2.2.4.2 Elastic Bunch Graph Matching

In elastic graph matching, only the magnitudes of the coefficients are used for matching and recognition. Elastic bunch graph matching [62] has made three improvements to this. First, the phases of the complex Gabor wavelet coefficients are employed to find the location of the nodes. Assuming that two jets J_n^I and J_n^B refer to object locations with a small relative displacement **d**, the phase shifts can be approximately compensated for by the terms $\mathbf{dk}_n$, leading to a phase sensitive similarity function as follows:

$$S_a\left(J^I, J^B\right) = \frac{\sum_n a_n{}^I a_n{}^B \cos\left(\phi_n{}^I - \phi_n{}^B - \mathbf{dk}_n\right)}{\sqrt{\sum_n a_n{}^{I^2} \sum_n a_n{}^{B^2}}}. \qquad (2.62)$$

Second, object adapted graphs are used so that the correspondences between two faces can be found across large viewpoint changes. Third, the face bunch graph, which serves as a generalized representation of faces by combining the jets of a small set of individual faces, is used. The face bunch graph represented a set of M individual model graphs. The corresponding face bunch graph B will give the same grid structure as the individual graphs; its nodes are labeled with the bunches of jets J_j^{Bm} and its edges are labeled with the averaged distances $\Delta_{ij}^{B} = \sum_{m} \Delta_{ij}^{B} / M$. The similarity is defined as

$$S_B(I,B) = \frac{1}{N}\sum_{n}\max_{m}\left(S_\phi\left(J_n{}^I, J_n^B\right)\right) - \frac{\lambda}{B}\sum_{e}\frac{S_e\left(\Delta_{ij}^I, \Delta_{ij}^B\right)}{\left(\Delta_{ij}^B\right)^2} \tag{2.63}$$

A person is recognized when the correct model yields the highest graph similarity.

Gonzalez-Jimenez and Alba-Castro [20] improved elastic bunch graph matching by locating salient points in face images by means of the ridges and valleys operators. The shape matching algorithm is used to solve the correspondence problem between two face images. The comparison between shape-matched jets takes the shape and texture into account for face authentication.

2.2.5 Genetic Algorithm

Genetic algorithm (GA) is a searching method that can be used in Gabor feature selection. Campbell and Thomas [78] proposed to select Gabor features for pixel classification. A population of randomly selected combinations of features is created, each of which is considered a possible solution to the feature selection problem. Polzleitner [108] suggested selecting Gabor features for defect detection on a wooden surface. Wang and Qi [114] used GA to obtain the optimal 2-D Gabor wavelet basis derived from different GWs to represent a face image. However, the computation cost of GA is very expensive; especially when a huge number of features are available. As a result, GA may not efficient for the face recognition.

Li and Xu [28] and Gökberk et al. [5] proposed to find the optimal Gabor features using the GA in the training stage. These algorithms use the classification accuracy in their fitness functions. Li and Xu [28] used the Hybrid Genetic algorithms-based (HGAsb), which is an improved version of GA, to select the optimal Gabor kernel's scales and orientations for face classification. The feature locations are based on the fiducial points. On the other hand, Gökberk et al. [5] used the GA to select the optimal locations in face images for face classification. Then, a sequential floating forward search is used to select the Gabor kernel's scales and orientations. Nanni and Maio [58] used the GA to obtain the weighting for each local region of the faces in order to alleviate the effect of facial expression.

2.2.6 AdaBoost Gabor Features

[53], [70], [71], [86] proposed to use AdaBoost algorithm for face recognition by introducing the intra-face and extra-face differences in the Gabor feature space. The AdaBoost algorithm [85] is based on the idea that a “strong classifier” can be created by linearly combining a number of “weak classifiers”. A small set of “weak classifiers” from the original high-dimensional Gabor feature space is selected for face recognition. The two Gabor feature difference sets, intrapersonal difference and extrapersonal difference, are used for training in the AdaBoost algorithm, and the detail is described in Figure 2.11.

```
1.  Input: $N$ Training samples $(x_i, y_i)$, $i$=1, 2,…, $N$ with $m$ positive ($y_i$=1) and $l$ negative ($y_i$=0) samples
2.  Initialization: weights
        $w_{1,i} = \begin{cases} \frac{1}{2m}, \text{if } i \text{ is a positive sample} \\ \frac{1}{2l}, \text{if } i \text{ is a negative sample} \end{cases}$
3.  For $t$=1, …, $T$
        a.  Normalize all weights
        b.  For each feature $j$, train a weak classifier $h_j$ with error  $\varepsilon_j = \sum_i w_{t,i} \left| h_j(x_i) - y_i \right|$
        c.  Choose $h_t$ with lowest error $\varepsilon_t$
        d.  Update weights:  $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$  with  $e_i = \begin{cases} 1: x_i \text{ correctly classified} \\ 0: \text{otherwise} \end{cases}$  and  $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$
4.  Final strong classifier:  $H(x) = \begin{cases} 1 \text{ if } \sum_{t=1}^{T} \alpha_t h_t(x) > \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 \quad \text{otherwise} \end{cases}$  with  $\alpha_t = \mathbf{log}\left(\frac{1}{\beta_t}\right)$
```

Figure 2.11: AdaBoost learning algorithm

Shen and Bai [55], [56] used the mutual information to further eliminate redundancy among Gabor features for feature selection using the AdaBoost algorithm.

2.2.7 Other Gabor Feature Selection Methods

Arca et al. [92] presented a feature-based approach by extracting the Gabor features of 24 facial fiducial points. Kalocsai et al. [82] proposed to extract the features at 48 facial fiducial points by using 40 different Gabor filters. The weight of each Gabor feature is devised according to its discriminative ability by using a statistical analysis.

Ayinde and Yang [79] used rank correlation of the Gabor features for face recognition. A set of Gabor filters is applied on a facial image. The resulting filtered image and the original image are used to compute the representation of the image. A selected set of Gabor filters is then used to fine tune this technique.

Alterson and Spetsakis [87] presented an adaptive-sampling algorithm for spectral signature generation by using a grid sampling method. The sample points are selected in order to increase inter-object differentiation.

Gong et al. [19] designed a family of directional block partitions to compute the block-level directional projections of the classical Gabor features.

Then, the mean kernel and the variance kernel are used to extract the statistical characteristics of those block-level directional projections for face recognition.

Liu et al. [21] proposed a method to determine the optimal position for extracting the Gabor feature. The sampling positions are represented by a mask, which is generated by means of PCA from a set of training images. A certain percentage of the points with the largest magnitude are then selected. Finally, LDA is used to extract the optimal discriminant vectors for face recognition.

Chapter 3 : Recovering the 3-D Shape and Poses of Face Images Based on the Similarity Transform

3.1 Introduction

To construct the 3-D face model, the estimation of the face structure is an important step. The 2-D information can be easy to capture in the frontal view, but the depth information of the face is not easy to recovery from 2-D digital camera images.

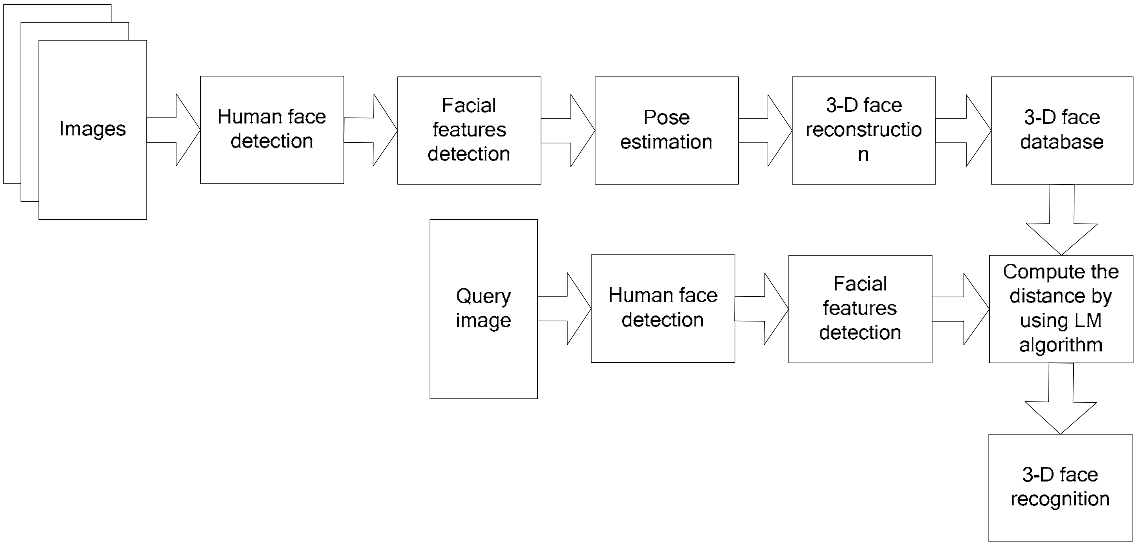


Figure 3.1: Structure for 3-D face recognition

Figure 3.1 introduces the procedure of the face modeling from a group of face images. The first step is to detect the human faces in these face images [85], [126], [127]. The face regions are detected in these images, which may have a simple or a complex background. However, this is a challenging task because the human face is highly variable. The detection performance can be affected by the presence of glasses, different skin color, gender, facial hair, facial expressions, etc.

After detecting the face regions in these face images, their facial features including the eye corners, nose tip and mouth corners have to be detected [128], [129], [130]. The locations of these facial features are the important information for 3-D face reconstruction which requires the correspondences among different face images.

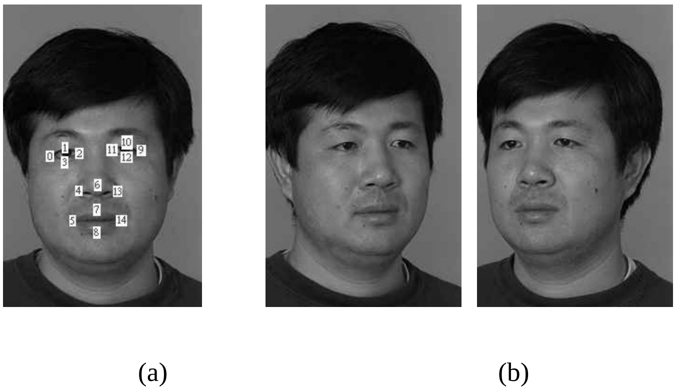
Based on the detected facial features in these images, the face mesh geometry and the head pose are determined in this stage. After that, the 3-D face model can be reconstructed and this face model can be used for 3-D face recognition.

In our proposed algorithm, three or more face images of the same subject under different poses are used to construct a 3-D face model. One of them is a frontal view, while the other images are under arbitrary poses. To recover the 3-D face structure, the 2-D frontal-view face image is adapted to the CANDIDE model. Then, the pose and the feature-point depths of the CANDIDE model are adjusted to fit the poses of the respective 2-D non-frontal-view face images in such a way that the feature-point distance between the projected 3-D model and the 2-D face images under different poses is minimized under the similarity transform [75]. However, searching for the best pose to provide the best alignment is so computationally intensive that an exhaustive search is impossible. Thus, the genetic algorithm (GA) is employed to search the optimal poses and depths of the feature points of the face model, which are computed iteratively so as to fit the face images accurately and efficiently. In addition, our method does not need any camera calibration. However, it requires that all the face images be of the same facial expression, and assumes that the heads are under rigid motion.

The similarity transform is also used to measure the accuracy of the constructed 3-D face model. After constructing a face model, it can be compared to those training 2-D face images used in the construction by means of the similarity transform. The Levenberg-Marquardt method is used to optimize the alignment of the face model to the respective face images. If the structure of the constructed face model is similar to that of the face image, the distance will be small. This concept can be applied on 3-D face recognition. Since the face image is not the same subject as the face model, the similarity distance between the 3-D face model and this image is larger than the face image constructing this 3-D face model. In summary, our algorithm can construct the 3-D face model and estimate the poses of the respective face images, and in the meantime can provide a measurement of the accuracy of the model as well as a tool for 3-D face recognition.

3.2 3-D Face Model

To construct the 3-D face model, at least three images under different poses and with a neutral expression are required. Our 3-D face model is represented by n=15 feature points, as illustrated in Figure 3.2(a). These can be located automatically or manually. The facial features selected are the most important features in the human. Moreover, the detection methods of these facial features such as corner detection [128], [129], [130] have already been developed. Ullman [98] proved that four point correspondences over three views can yield a unique solution to motion and structure. Thus, three or more face images under different viewing angles are required to construct the 3-D face model in our algorithm. The first image in our experiments is a frontal view, and the poses of the other images are estimated with reference to the frontal-view image. The frontal-view face image provides the 2-D information of the human face by simply adapting the 3-D face model to the frontal-view face image. Other non-frontal-view face images are used to derive the depths of the selected facial feature points. Figure 3.2(b) shows two other images of the same person under different poses.



(a) (b)

Figure 3.2: (a) A frontal-view image with 15 landmark points, and (b) two more face images with different poses.

In our proposed method, the CANDIDE model [35] is employed as our face model. The CANDIDE model is only used for initialization in our iterative process because the 3-D face structure is unknown in the first iteration. The definitions of the three axes are shown in Figure 3.3. Based on the position of the important feature points, the CANDIDE model is first adapted to the frontal-view face image, as shown in Figure 3.4(a). Then, the CANDIDE model is rotated to the same poses of the non-frontal-view face images, and the depths of the feature points of the model are adjusted so that the feature points obtained by projecting the 3-D model onto the 2-D space can fit the corresponding feature points of the images accurately. Figure 3.4(b) illustrates the adaptation of the 3-D CANDIDE model to two other face images.

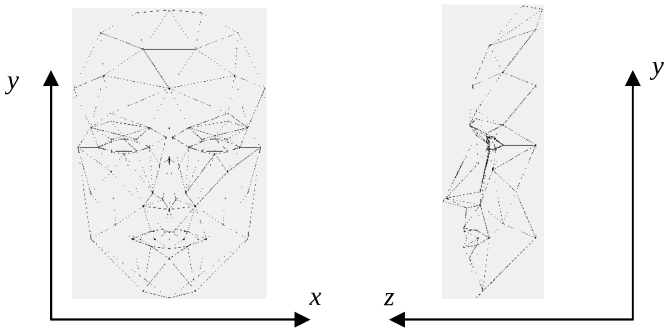
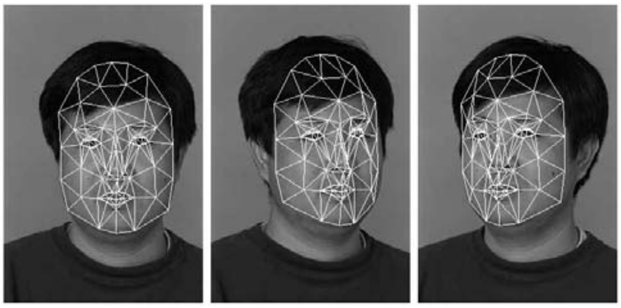


Figure 3.3: The CANDIDE model in frontal view and profile view.



(a) (b)

Figure 3.4:(a) Face images with an adapted face model, and (b) face images under different poses adapted by the rotated face model.

3.3 Our Algorithm

Our algorithm can recover the structure and poses of a face based on a number of 2-D images by projecting its 3-D model to the 2-D plane, i.e. a 2-D to 3-D problem. We assume that one frontal-view face image and N ($N \geq 2$) non-frontal-view face images are available. The poses and the scales of the non-frontal-view images with respect to the frontal-view face image are all unknown. We also assume that n feature points in the respective training images have all been located accurately. The 3-D to 2-D projection is performed by the following transformation:

$$\mathbf{p}_i = s_i \, \mathbf{R}_{i2\times3}\mathbf{C}+\mathbf{T}_i\,, \quad \text{for } i = 1\ldots N, \tag{3.1}$$

where N is the number of non-frontal-view face images, s_i is the scaling factor, $\mathbf{T}_i=[t_{i1}, t_{i2}]^{\mathrm{T}}$ represents the translation matrix and $\mathbf{R}_i$ denotes the rotation matrix representing the relative orientation between the frontal-view image and the i^{th} non-frontal-view face image. $\mathbf{R}_i$ can be specified as three successive rotations around the x-, y-, and z-axes, by angles ϕ_i, ψ_i and θ_i, respectively, and can be written as the product of these three rotations as follows:

$$\begin{aligned}\mathbf{R}_i &= \begin{bmatrix} \cos\phi_i & \sin\phi_i & 0 \\ -\sin\phi_i & \cos\phi_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\psi_i & 0 & -\sin\psi_i \\ 0 & 1 & 0 \\ \sin\psi_i & 0 & \cos\psi_i \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_i & \sin\theta_i \\ 0 & -\sin\theta_i & \cos\theta_i \end{bmatrix} \\ &= \begin{bmatrix} r_{i_{11}} & r_{i_{12}} & r_{i_{13}} \\ r_{i_{21}} & r_{i_{22}} & r_{i_{23}} \\ r_{i_{31}} & r_{i_{32}} & r_{i_{33}} \end{bmatrix}. \end{aligned} \tag{3.2}$$

$\mathbf{R}_{i2\times3}$ contains the first two rows of the 3×3 rotation matrix $\mathbf{R}_i$. Let n be the number of feature points in a face image. The matrix $\mathbf{C}$ (=$[\mathbf{X}_\mathbf{C}, \mathbf{Y}_\mathbf{C}, \mathbf{Z}_\mathbf{C}]^{\mathrm{T}}$) is a $3\times n$ matrix, which represents n 3-D coordinates in the adapted face model. $\mathbf{X}_\mathbf{C}$, $\mathbf{Y}_\mathbf{C}$ and $\mathbf{Z}_\mathbf{C}$ are three $n\times1$ matrices, which are the x, y and z-coordinates, respectively, of the feature points in the adapted face model. $\mathbf{X}_\mathbf{C}$ and $\mathbf{Y}_\mathbf{C}$ are measured from the image being adapted, while $\mathbf{Z}_\mathbf{C}$ is initially set at the default values of the CANDIDE model with a particular scale according to the size of the face image. $\mathbf{p}_i$ is a $2\times n$ matrix which represents n 2-D coordinates of the feature points in the i^{th} non-frontal-view face images. Also, the first row and the second row of $\mathbf{p}_i$ represent the *x*- and *y*-coordinates, respectively.

If the pose of the face model and the depths of the feature points fit the i^{th} non-frontal-view face images, the following equation will be a minimum:

$$\mathrm{D1}^2 = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{p}_i - s_i\,\mathbf{R}_{i\,2\times3}\mathbf{C} - \mathbf{T}_i\right\|^2 . \tag{3.3}$$

Before taking the norm of the difference between the face model and the images, we must remove the differences caused by irrelevant effects, such as the arbitrary image size under scaled orthography or the arbitrary location due to the translation and rotation of the face in the image. These irrelevant effects are useless when comparing the similarity between the 3-D face model and the 2-D face image. To remove the irrelevant effects, image alignment is performed. The alignment transformation, which is a series of transformations - including translation, scaling, and rotation - is applied to one image to obtain an optimal alignment to another image.

All the point sets to be compared are translated to their respective centroids so that the centroids become the origin of the coordinate system, and their first moments are zero. Let $\mathbf{M}$ $(=[\mathbf{X_M}, \mathbf{Y_M}, \mathbf{Z_M}]^{\mathrm{T}})$ be a $3\times n$ matrix which represents *n* centered 3-D model point set. Similarly, suppose that $\mathbf{q}_i$ denotes a $2\times n$ matrix which represents *n* centered 2-D point sets of the i^{th} image. In other words, $\mathbf{q}_i$ and $\mathbf{M}$ are the centered point sets of $\mathbf{p}_i$ and $\mathbf{C}$, respectively, and (3.3) becomes:

$$\mathrm{D2}^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left\|\mathbf{q}_i - s_i\,\mathbf{R}_{i\,2\times 3}\,\mathbf{M}\right\|^2 . \tag{3.4}$$

To accomplish the optimal alignment, we employ the genetic algorithm (GA) to search for the optimal solution, i.e. the optimal poses of the non-frontal-view images. GA is used because it can search for the optimal solution even in a large searching space. This approach can provide an accurate solution, although the computational time is a little bit longer. However, for many applications, such as face recognition, this 3-D face reconstruction can be performed offline.

3.3.1 *The Chromosome*

In the GA, the relative poses of the adapted face model to the respective non-frontal-view face images are randomly generated and evenly distributed to form the initial population. The fitness value of each candidate in a population is measured based on (3.4). When the population evolves, the number of candidates with the correct poses will dominate gradually. The iterative process will be stopped either when the fitness value of the population does not change significantly over a number of iterations or when a certain number of iterations have been done. Finally, the parameters of the best candidate in the population are used to represent the best poses of the face model to the non-frontal-view face images.

The chromosome designed for the GA should be able to represent the solution effectively, and its length should be as short as possible. Figure 3.5 illustrates the chromosome structure used in our algorithm for having *N* non-frontal-view face images, where ϕ_i, ψ_i and θ_i are the angles rotated about the *z*-, *y*- and *x*-axes, respectively, for adapting the face model to the i^{th} face images. In our approach, the number of elements in the chromosome is therefore 3*N*.

Chromosome: pose parameters

θ_1	ψ_1	ϕ_1	θ_2	ψ_2	ϕ_2	$\ldots$	θ_N	ψ_N	ϕ_N

Figure 3.5: Structure of a chromosome with *N* non-frontal view training face images.

When the number of face images used to construct the face model increases, the chromosome size will also increase. Then, increasing the population size and the maximum iteration numbers are also required because the chromosomes will form a much larger solution space.

3.3.2 *The Optimal Depths of the Feature Points*

To minimize the fitness function in the GA, the information provided from the chromosomes is insufficient because the depths of the feature points are unknown. The following equation shows the feature-point distance between the i^{th} face image and the projected feature points of the face model:

$$\mathrm{D3}_i^{\,2} = \left\| \mathbf{q}_i - s_i\, \mathbf{R}_{i\,2\times3}\, \mathbf{M} \right\|^2 \qquad i = 1\ldots N. \tag{3.5}$$

In (3.5), the scaling factor and the poses between the 3-D face model and the non-frontal-view face image are unknown. To compute the distance, ϕ_i, ψ_i and θ_i are substituted into (3.2) to calculate $\mathbf{R}_i$, and then $\mathbf{R}_{i2\times3}$ is obtained and substituted into (3.5). As a result, N equations can be formed. Since the depths of the features points are the default values of the CANDIDE model in the first iteration, the initial structure of the face model is an approximation only. Therefore, the z-coordinates in $\mathbf{M}$ are calculated by applying partial differentiation to (3.5) with respect to the z-coordinates. From (3.5), we can calculate N different z-coordinates for $\mathbf{M}$. There are N different combinations between the frontal view image and each of the N non-frontal view images. Let $\mathbf{Z}_{\mathbf{M}i}$ be the $n\times1$ matrix which represents the z-coordinates in $\mathbf{M}$ constructed based on the frontal view image and the i^{th} non-frontal view image. We also denote $\mathrm{r1}_i = \lfloor r_{i_{13}} \quad r_{i_{23}} \rfloor$, $\mathrm{r2}_i = \lfloor r_{i_{31}} \quad r_{i_{32}} \rfloor$, $\mathrm{r3}_i = \lfloor r_{i_{11}} \quad r_{i_{12}} \rfloor$, $\mathrm{r4}_i = \lfloor r_{i_{21}} \quad r_{i_{22}} \rfloor$ and $\mathbf{M}_{xy}$ (=$[\mathbf{X_M}, \mathbf{Y_M}]^T$). Then, by applying partial differentiation to (3.5) with respect to $\mathbf{Z}_{\mathbf{M}i}$, we have

$$\mathrm{Z}_{\mathrm{M}i}^{\;T} = \frac{\mathrm{r1}_i \cdot \mathrm{q}_i + s_i \cdot r_{i_{33}} \cdot \mathrm{r2}_i \cdot \mathrm{M}_{xy}}{s_i \cdot \mathrm{r1}_i \cdot \mathrm{r1}_i^{\,T}}, \qquad i = 1\ldots N. \tag{3.6}$$

Then, (3.6) is substituted into (3.5) and partial differentiation with respect to s_i

is applied to (3.5). Denote $\mathrm{a}_i = \mathrm{r3}_i \cdot \mathrm{M}_{xy} + r_{i_{13}} \cdot \frac{r_{i_{33}} \cdot \mathrm{r2}_i \cdot \mathrm{M}_{xy}}{\mathrm{r1}_i \cdot \mathrm{r1}_i^T}$ and $\mathrm{b}_i = \mathrm{r4}_i \cdot \mathrm{M}_{xy} + r_{i_{23}} \cdot \frac{r_{i_{33}} \cdot \mathrm{r2}_i \cdot \mathrm{M}_{xy}}{\mathrm{r1}_i \cdot \mathrm{r1}_i^T}$, which are both 1×*n* matrices. Then, we have

$$s_i = \frac{tr\left[\mathrm{q}_i \cdot \begin{bmatrix} \mathrm{a}_i \\ \mathrm{b}_i \end{bmatrix}^T\right]}{\mathrm{a}_i \cdot \mathrm{a}_i^T + \mathrm{b}_i \cdot \mathrm{b}_i^T}, \tag{3.7}$$

where *tr*[] denotes the trace, which is the sum of the diagonal elements in a matrix. As a result, the scaling factor between the adapted face model and the i^{th} non-frontal-view images can be computed

From (3.6), there are *N* different *z*-coordinates for the face model. To find the optimal depths of the feature points in the face model, the *z*-coordinates in **M** are calculated by applying partial differentiation to (3.4) rather than (3.5), with respect to $\mathbf{Z_M}$:

$$\mathrm{Z_M}^T = \frac{\sum_{i=1}^{n} \left(s_i \cdot \mathrm{r1}_i \cdot \mathrm{q}_i + s_i^2 \cdot r_{i_{33}} \cdot \mathrm{r2}_i \cdot \mathrm{M}_{xy} \right)}{\sum_{i=1}^{n} s_i^2 \cdot \mathrm{r1}_i \cdot \mathrm{r1}_i^T}, \tag{3.8}$$

where the respective s_i are calculated using (3.7). Then this set of new *z*-coordinates replaces the original one. Therefore, the optimal depths derived from one non-frontal-view face image are replaced by the optimal depths derived from a group of non-frontal-view face images. The proof of (3.6), (3.7) and (3.8) has been included in Appendix 1.

To calculate the fitness of a chromosome, we first substitute its values, i.e.

ϕ_i, ψ_i and θ_i, to (3.2) in order to calculate $\mathbf{R}_i$. Then, the corresponding scaling factor s_i is computed using (3.6) and (3.7), and the depths of the feature points in the adapted face model are calculated using (3.8). After that, its fitness can be calculated by substituting all the above parameters into (3.4), which consider the fitness to all the training face images. Finally, after the GA, the scaling factors and the poses of the non-frontal-view face images with respect to the 3-D face model, and the depths of the feature points of the face model which minimizing the similarity distance can be obtained.

3.3.3 *The Genetic Operators*

Having defined the chromosome and the fitness function, the genetic operators - selection, crossover, and mutation [18] - which are performed to search the optimal poses of the face images and the optimal depths of the face model are described in this section. In our algorithm, the rank selection method is used to select two chromosomes to perform crossover and/or mutation. Rank selection first ranks the population and then every chromosome receives fitness from this ranking. The worst will have fitness 1, second worst 2 etc. and the best will have fitness N_c (number of chromosomes in population). After selecting two chromosomes, two crossover points are selected randomly. The elements between these two crossover points in the two chromosomes are exchanged to form a pair of new offspring. In the GA, not all the selected chromosomes are performed crossover. If there is no crossover, offspring is exact copy of parents. The aim of crossover is that new chromosomes have good parts of old chromosomes and the new chromosomes may be better. However, it is good to leave some part of population survive to the next generation. Figure 3.6 illustrates the crossover operation.

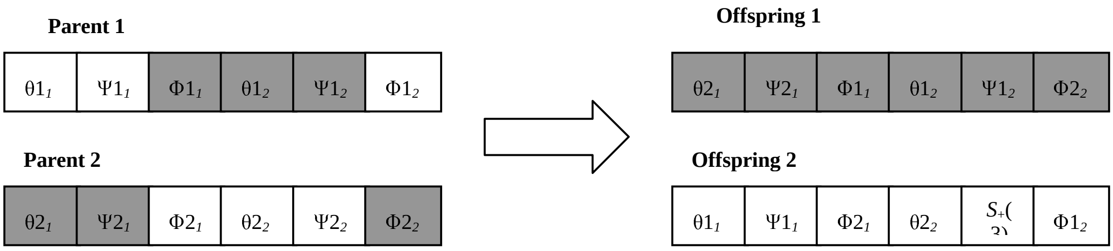


Figure 3.6: An example of the crossover operation.

Mutation is intended to prevent all the solutions in a population falling into a local minimum by exploiting new candidates randomly. In our algorithm, the number of elements in a chromosome being mutated depends on the number of training face images. The *N* elements in each chromosome are randomly selected and replaced by *N* randomly generated numbers, where *N* is the number of non-frontal-view face images.

3.4 The Similarity Measure

After constructing the face model of a person, it can be adapted to any face image. If the face model is constructed from a particular subject, the feature-point distance between this face model and a face image of this particular subject should be smaller than that of another subject. Unlike other face model construction algorithms [1], [131] our algorithm can evaluate the accuracy of the constructed 3-D face model. The accuracy of the 3-D face model can be determined by measuring the feature-point distance between the face model and the respective training face image. This is especially useful since we usually do not have the exact data of the 3-D face structure. Therefore, a measurement to determine the accuracy of the constructed 3-D face model is necessary. This distance can also be applied to human face recognition because the similarity distance between the 3-D face model and the image with different subject is larger than the face image constructing this 3-D face model.

To compute the feature-point distance between the 3-D face model and a 2-D face image, the Levenberg-Marquardt method [24], [46] is used to optimize the following equation

$$\mathrm{D}^2 = \min_{s,\mathrm{R}_{2\times3}} \frac{1}{n}\left\|\mathrm{u} - s\mathrm{R}_{2\times3}\mathrm{M}\right\|^2, \tag{3.9}$$

where $\mathbf{u}$ is the $2\times n$ matrix representing the centred (x, y) coordinates of the feature points in a test face image, and $\mathbf{R}_{2\times3}$ and s are the rotation matrix and scaling factor between the 3-D face model and the 2-D testing image, respectively, that can minimize the above equation. $\mathbf{R}_{2\times3}$ contains the first two rows of the 3×3 rotation matrix **R** that can be specified as the three successive rotations around the *x*-, *y*-, and *z*-axes, by an angle of ϕ, ψ and θ, respectively. This matrix can be written as the product of these three rotations by using (3.2). Having constructed the 3-D face model of a face subject, the depths of the feature points are known. Hence, when a 2-D face image is compared to the 3-D face model for face recognition, a simpler optimization method, the Levenberg-Marquardt method instead of the GA, can be used to estimate the pose and scale of the query image.

For face recognition, (3.9) is used to compare the similarity distance between the query or test face image and different face models in a 3-D face database which is develop by constructing a group of 3-D face models by the method described in section 3.3. The face model that results in the minimum feature-point distances should have the best representation of the query face image. However, not all the *n* feature points in the query face images are visible, because the query image may have an arbitrary pose. As a result, some modifications have to be made to (3.9). First, the columns of **M** corresponding to the invisible feature points are removed. Then, *n* is replaced by the number of visible feature points in the face image. Therefore, only the visible facial features in the test image are used to compare with the 3-D face models. As a result, the testing face images with large pose variations can also be considered as the query images for face recognition. Experiments in the next section will show the validity of using the 3-D face model for face recognition.

3.5 Experimental Results

3.5.1 *Pose Estimation*

In order to check the accuracy of the pose estimation in our proposed algorithm, the face database with known pose variation has been constructed. The face images were captured by using CASIO EX-Z55 digital camera and the camera was placed about 3 meters from the subjects. When capturing the face images in different directions, the subjects remained their neutral facial expressions. A several photos have been taken for each subject under different pose variations. The face only rotated about the *y*-axis because the measurement of the face rotated about this direction is simple and the validation of the pose estimation can be more reliable.

One of the subjects under different poses is shown in Figure 3.7.

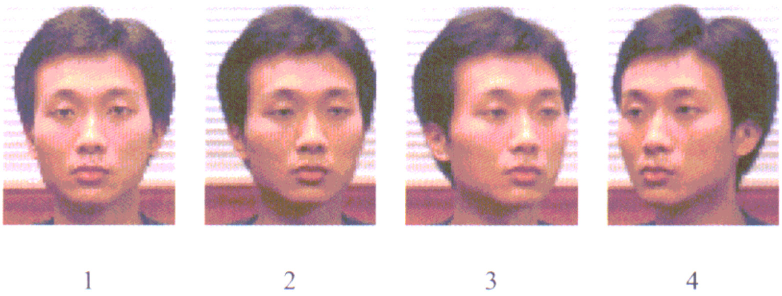


Figure 3.7: Face images under different poses of the same subject

	Actual angle rotated around		
Face image indices	*x*-axis	*y*-axis	*z*-axis
1	0	0	0
2	0	-10	0
3	0	-20	0
4	0	20	0

Table 3.1: The actual poses of the face images in Figure 3.7.

	Estimated angle rotated around		
Face image indices	*x*-axis	*y*-axis	*z*-axis
1	0	0	0
2	-2	-9	0
3	-3	-19	-1
4	1	20	0

Table 3.2: The estimated poses of the face images in Figure 3.7.

Compared to Tables 3.1 and 3.2, we can see that the estimated poses were almost the same as the actual poses of the face images. These results show that our proposed pose estimation algorithm is accurate in acceptable rate. Table 3.2 shows that the face has been rotated about the *x*- and *z*-axes. However, this is acceptable because it is difficult to determine the human face remaining 0° about these axes.

3.5.2 *3-D Face Model Construction*

In the following experiment, a subset of the FERET database [81] is selected for our experiments. This is a standard database for face recognition evaluation, which contains images in various poses. To construct different face models, 60 frontal face images, corresponding to 60 distinct subjects, were selected in our experiment: 13 of the subjects have 4 non-frontal-view face images, 12 have 3 non-frontal-view face images, and the remaining 35 have 2 non-frontal-view face images only. All the 15 feature points are visible in the selected non-frontal-view face images. However, in the face recognition experiment, face images with larger pose variations can be selected because not all the 15 feature points are required for the matching between a test image and the face model. In addition, the 15 feature points were selected manually in our experiments so that potential errors in the detection of the facial feature points can be eliminated.

In this experiment, different numbers of face images of the same subject under different poses may be used to construct a face model. For example, suppose that one frontal face image and four images under different poses are available. Then, six different face models can be constructed when two of the different non-frontal-view face images and one frontal-view face image are considered in the construction. Similarly, there are four different models when three of the non-frontal-view images are used and only one when all the four non-frontal-view images are considered. Therefore, 11 different face models can be constructed.

To construct the face models, all the point sets of the training images were translated to their respective centroids. Then, the best poses of the face models were aligned using the GA. The ranges of the elements in the chromosomes, i.e. ϕ_i, ψ_i and θ_i, were set between -50° and 50°; this allows all 15 feature points to be visible after 2-D projection. Table 3.3 shows the population size and the maximum number of iterations for face model construction using different numbers of images under different poses. The maximum runtime required to generate a face model is about 1.8s using 3 face images under different poses. This runtime is measured with a Pentium IV computer system with 2.3GHz and 512MB RAM. The crossover rate and the mutation rate were set at 80% and 20%, respectively.

Number of face images	Population size	Maximum iterations	Maximum runtime per model
3	800	200	1.8s
4	1200	300	2.6s
5	1500	400	4.0s

Table 3.3: The parameters of the GA under different numbers of face images.

Figures 3.8, 3.9 and 3.10 show the face images of three different subjects that were used to construct their face models. The left-most image is the reference image, i.e. the frontal-view image, while the other images are under different poses. In Figure 3.8, five images are available, so at most 11 different face models can be constructed by different combinations of the non-frontal-view face images; while in Figures 3.9 and 3.10, four images are available, so at most 4 different face models can be constructed. To illustrate the estimation of the poses using different combinations of the front-view image and non-frontal-view images for training sets from Example 1, Example 2 and Example 3, Table 3.4 tabulates the corresponding indices of the face models determined based on the different combinations of the images.

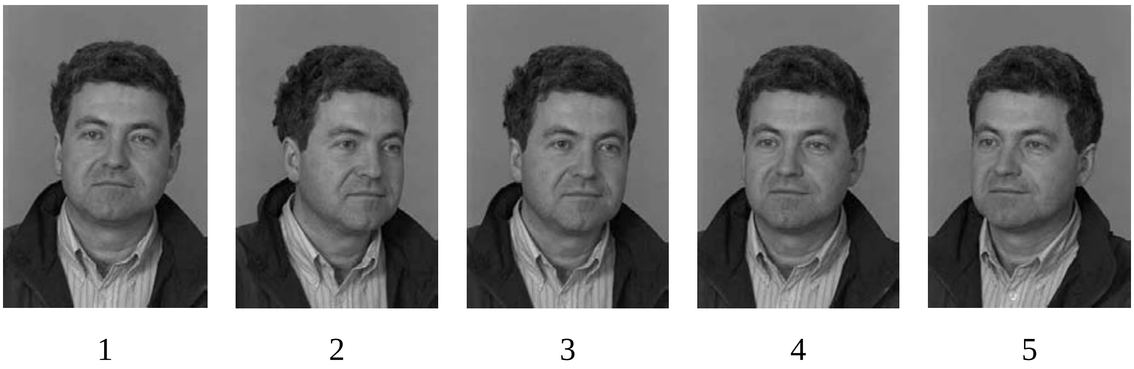
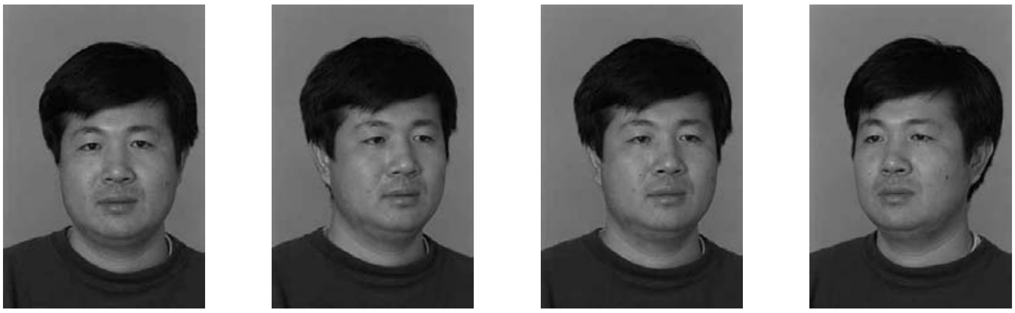
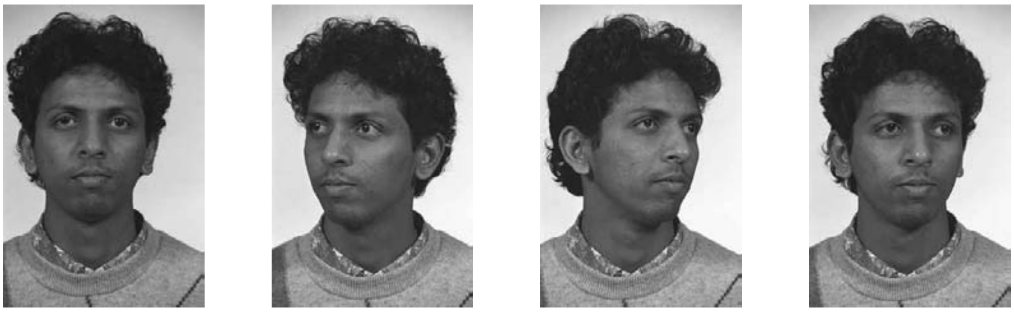


Figure 3.8: Example 1 - five face images under different poses used to construct the face model.



1 2 3 4

Figure 3.9: Example 2 - four face images under different poses used to construct the face model.



1 2 3 4

Figure 3.10: Example 3 - four face images under different poses used to construct the face model.

	Face Model Indices for Example 1
Model 1	Images 1, 2, 3
Model 2	Images 1, 2, 4
Model 3	Images 1, 2, 5
Model 4	Images 1, 3, 4
Model 5	Images 1, 3, 5
Model 6	Images 1, 4, 5
Model 7	Images 1, 2, 3, 4
Model 8	Images 1, 2, 3, 5
Model 9	Images 1, 2, 4, 5
Model 10	Images 1, 3, 4, 5
Model 11	Images 1, 2, 3, 4, 5

(a)

	Face Model Indices for Example 2 and Example 3
Model 1	Images 1, 2, 3
Model 2	Images 1, 2, 4
Model 3	Images 1, 3, 4
Model 4	Images 1, 2, 3, 4

(b)

Table 3.4: Indices of the face models for (a) Example 1 and (b) Example 2.

Tables 3.5, 3.6 and 3.7 tabulate the best poses of the adapted face models to the respective non-frontal-view face images from Example 1, Example 2 and Example 3 respectively. The entries in these tables show the angles of the non-frontal-view face images about the *x*-, *y*- and *z*-axes. It can be observed that the estimated poses of the different models for the same face image are consistent. These results show that almost the same face models of same subject can be constructed by using different combinations of the non-frontal-view face images. Figure 3.11 shows the variance of the poses for 13 distinct subjects. The face images rotated around the *y*-axis result in the largest variance as the faces are mainly rotated around this axis.

	Poses of image 2			Poses of image 3			Poses of image 4			Poses of image 5		
	tile (degs)	pan (degs)	roll (degs)	tile (degs)	pan (degs)	roll (degs)	tile (degs)	pan (degs)	roll (degs)	tile (degs)	pan (degs)	roll (degs)
Model 1	0	-16	1	1	-10	5						
Model 2	0	-22	3				2	11	8			
Model 3	0	-24	3							2	24	5
Model 4				1	-14	8	0	16	8			
Model 5				1	-15	7				1	31	9
Model 6							2	13	8	2	25	6
Model 7	0	-22	3	1	-14	8	2	11	9			
Model 8	0	-24	3	2	-16	10				2	25	6
Model 9	0	-23	4				2	11	7	2	24	6
Model 10				3	-14	8	3	9	4	3	24	5
Model 11	0	-23	3	2	-13	8	2	13	8	2	25	6

Table 3.5: The best estimated poses of the non-frontal-view images from Example 1.

	Poses of image 2			Poses of image 3			Poses of image 4		
	tile (degs)	pan (degs)	roll (degs)	tile (degs)	pan (degs)	roll (degs)	tile (degs)	pan (degs)	roll (degs)
Model 1	-1	-20	-5	-1	-15	-6			
Model 2	-1	-18	-5				-1	16	-4
Model 3				-1	-14	-6	-1	15	-4
Model 4	-1	-19	-4	-1	-15	-5	-1	15	-4

Table 3.6: The best estimated poses of the non-frontal-view images from Example 2

	Poses of image 2			Poses of image 3			Poses of image 4		
	tile (degs)	pan (degs)	roll (degs)	tile (degs)	pan (degs)	roll (degs)	tile (degs)	pan (degs)	roll (degs)
Model 1	0	27	-2	-6	-42	-7			
Model 2	0	30	1				0	-14	2
Model 3				-4	-35	-3	0	-12	1
Model 4	0	29	1	-6	-40	-5	1	-14	6

Table 3.7: The best estimated poses of the non-frontal-view images from Example 3

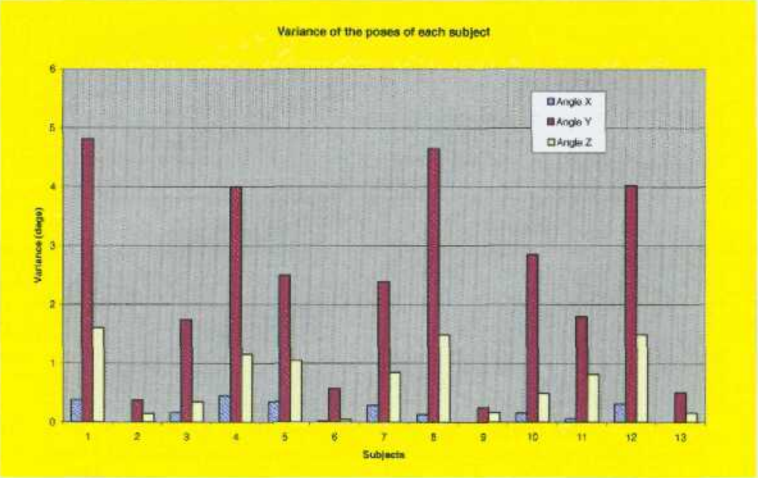


Figure 3.11: Variance of the poses for 13 distinct subjects.

Tables 3.8, 3.9 and 3.10 tabulate the structure of the face models from Example 1, Example 2 and Example 3 constructed using all the available training face images, respectively. Each of the columns in these tables shows the (x, y, z) coordinates of the corresponding feature point, which have been defined as shown in Figure 3.4(a). Figure 3.12, Figure 3.13 and Figure 3.14 show the means and the standard deviations of the depths of the feature points from different face models in Example 1, Example 2 and Example 3, respectively.

	Indices of the feature points (pixels)														
Example 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x	98	125	111	112	112	121	140	137	137	190	163	163	177	174	157
y	164	165	227	159	168	202	197	223	230	171	170	229	164	175	203
z	0	7	14	6	10	21	45	29	33	3	4	15	6	6	24

Table 3.8: The structure of the face model constructed from five images in Example 1.

	Indices of the feature points (pixels)														
Example 2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x	98	124	120	111	113	126	145	144	144	186	162	168	175	174	162
y	180	178	249	174	184	223	212	241	259	176	177	248	171	181	223
z	0	3	2	9	5	12	31	24	25	7	8	8	8	9	21

Table 3.9: The structure of the face model constructed from four images in Example 2.

	Indices of the feature points (pixels)														
Example 3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x	65	92	89	79	80	95	115	113	112	163	134	135	151	150	129
y	150	147	215	141	154	189	181	208	228	151	150	215	142	156	194
z	0	8	16	8	8	22	46	28	29	4	10	20	10	11	24

Table 3.10: The structure of the face model constructed from four images in Example 3.

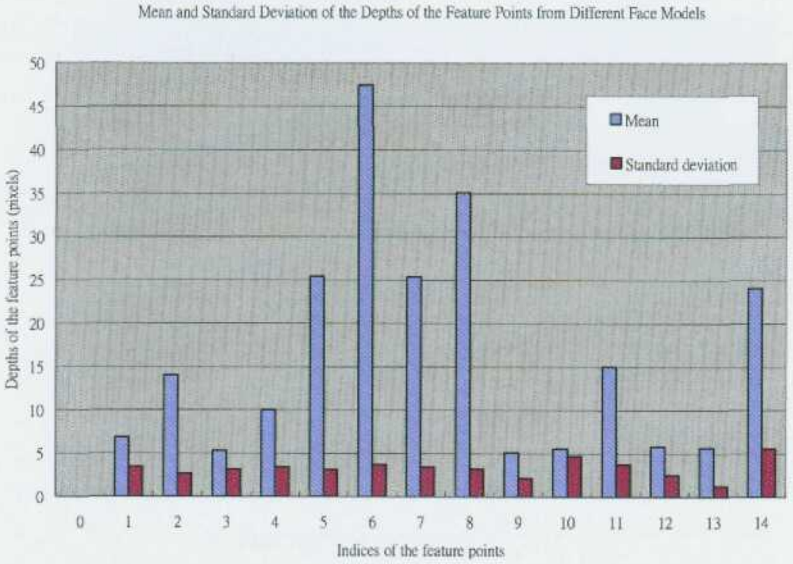


Figure 3.12: The mean and standard deviation of the depths of the feature points from different face models in Example 1.

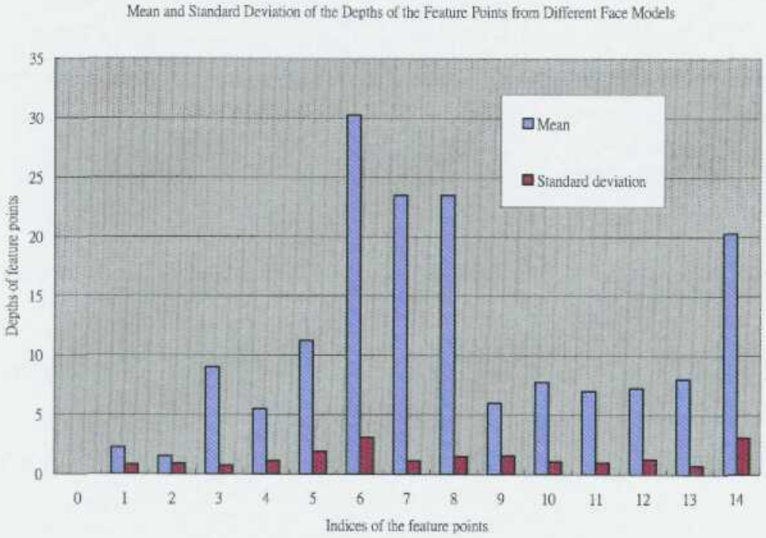


Figure 3.13: The mean and standard deviation of the depths of the feature points from different face models in Example 2.

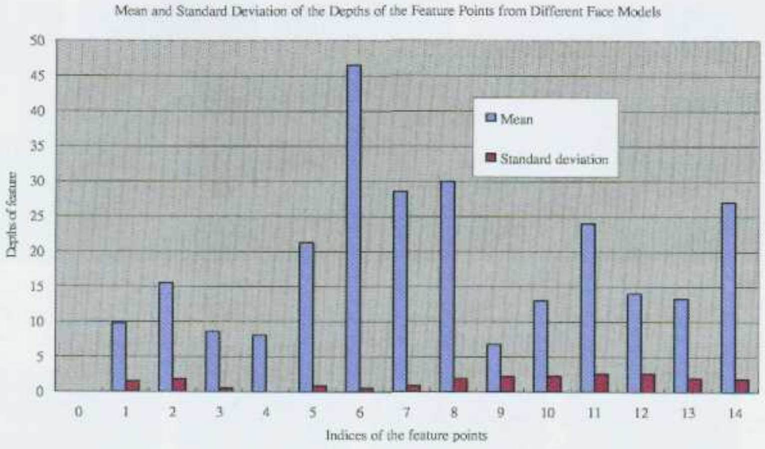


Figure 3.14: The mean and standard deviation of the depths of the feature points from different face models in Example 2.

From Tables 3.8, 3.9 and 3.10, we see that feature point 7 has the largest *z*-coordinate value, as this point represents the nose tip, which is the outermost part in a face. In addition, feature points 8 and 9 have larger *z*-coordinate values than other feature points because these two points represent the lips, which protrude more than all other feature points except the nose tip. Feature points 1 to 6 have very similar z-coordinate values to feature points 10 to 15 since the structure of a face is usually quite symmetrical. These show that the structure of the constructed face models conforms to the structure of the human faces. Figures 3.15, 3.16 and 3.17 show the models adapted to the non-frontal-view face images in Example 1, Example 2 and Example 3 after the optimal poses of the face images and their 3-D face models have been determined.

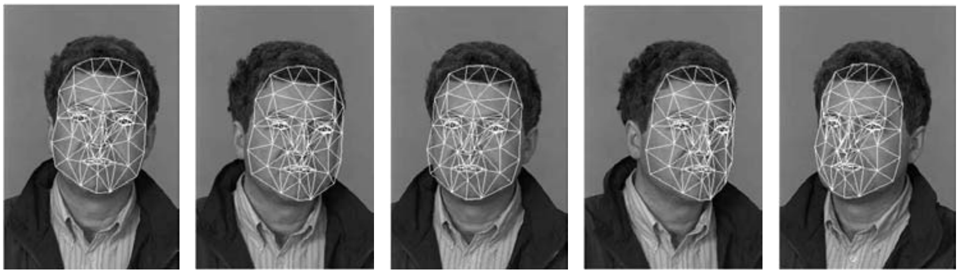


Figure 3.15: Adaptation of face model to the non-frontal-view face images of Example 1.

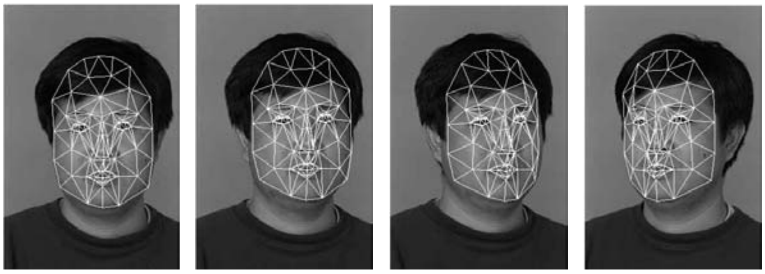


Figure 3.16: Adaptation of face model to the non-frontal-view face images of Example 2.

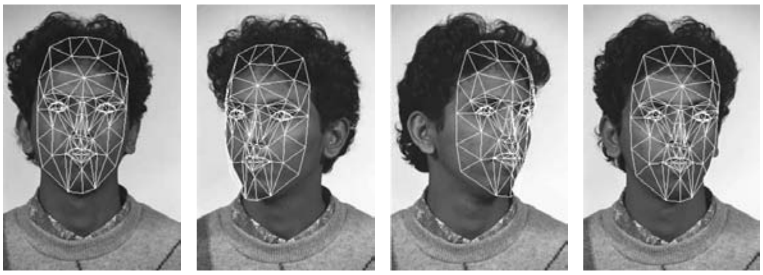


Figure 3.17: Adaptation of face model to the non-frontal-view face images of Example 3.

3.5.3 *Evaluation of the Accuracy of the Face Models*

To evaluate the accuracy of the constructed 3-D face models, the similarity transform described in Section 3.4 was used. With the various face models generated using Example 1, Example 2 and Example 3, the smallest distances between a number of face images and the respective face models are measured, as shown in Figures 18, 3.19 and 3.20. Some of the test images are of the same person as the face model, while the others are of other subjects. The similarity distances between the face images and the face models of the same subject are small when compared to those between the face images and the face models of different subjects. In addition, the similarity distances between the face images and the face models of the same subject are similar irrespective of the training images used to construct the face models. Therefore, this method can also be used as a face recognition algorithm, which can alleviate the effect of perspective variations.

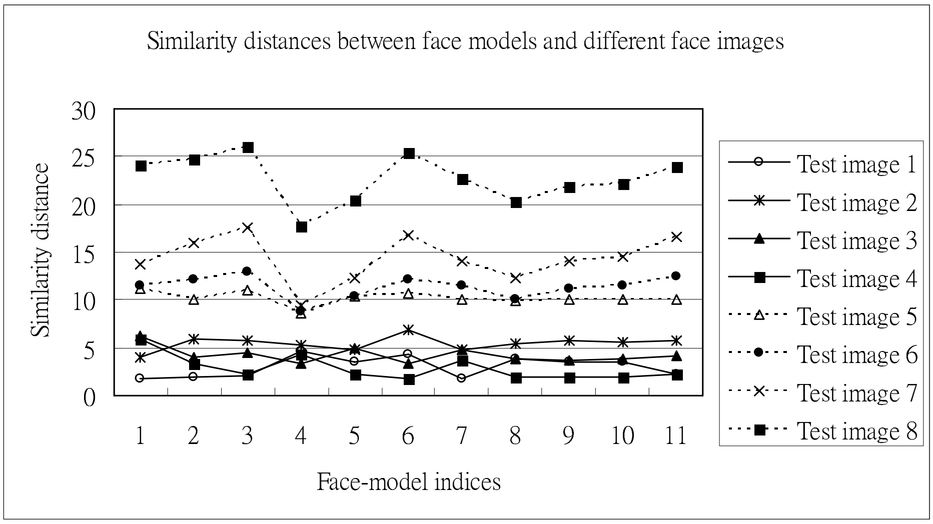


Figure 3.18: The similarity distances between a number of test images and each of the face models generated from Example 1. Test images 1 to 4 are the same subject as the face model, while the others are different subjects to the face model.

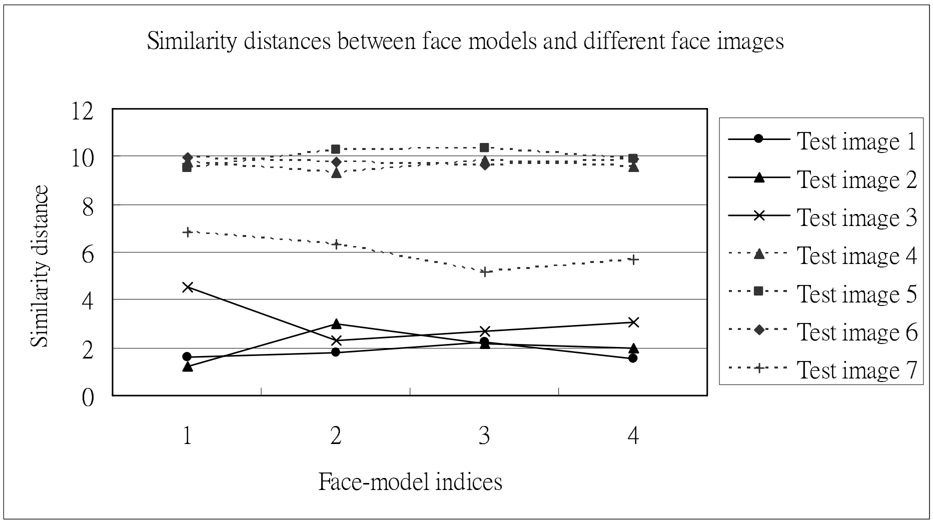


Figure 3.19: The similarity distance between a number of images and each of the face models generated from Example 2. Test images 1 to 3 are the same subject as the face model, while the others are different subjects to the face model.

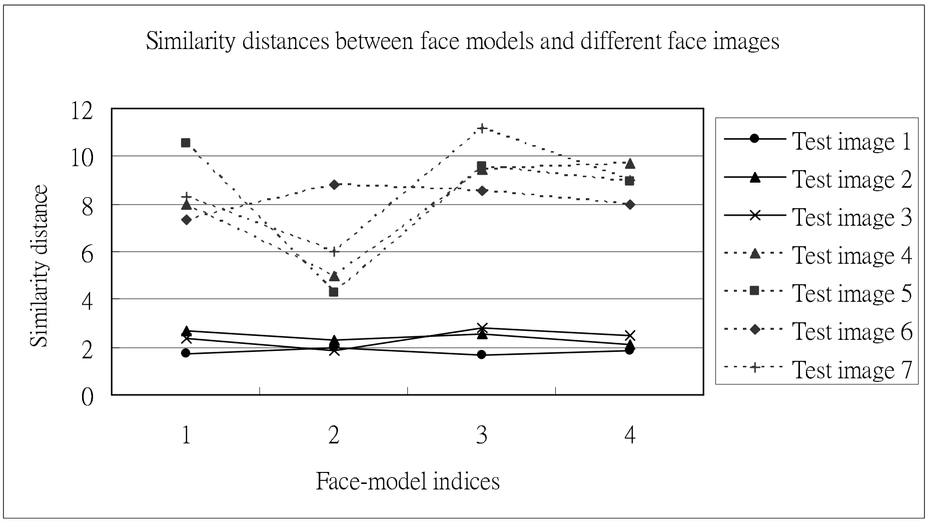


Figure 3.20: The similarity distance between a number of images and each of the face models generated from Example 3. Test images 1 to 3 are the same subject as the face model, while the others are different subjects to the face model.

3.5.4 *Face Recognition Using the 3-D Face Models*

After the face models for each subject have been constructed, the feature-point distance can also be used for face recognition. In this experiment, each face model was constructed using 3 different face images under different poses (one of which is frontal-view). Each distinct subject is represented by a corresponding 3-D face model, so 180 face images were used to construct 60 distinct face models.

To perform face recognition, other face images which have not been used to construct the face models are used as testing face images, and are compared to the different face models using the similarity transform. The face images used to construct the face models are the training images for PCA and LDA, while other face images are used as testing images. If the similarity distance between a face model and a testing face image is a minimum, this face image will then be classified as the subject of the face model. Therefore, for each testing face image, its similarity distances to all the face models are computed, and the faces are listed in ascending order according to these distances. As described in Section 3.4, not all the feature points are needed to calculate the feature-point distance. Consequently, face images with large pose variations can be recognized, even though not all the feature points are visible in these face images.

In this experiment, 72 testing face images of 28 different subjects were selected. All these subjects have their own face models stored in the face model database. These images are divided into two sets. The first set includes those face images under large pose variations in which the absolute angle rotated around the *y*-axis is larger than 50°. The second one contains those face images under small pose variations in which the absolute angle rotated around the *y*-axis is smaller than 50°. There are 45 testing face images in the first set, while the remaining face images are in the other set.

Figure 3.21 shows the recognition rates when the correct face models of the testing images are in the first top *k* of the list according to the similarity distances, where *k* =1, …, 10. Our method is also compared to two other face recognition techniques: PCA and LDA. These two methods can achieve better performances when the top 3 in the list are considered. Nevertheless, the recognition rate of our method is about 80%, but the two methods have a similar recognition rate of about 60%, when the top 10 in the list are considered. Figures 3.22 and 3.23 show the face recognition rates of the testing images under small and large pose variations, respectively. Figure 3.22 shows that PCA and LDA outperform our algorithm up to the first nine most similar faces. The reason for this is that the training images are those face images under small pose variations. Our algorithm has a similar recognition performance when the top 10 of the list are considered. Figure 3.23 shows that the recognition rates using PCA and LDA are lower than that of our algorithm. These results show that our algorithm outperforms PCA and LDA when the testing face images have large pose variations. This face recognition algorithm is based on the facial–feature points only, which is not sufficient to achieve a high recognition rate. However, for a large face database, our algorithm can be used to select a subset of face images from the database for further analysis. The problem due to pose variation can be alleviated, and the computational processing time required for comparing the feature points is much lower than with other advanced face recognition techniques.

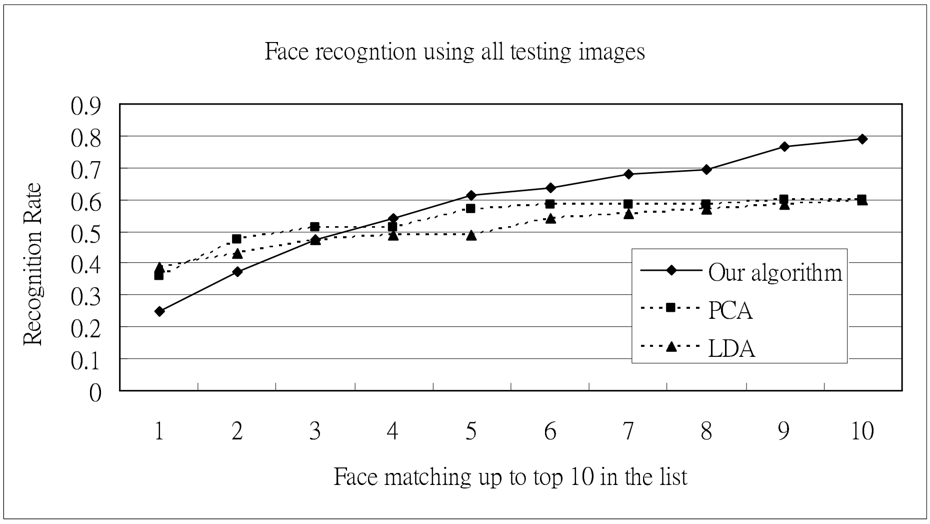


Figure 3.21: The face recognition rates of different face recognition techniques using all testing images.

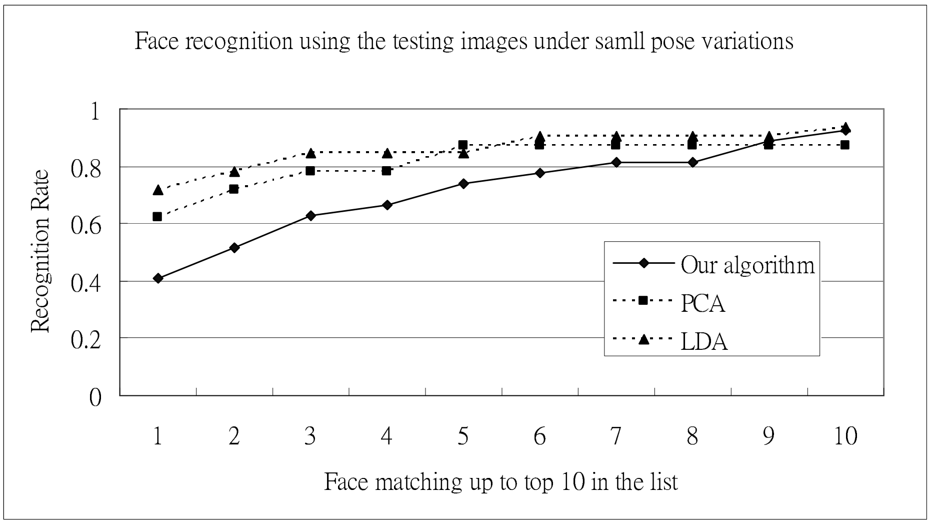


Figure 3.22: The face recognition rates of different face recognition techniques using the testing images under large pose variations.

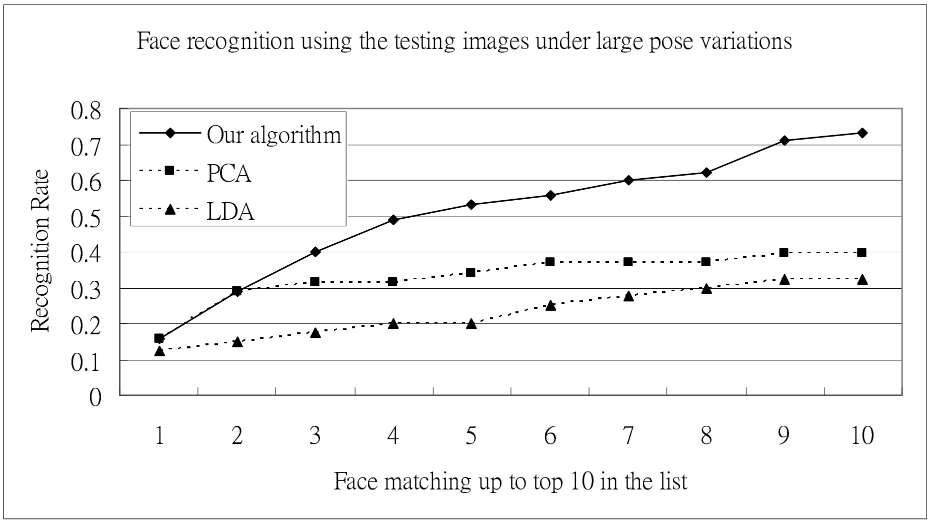


Figure 3.23: The face recognition rates of different face recognition techniques using the testing images under small pose variations.

3.6 Conclusion

In this chapter, a 3-D face reconstruction method is proposed to estimate the depth information about a human face based on face images under different poses. Our method does not require any camera calibration. In order to estimate the poses and the depths of the face model efficiently, the genetic algorithm is applied to minimize the similarity distance between the adapted face model and the faces under different poses.

Since the 3-D information about human faces is not available in most applications, a measurement to assess the accuracy of the constructed face model has been proposed, which is based on the similarity transform and the Levenberg-Marquardt method to find the optimal solution. With our proposed algorithm, both the poses and the scaling factors of the training face images with respect to the adapted 3-D face model, and the 3-D structure of the face model can be determined. In addition, experiments have shown that the estimation of the poses is consistent, and the estimated 3-D face models can be used for face recognition.

Chapter 4：Gabor Feature Selection and Extraction for Efficient Face Recognition

4.1 Introduction

Gabor features are an effective facial image representation, which are robust to variations caused by translation, rotation, and scaling, and are effective as local descriptors. However, the dimension of the Gabor features is large, and the features are highly redundant. This high dimensionality results in computation and memory requirements prohibitively large for face recognition. One possible way to reduce the dimension is to by sub-sampling the features spatially, i.e. extract the Gabor features at sub-sampled positions in a face region. However, this method cannot provide an acceptable face recognition performance if the features are extracted evenly without considering their importance over a face region. It is important to recognize that different facial regions in a face image have different levels of importance for face recognition, so a method to determine the optimal positions in a face region for the Gabor features can reduce the feature dimension, and can also improve the accuracy for face recognition. The feature dimension can be further reduced by selecting suitable Gabor kernels (i.e. particular scales and orientations) at different locations because the responses of the Gabor wavelets (GW) are strongly related to the pattern of a local face region.

In this chapter, a novel, local, feature-based face representation method for determining the informative regions in human faces is proposed. Gabor features at these informative regions are than extracted for face representation and recognition. Since the salient positions are different for different faces, where and which Gabor features should be selected must be image-dependent. The response of a GW is related to the edges in an image, which will have a large response if its wave vector is perpendicular to the edges. Therefore, the Gabor features to be selected for a pixel position should consider the edge orientation at that position. In our algorithm, an edge detector with different orientations is applied to face images. At an edge position, those Gabor filters whose kernels have their orientation perpendicular to the edge will be selected for feature extraction. In order to reduce the required computation, the simplified Gabor wavelets (SGWs) are employed, which can reduce the extraction time by 30% compared to the original Gabor wavelets.

The extracted Gabor features of the example faces will then form a face database. Figure 4.1 shows the structure of our face recognition system. There are two options for constructing a face database. The first is to extract all the Gabor features for each training face image, while the Gabor features of a query image are extracted selectively. The second is a reverse of the first approach, i.e. the Gabor features of each training face image are selected adaptively and then stored in a database, while all the Gabor features of the query image are extracted. For our algorithm, the second approach is preferred because it is more effective when vantage objects are used to construct a condensed database, which will be described in Section 4.4.1.

After a condensed database has been constructed, the distances between a query image and the images in the database are computed. The minimum distance will be chosen, and the corresponding database image is the best match to the query face image.

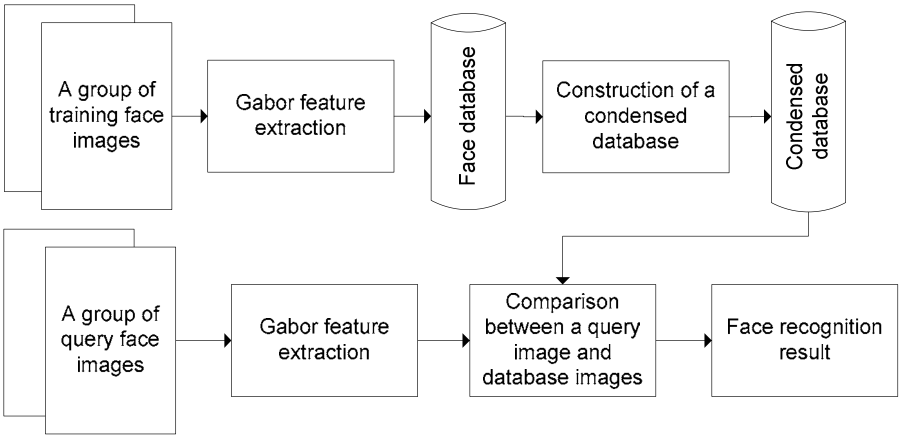

A group of training face images
Gabor feature extraction
Face database
Construction of a condensed database
Condensed database
A group of query face images
Gabor feature extraction
Comparison between a query image and database images
Face recognition result

Figure 4.1: Face recognition system of our proposed algorithm.

4.2 Gabor Feature Extraction

Gabor features are extracted from images for face recognition. Before feature extraction, each of the face images is normalization to a size of 64×64, and then processed using histogram equalization to make the faces have a similar lighting condition. When two faces are compared, they are aligned based on the positions of their two eyes.

Recently, a simplified version of the Gabor wavelets (SGWs) and a fast algorithm for extracting the features have been proposed. With the use of 3 center frequencies and 4 orientations, the features extracted using these SGWs achieve a comparable performance to that of using GWs for face recognition, and the runtime required for extracting features using the SGWs is only 70% of that required by GWs.

4.2.1 *Gabor Wavelet*

The Gabor wavelet has been described in Chapter 2.2.1. Recalling from (2.35), the equation of the GW is:

$$\psi_{\omega,\theta}(u,v)=\frac{1}{2\pi\sigma^2}e^{-\left(\frac{(u\cos\theta+v\sin\theta)^2+(-u\sin\theta+v\cos\theta)^2}{2\sigma^2}\right)}\cdot\left[e^{i(\omega u\cos\theta+\omega v\sin\theta)}-e^{-\frac{\omega^2\sigma^2}{2}}\right], \qquad (4.1)$$

where *u*, *v* denote the pixel position in the spatial domain, *ω* is the radial center frequency of the complex exponential, *θ* is orientation of the GW, and *σ* is the standard deviation of the Gaussian function.

In our algorithm, 12 Gabor filters, which consist of 3 scales and 4 orientations, are used to extract the features for face recognition. The Gabor features of an image can be obtained by convolving with the Gabor filters. Let $I(u, v)$ represent the gray-scale image, and $\psi_{\omega,\theta}(u, v)$ be the Gabor kernel of center frequency ω and orientation θ. The convolution output is defined as follows:

$$O_{\omega,\theta}(u,v) = I(u,v) * \psi_{\omega,\theta}(u,v), \tag{4.2}$$

where * denotes the convolution operator. A 1-D Gabor representation for the input image can be obtained by concatenating the rows as follows:

$$\mathrm{O}_{\omega,\theta} = \left[O_{\omega,\theta}(0,0), O_{\omega,\theta}(0,1), \ldots, O_{\omega,\theta}(0,N_r), O_{\omega,\theta}(0,1), \ldots, O_{\omega,\theta}(N_c,N_r)\right]^T, \tag{4.3}$$

where N_c and N_r are the numbers of columns and rows in the image. This 1-D Gabor representation is normalized to zero mean and unit variance. A jet at a pixel position is formed by a group of Gabor features at that position. For example, the Gabor jet at (u, v) is given as follows:

$$\begin{aligned} J(u,v) &= \left[O_{0,0}(u,v), O_{0,1}(u,v), \ldots, O_{0,3}(u,v), O_{1,0}(u,v), \ldots, O_{2,3}(u,v)\right]^T \\ &= \left[J_1(u,v), J_2(u,v), \ldots, J_4(u,v), J_5(u,v), \ldots, J_{12}(u,v)\right]^T. \end{aligned} \tag{4.4}$$

4.2.2 *A Simplified Version of the Gabor Wavelets*

In this sub-section, the formation of the SGWs and an efficient way of extracting features using the SGWs will be described. To simplify the description, only the real part of a SGW is discussed, as the same procedures can be applied to the imaginary part.

4.2.2.1 Formation of Simplified Version of Gabor Wavelet

Figure 4.2(a) shows the contour of a GW with the gray-level intensities representing the magnitudes of the wavelet. To simplify it, its values are quantized to a certain number of levels. Figure 4.2(b) shows the contours of the quantized GW. In SGWs, the quantized contours are approximated by rectangles. Figure 4.2(c) shows a rectangle whose size is just large enough to contain the corresponding contour of the quantized GW.

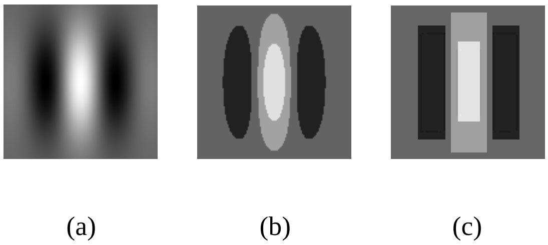


Figure 4.2: (a) The real part of the contour of a GW, (b) the quantized contour, and (c) the approximation of the elliptical quantized contours using rectangles just large enough to enclose them.

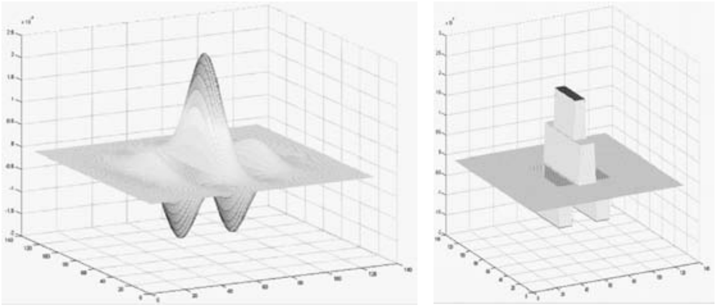
The number of rectangles in a SGW depends on the number of quantization levels used to quantize the GW. If more quantization levels are employed, the SGWs will be more similar to the GW, but more computation will then be involved in feature extraction.

The uniform quantization for determining the quantization levels has been used in constructing the SGWs. Let n_p and n_n be the number of quantization levels for the positive and negative values, respectively. When a zero quantization value is added, the total number of quantization levels of a SGW is n_p+n_n+1.

Let A_+ and A_- be the most positive and negative values of a SGW, respectively. The quantization levels for positive levels $c_+(k)$ and negative levels $c_-(k)$ are as follows:

$$\begin{aligned} c_+(k) &= \frac{A_+}{2n_p+1}\cdot 2k, \text{ where } k=1,\ldots,n_p \text{ and} \\ c_-(k) &= \frac{A_-}{2n_n+1}\cdot 2k, \text{ where } k=1,\ldots,n_n. \end{aligned} \tag{4.3}$$

Figure 4.3(a) and Figure 4.3(b) show the real part of a GW and SGW, respectively. In our experiment, the total number of quantization levels of a SGW is 7. Thus, $n_p = n_n = 3$.



(a) (b)

Figure 4.3: The 3-D structures of (a) the real part of a 2-D Gabor wavelet, and (b) the real part of the corresponding simplified Gabor wavelet.

4.2.2.2 Feature Extraction Using the Simplified Gabor Wavelets

The Gabor features of a face image can be extracted using the method described in 4.2.1. To have a SGW which is less sensitive to lighting conditions in an image, the coefficients of each SGW are subtracted by its mean value *m*. Since the SGWs are formulated by a group of rectangles, a fast algorithm for feature extraction can be used.

Consider a SGW that is convolved with an image $f(x,y)$, and the SGW is shifted to the pixel position (x_c, y_c), as shown in Figure 4.4. The convolution output at (x_c, y_c) is given as follows:

$$Y(x_c, y_c) = \sum_{k=1}^{NR_p} q_+(k) S_+(k) + \sum_{k=1}^{NR_n} q_-(k) S_-(k) + q_m S_F \, , \tag{4.4}$$

where $q_+(k)=c_+(k)-m$, $q_-(k)=c_-(k)-m$, $q_m=-m$ are demeaned SGW coefficients, and $S_+(k)$, $S_-(k)$ and S_F are the sum of the gray-level intensities of those pixels covered by the regions, either a single rectangular region or the difference between two rectangles, with quantization values $q_+(k)$, $q_-(k)$ and q_m, respectively. NR_p and NR_n are the number of rectangles with positive quantization values and negative quantization values, respectively.

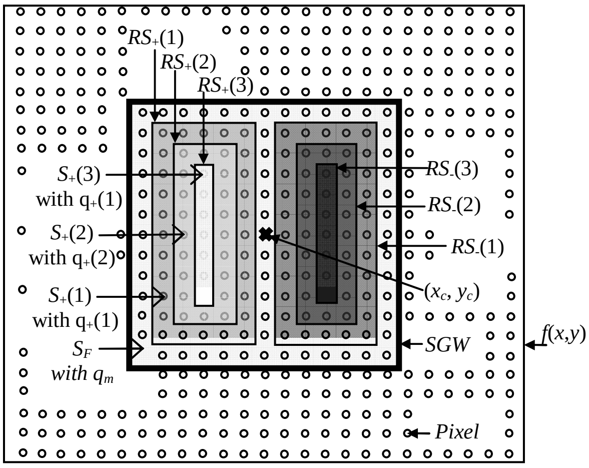


Figure 4.4: Image $f(x,y)$ is convolved with a SGW whose center is shifted to the pixel position (x_c, y_c)

Let $RS_+(k)$, $RS_–(k)$ and RS_F be the rectangular sum of the gray-level intensities of those pixels inside the rectangles with quantization values $q_+(k)$, $q_–(k)$ and q_m, respectively. $RS_+(k)$, $RS_–(k)$ and RS_F are computed based on the idea of an integral image [126], which can efficiently calculate the sum of pixel values within a rectangle. In addition, a fast algorithm for rectangles rotated by 45° or 135° is also available [127]. As a result, the SGWs consider 4 orientations only, which are 0°, 45°, 90°, and 135°.

Figure 4.6 and Figure 4.7 show the magnitudes of the Gabor features of the face image shown in Figure 4.5 extracted using GWs and SGWs at 3 scales and 4 orientations, respectively. The magnitudes reflect the similarity of the vicinity of each region to the kernels of the GWs and SGWs, which also show the characteristics of spatial locality, scale and orientation selectivity of GWs and SGWs. Such characteristics produce salient local features, such as the eyes, nose and mouth, which are suitable for face recognition.

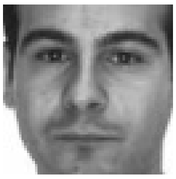


Figure 4.5: Human face image.

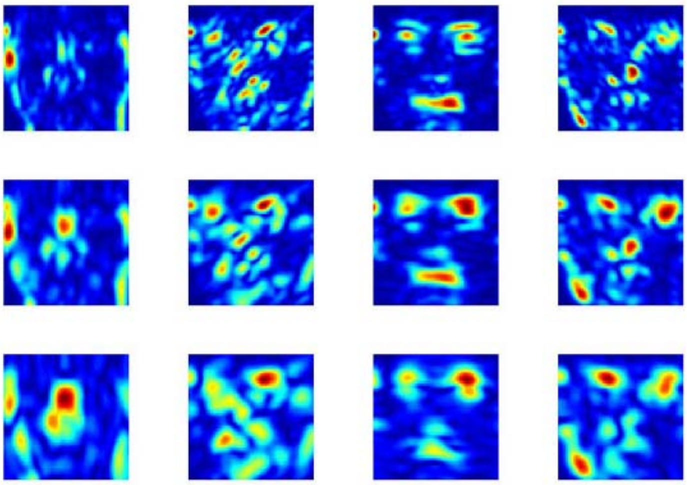


Figure 4.6: Magnitudes of the Gabor features extracted using the Gabor filters at 3 scales and 4 orientations.

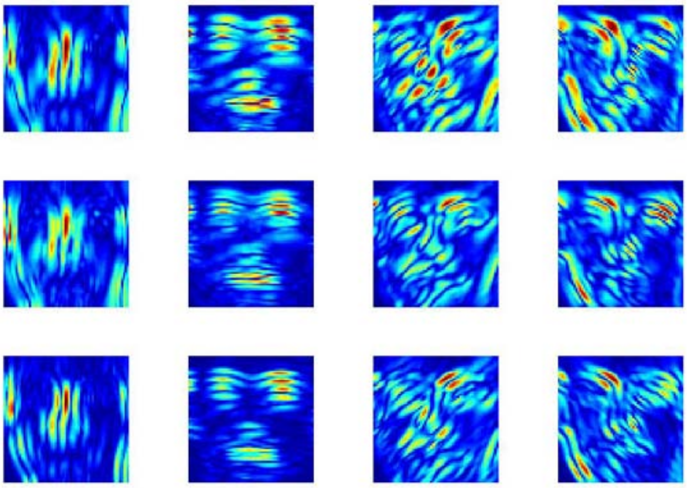
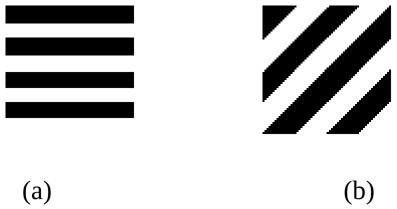


Figure 4.7: Magnitudes of the Gabor features extracted using the simplified Gabor filters at 3 scales and 4 orientations.

4.3 Gabor Feature Selection

The key ideas of our Gabor feature selection scheme are that more Gabor features will be selected for those positions which are more informative, and that the respective kernels to be chosen should produce strong responses. Because of the fact that the response of a GW is strong when the edges at the region under consideration are perpendicular to the wave vector of the GW, so the orientations of the edges in an image are used in the selection of useful Gabor features for face representation and recognition. For example, Figure 4.8 shows two simple images with strong edge characteristics in two different directions, while Figure 4.9 and Figure 4.10 show the corresponding magnitudes of the Gabor features extracted with different orientations. These results illustrate that the magnitudes of Gabor features will be the largest when the edges in an image are perpendicular to the wave vector of the Gabor function.



(a) (b)

Figure 4.8: Two simple images with strong edge characteristics in two different directions, (a) 0° and (b) 45°.

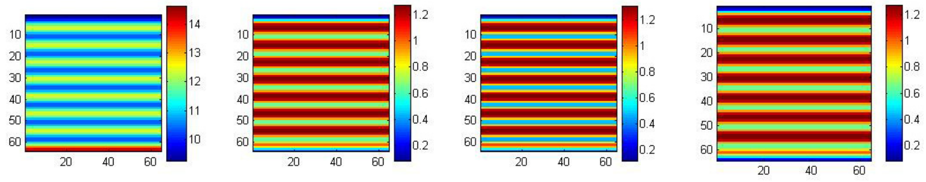


Figure 4.9: Magnitudes of the Gabor features extracted using Gabor functions of four different orientations based on the image shown in Figure 4.8(a).

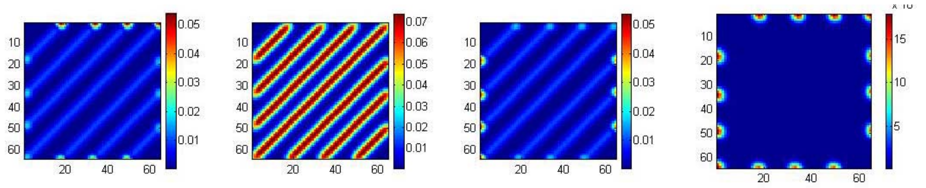


Figure 4.10: Magnitudes of the Gabor features extracted using Gabor functions of four different orientations based on the image shown in Figure 4.8(b).

4.3.1 *Edge Detection with Orientations*

An edge detector performs a 2-D spatial gradient measurement on an image. Typically, it is used to find the approximate absolute gradient magnitude at each point in an input gray-scale image. In order to detect edges of different orientation and to minimize the required computation, edge detection is usually performed with a group of 3×3 convolution masks. In our algorithm, four convolution masks with different orientations are used. Figure 4.13 shows the 3×3 convolution masks with different orientations. Each convolution mask estimates the gradient in the corresponding direction. G_0 and G_{90} are the pair of 3×3 convolution masks used in the Sobel edge detector, while G_{45} and G_{135} are the corresponding rotated convolution masks used to detect diagonal edges. In summary, G_0, G_{90}, G_{45} and G_{135} are used to detect horizontal edges, vertical edges, 45° edges and 135° edges, respectively.

1	2	1
0	0	0
-1	-2	-1

G_0

-1	0	1
-2	0	2
-1	0	1

G_{90}

0	1	2
-1	0	1
-2	-1	0

G_{135}

-2	-1	0
-1	0	1
0	1	2

G_{45}

Figure 4.11: The edge detectors used to detect edges of 4 different orientations.

Thresholding is used to obtain a binary version of the gradient map. All gradient values whose magnitudes are less than the threshold are set to 0, and are set to 255 if greater than the threshold. After thresholding, non-maximal suppression is applied to thin the edge map. Non-maximal suppression removes an edge pixel if its gradient magnitude is not a maximum along the line perpendicular to its gradient orientation.

Figure 4.12 shows the respective edge images of the face image shown in Figure 4.5 based on the 4 3×3 convolution masks. In our feature selection scheme, Gabor features with kernel orientation perpendicular to the edge direction of an edge pixel will be extracted. With the four edge images, a minimal set of Gabor features will be selected, which should be able to retain the as much information as possible about the face under consideration.

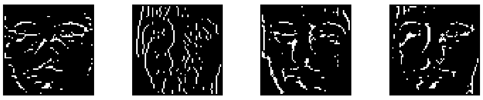


Figure 4.12: The four edge images of the face image shown in Figure 4.5 obtained using the four different 3×3 convolution masks.

4.4 Face Recognition in the Large Face Database

All the Gabor features of the training images and the query images are extracted by the SGWs in order to reduce computation. In our algorithm, two options to construct a face database have been proposed. The first, as shown in Figure 4.13, is to extract all the Gabor features for each training face image and to store them in the face database. With a query image, its Gabor features are extracted using the method proposed in Section 4.3. The selected Gabor features of the query image are then compared with the corresponding Gabor features in the database. This option can reduce the computation required for feature extraction runtime, but a huge amount of memory is required to store all the features of the training images. The second, as shown in Figure 4.14, is to extract the Gabor features for each training face image using the method proposed in Section 4.3. Then, all the Gabor features of the query image are extracted. Similarly, the selected Gabor features of the training images are compared to the corresponding Gabor features in the query image. This option can reduce the amount of memory needed for the database, but requires a slightly longer runtime to extract the Gabor features of the query image.

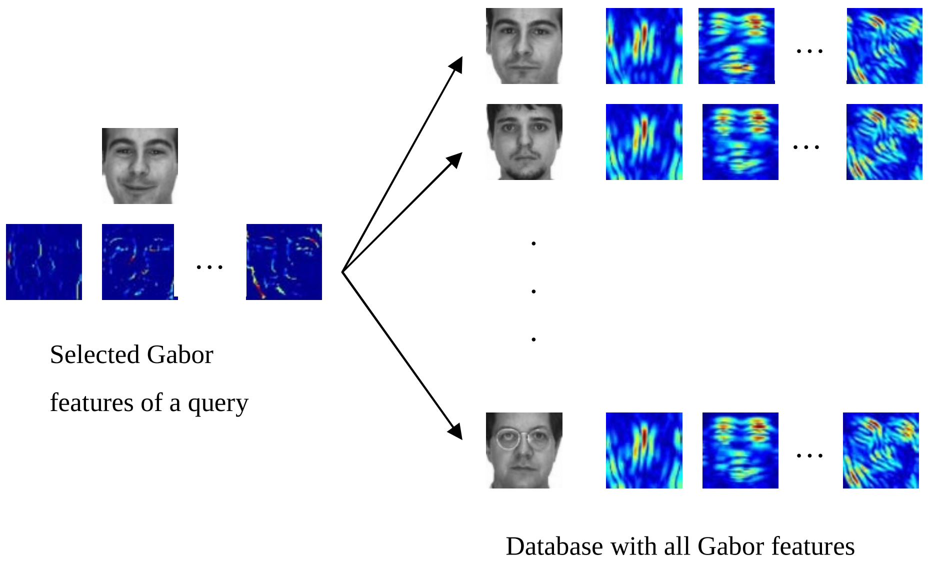


Figure 4.13: The first option of our face recognition system.

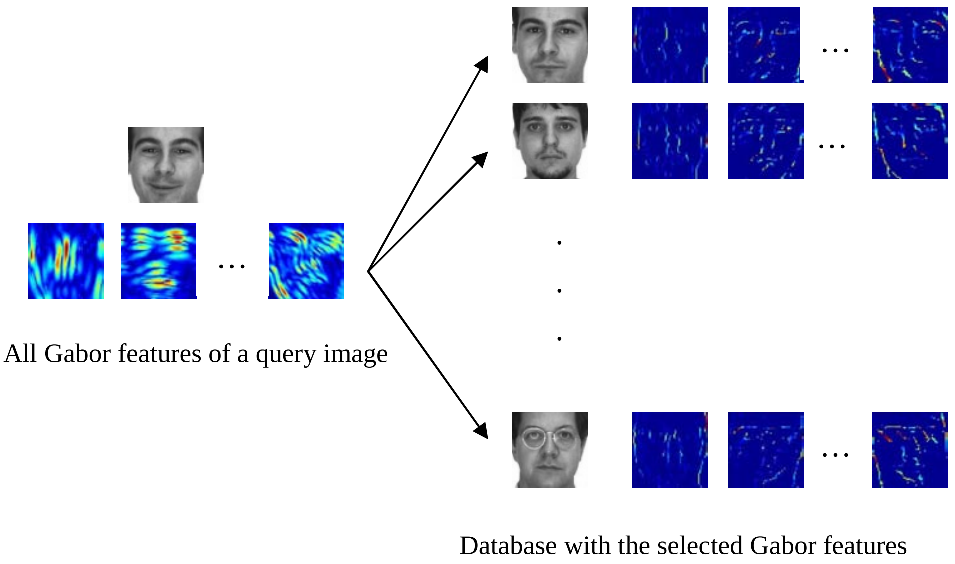

All Gabor features of a query image
Database with the selected Gabor features

Figure 4.14: The second option of our face recognition system.

4.4.1 *Condensed Database*

Although the dimension of the Gabor features have been reduced via the use of our selection scheme, the runtime required for face recognition will still be very long if the database size is very large. Therefore, to reduce this lengthy runtime for very large databases, a kind of indexing scheme or additional database structure is necessary. Vleugels and Veltkamp [135] suggested choosing a suitable number of objects from the database as “vantage objects”. The distances between the images in the database and each vantage objects are computed and ranked. For a query input, its corresponding distances to the vantage objects are computed. Then, those images in the database which have a similar distance to the vantage objects as the query input dues will be selected to form a smaller or condensed database for further analysis. This can help reduce a lot of computation in searching for a similar face to the query from a very large database. In our proposed algorithm, Gabor features of different kernel frequencies and orientations are considered to form the vantage-object structure.

4.4.1.1 Vantage Objects

In our face recognition system, 12 Gabor kernels formed from 3 scales and 4 orientations are used to extract the Gabor features. For the Gabor features extracted from each kernel, several representations such as the amplitude of their means, the mean of their amplitudes, the mean of their phases, etc. can be used for indexing, which are shown in (4.5), (4.6) and (4.7), respectively. From our experimental results, the amplitude of the Gabor features can achieve the best performance of having the matched faces selected in the condensed database. The amplitude of the mean of Gabor features:

$$M_{\omega,\theta} = \frac{1}{N_r \cdot N_c} \left| \sum_{i=1}^{N_c} \sum_{j=1}^{N_r} O_{\omega,\theta} \right| . \tag{4.5}$$

The mean of the amplitudes of Gabor features:

$$A_{\omega,\theta} = \frac{1}{N_r \cdot N_c} \sum_{i=1}^{N_c} \sum_{j=1}^{N_r} \left| O_{\omega,\theta} \right| . \tag{4.6}$$

The mean of the phases of Gabor features:

$$P_{\omega,\theta} = \frac{1}{N_r \cdot N_c} \sum_{i=1}^{N_c} \sum_{j=1}^{N_r} \angle O_{\omega,\theta} . \tag{4.7}$$

N_c and N_r are the numbers of columns and rows in the image, and ω and θ are the center frequency and orientation of the Gabor kernel.

4.4.1.2 Structure of the Indexing Scheme

To construct the condensed database for a query image, the average magnitudes of the Gabor features extracted using different SGWs are computed. Then, these values are sorted in ascending order. Figure 4.15 shows the structure of our indexing scheme. For the Gabor features extracted using the kernel of scale factor $\sqrt{2}$ and orientation of 0°, the 34th subject in the database has the minimum magnitude; the 26th subject has the second minimum magnitude and so on.

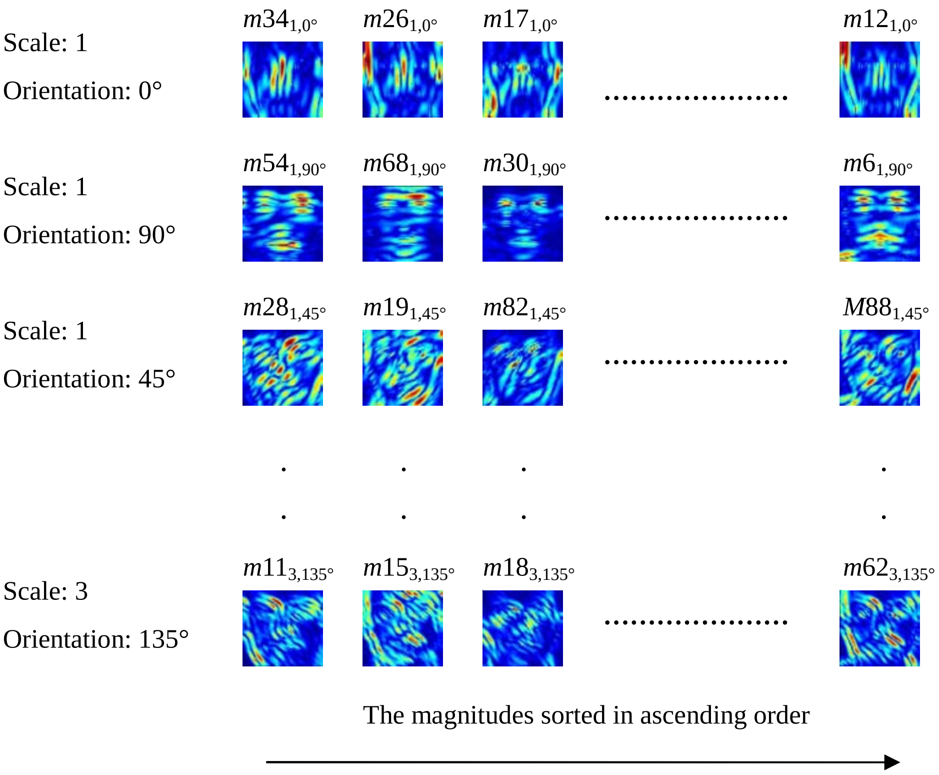


Figure 4.15: Structure of our indexing scheme.

When a query image is inputted, its Gabor features and its means of magnitudes corresponding to each Gabor kernel are calculated. Then, the means of magnitudes of these Gabor features are ranked in ascending order in the database, as shown in Figure 4.16. Since 12 different Gabor kernels are used, 12 different ranked lists can be formed. To select the face images to form the condensed database, those nearest neighbors in each ranked lists are chosen. The size of the condensed database is determined by the number of the nearest face images being selected.

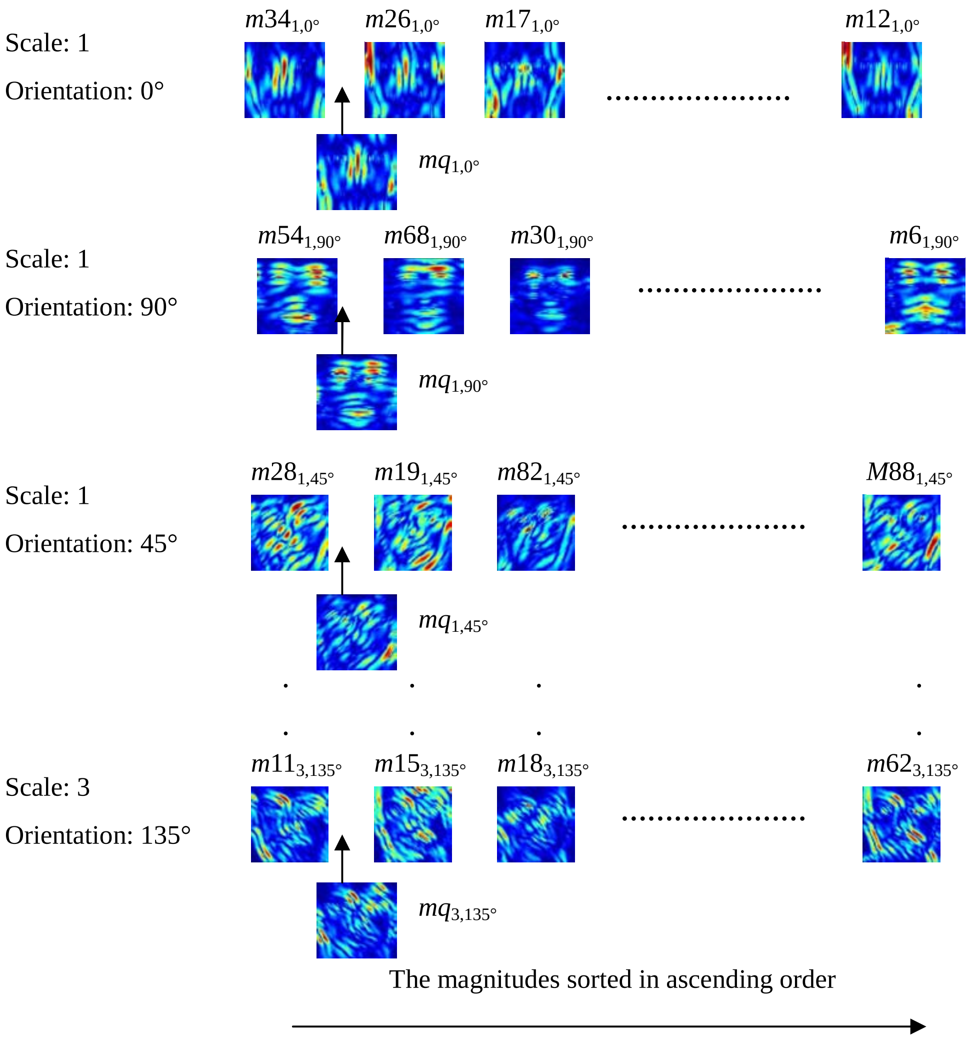


Figure 4.16: Retrieval of similar faces from a large database to form a condensed database.

Since the mean of magnitudes of the Gabor features for each Gabor kernel are required for the query image, the second option of our face recognition scheme is preferred. On the database side, only the selected Gabor features and their magnitudes are stored.

4.4.2 *Distance Measure*

In our face recognition algorithm, the distance between two images is computed using the following equation:

$$D_{\text{mag}} = \frac{1}{3N} \sum_{i=1}^{3N} \left\| X_i \right| - \left| Y_i \right\| , \tag{4.8}$$

where **X** is a 1-D selected Gabor feature of a database image, **Y** is the corresponding feature of the query image, and $N = N_{0°} + N_{45°} + N_{90°} + N_{135°}$. $N_{0°}$, $N_{45°}$, $N_{90°}$ and $N_{135°}$ are the numbers of edge pixels having an orientation of 0°, 45°, 90° and 135°, respectively. Since 3 kernel scales are used to extract the Gabor features, the dimensions of **X** and **Y** are both $3N$.

To compute the distances, only the magnitude difference between a database image and the query image is considered. This may turn out to be more robust with respect to changes in facial expression and other variations. The minimum distance between the query image and a database image means that the query image and this database image are corresponding to the same subject.

4.5 Experimental Results

4.5.1 *Feature Selection and Extraction*

A subset of the AR database [129] and the MIT database was used in the following experiments. In these databases, only the face images under normal conditions (neutral facial expressions, frontal lighting conditions, and frontal view) were considered. Those face images with occlusions were also excluded.

The resulting database used in our experiment contains 137 face images. To evaluate the performances of our scheme, 411 face images (363 from the AR database and 48 from the MIT database) were selected to form the testing set. For the AR database, face images with different facial expressions under the same lighting condition were selected. For the MIT database, images of three different scales for each person were selected.

For this experiment, the recognition rates based on our proposed algorithm are tabulated in Table 4.1. The recognition rates of "All Gabor features", "1st version" and "2nd version" are based on (4.8). The recognition rates based on "Gabor + PCA" and "Gabor + kernel PCA" are also shown. The quantization levels used in the simplified Gabor wavelet are five. Table 4.2 shows the corresponding recognition rates by which the Gabor features were extracted using the original GWs.

	Gabor+PCA	Gabor + kernel PCA	All Gabor features	1st version	2nd version
AR	93.388	95.868	96.419	96.419	96.28
MIT	83.333	89.583	93.75	91.66	91.66

Table 4.1: The recognition rates of different methods using the simplified Gabor features.

	Gabor+PCA	Gabor + kernel PCA	All Gabor features	1st version	2nd version
AR	94.766	96.97	97.245	96.694	96.97
MIT	87.5	91.667	95.833	93.75	95.833

Table 4.2: The recognition rates of different methods using the original Gabor features.

From the above results, the average numbers of edge pixels used for different orientations, i.e. $N_{0°}$, $N_{45°}$, $N_{90°}$ and $N_{135°}$, are about 200, 200, 400 and 200, respectively. As a result, the dimension of the Gabor features is about (200+400+200+200) ×3 = 3000, while the dimension using all the Gabor features is 64×64×12 = 49152. However, with our selection scheme, the degradation in the recognition rates using the SGWs is slight when compared to the GWs if the testing images are under normal lighting conditions.

As our feature selection scheme considers those pixel positions of the edges of an image, so the performance of the edge detector used will affect the recognition performances. Figure 4.17 shows the face recognition rates of our proposed algorithm based on the AR database and using different edge detectors. Although the Sobel edge detector and Canny edge detector can achieve a similar performance to our proposed one, more edges pixels are considered in order to keep the same performance level. Furthermore, the Canny edge detector can give a better performance level than the Sobel edge detector when the same number of pixels is used.

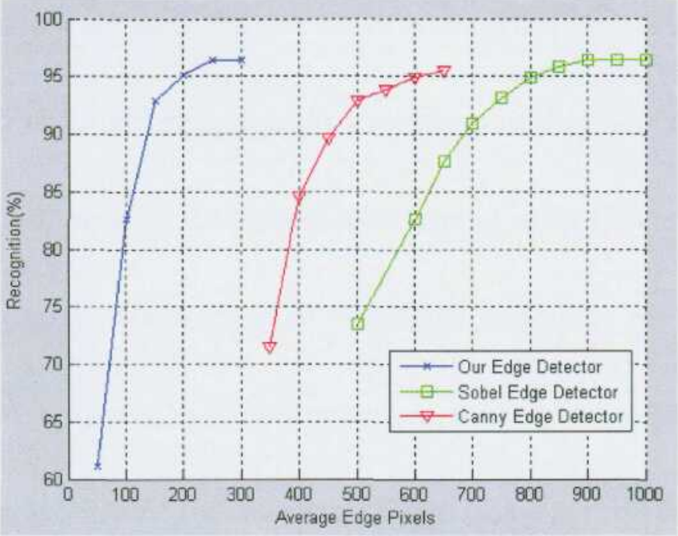


Figure 4.17: The face recognition rates of our proposed algorithm using different edge detectors.

From Table 4.1 and Table 4.2, face recognition based on features extracted by the SGWs has only slight degradation when compared to feature extracted by the original GWs. With the Pentium IV computer, the extraction runtime required each image using GWs is 44ms, while the extraction runtime for SGWs is 29ms. As a result, using the SGWs for feature extraction can save about 35% of the computational time. Furthermore, we have proposed two options for constructing the face database, as described in Section 4.4. From the experiment results, the two different methods result in the similar recognition rates.

4.5.2 *Condensed Database*

In this experiment, 1,152 fontal-view images of different subjects in the

FERET [81] database are used. Another set of 1,152 fontal-view images of the different subjects under different facial expressions is used as query images. In this section, we will evaluate the performance level of using Gabor features as vantage objects for the construction of condensed databases. For the Gabor features extracted using each of the Gabor kernels, the amplitude of their means, the mean of their amplitudes and the mean of their phase are computed. Then, these representations are used to build different human indexing schemes, and the performance for each of the representations is evaluated.

With a query image, its Gabor features are computed and the corresponding three different representations are ranked in the different rank lists for the various Gabor kernels. The number of faces selected to form the condensed database for the query input depends on the number of neighbors selected for each of the rank lists. The number of faces selected to the condensed database may vary far for different query inputs, although the same numbers of neighbors are selected. This is because the faces selected from one rank list may have appeared in another rank list. Since the number of face images selected to the condensed database is different, the average number of face images selected for each query image is measured for comparison. Figure 4.18 shows the recall rate against the sizes of condensed databases using different Gabor feature representations. The recall rate is defined as follows:

$$\text{Recall} = \frac{\text{Number of matched query images selected to the condensed database}}{\text{Total number of query images}}. \quad (4.9)$$

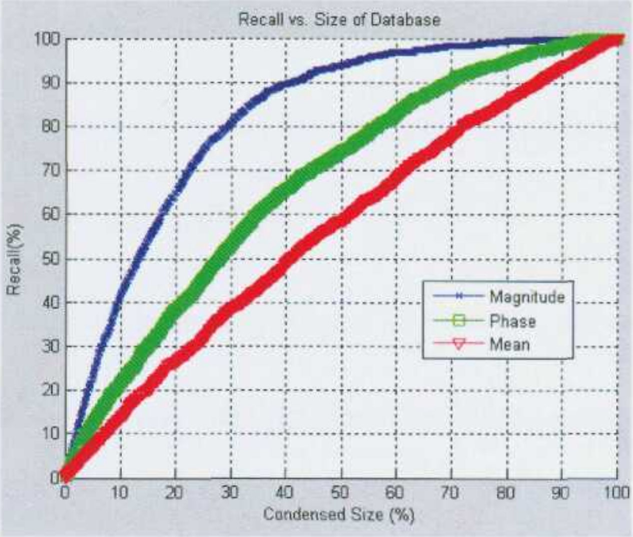


Figure 4.18: The performance levels of our Gabor vantage objects for construction of condensed databases.

According to Figure 4.18, about 70% of the original database must be considered in the condensed database in order to guarantee that the matched faces will be there if the mean of the magnitude of the Gabor features is used as the representation. The reason why the mean of magnitude has a better performance as a representation is that this representation is more robust with respect to changes in facial expressions and other variations. As the matched faces are included in the condensed database, the face recognition performance is the same as when using the original database.

The most computational part of the generation of the condensed database is the search of the positions of the query images on the respective rank lists.

The computation to search the position on each rank list is $O(\log n)$, and is done by using the self-balancing binary search tree. As a result, in this experiment the overall computation required to form a condensed database is about 1,152* $O(\log 1{,}152)$ which is much lower than the computation required for computing the distances between the Gabor features.

After the condensed database has been constructed, either the method proposed in Section 4.4 or a more computational and more accurate method can be used for face recognition.

4.6 Conclusion

In this chapter, a novel, local, feature-based face representation method has been described. The oriented edge images are used to select the Gabor features. Moreover, Gabor features of images can be extracted using the simplified Gabor wavelets, which can reduce the required extraction runtime by 30% when compared to the original Gabor wavelets, while the recognition rate can be preserved.

In our face recognition system, the Gabor features of the training face images and their magnitudes corresponding to different Gabor kernels are stored selectively, while all the Gabor features of the query image are extracted. The corresponding Gabor features of the query images are then compared to the selected Gabor features of the images in the database. To reduce the runtime for face recognition when the database size is very big, we have also proposed to use Gabor features to form vantage objects and rank lists so that a small or condensed database can be formed for a query input. Experimental results show that the recognition rates of our face recognition methods can be maintained while the computational time is reduced.

Chapter 5：Conclusion and Future Work

5.1 Conclusion

In this thesis, the current literature in 3-D reconstruction and 3-D face modeling for face recognition has been reviewed. In addition, the face recognition techniques based on selected Gabor features for face recognition has been described.

A novel and efficient method has been proposed to estimate the depth information about a 3-D face model based on face images under different poses for face recognition. The similarity distance between the adapted face model and the faces under different poses has been used to estimate the poses and the depths of the face model. The genetic algorithm is applied to search the optimal poses and depths of the face model. Our method does not require any camera calibration. Since the 3-D information about human faces is not available in most applications, a measurement to assess the accuracy of the constructed face model has been proposed. The similarity transform are used again to check the accuracy of the face model constructed. Our proposed algorithm can determine both the poses and the scaling factors of the training face images with respect to the adapted 3-D face model, and the 3-D structure of the face model. The estimated 3-D face models can be used for face recognition, especially for the face images with pose variations.

To retrieve face images in a database, a local feature-based face representation method has been proposed. The Gabor features of the images are extracted using the simplified Gabor wavelets, which can reduce the extraction runtime by 30%. Since the responses of the Gabor wavelets are strongly related to the edge orientations, Gabor filters whose kernel orientations perpendicular to the edges are selected for feature extraction. In our face recognition system, the magnitudes of the selected Gabor features of the training face images are stored, while all the Gabor features of a query image are extracted. The processing time for face recognition can be further reduced by constructing a condensed database for each query image. The condensed database is constructed by using the vantage object structure based on the Gabor features.

5.2 Future Work

In our proposed 3-D face model reconstruction algorithm, the features on a face are located manually. It is possible to detect the facial features using the current techniques such as corner detection [129], [130] and the active shape model. However, the correspondence between different face images is one of the challenges for 3-D face reconstruction. Moreover, more feature points, for example, at the chin and cheek can be considered in our proposed algorithm. These features have larger variations between different people so that they are more discriminant for 3-D face recognition. The detection of these features is the main challenge since no accurate algorithm can detect these features in the current researches. On the other hand, the computation is quite simple for the 3-D reconstruction on orthographic projection because intrinsic parameters are not considered. However, the approximation of the affine camera model is not proper when the face is close to the camera. The reconstruction methods based on this camera model yield distorted shapes due to the perspective effect. Therefore, perspective reconstruction of 3-D structure and motion has been considered. However, the camera calibration is another challenge for 3-D reconstruction.

In our proposed algorithm for face recognition in large face databases, only the Gabor kernels at different locations and the features with the specific kernel orientations have been used to select the Gabor features. To further reduce the redundancy in the Gabor features, the features with the specific kernel frequencies can be considered. In addition, the performance becomes worse when the face images are under different lighting condition. There are few approaches such as invariant features [136], variation modeling [137] and canonical form [138], for coping with variation in appearance due to illumination.

The 3-D face models can be used to recognize the face images under different poses. On the other hand, the optimally selected Gabor features and the indexing scheme can be applied to a large face database. However, the combination of them has not been investigated. Since the Gabor features of face images under different poses are very different, how to combine these techniques for automatic face recognition is an interesting and challenging research topic.

Appendices

Appendix 1: Proof of the Similarity Transform

Assume that there are *n* points in two different point sets, and (M_{x_i}, M_{y_i}, M_{z_i}) are the 3-D coordinates of the *i*th feature point in the adapted face model in which all the feature points have been centered. Similarly, (q_{x_i}, q_{y_i}) are the 2-D coordinates of the *i*th feature point in the non-frontal-view face image in which the feature points have also been centered. Then, the similarity distance of the *i*th feature point between the face model and the non-frontal-view face image is:

$$D^2 = \left\| q_{x_i} - s\left(r_{11}M_{x_i} + r_{12}M_{y_i} + r_{13}M_{z_i}\right)\right\|^2 + \left\| q_{y_i} - s\left(r_{21}M_{x_i} + r_{22}M_{y_i} + r_{23}M_{z_i}\right)\right\|^2 . \qquad \text{(A1.1)}$$

By applying partial differentiation to (A1.1) with respect to M_{z_i},

$$\frac{\partial D^2}{\partial M_{z_i}} = \left[q_{x_i} - s\left(r_{11}M_{x_i} + r_{12}M_{y_i} + r_{13}M_{z_i}\right)\right]\left(-sr_{13}\right) + \left[q_{y_i} - s\left(r_{21}M_{x_i} + r_{22}M_{y_i} + r_{23}M_{z_i}\right)\right]\left(-sr_{23}\right) = 0,$$

$$\text{i.e. } M_{z_i} = \frac{r_{13}q_{x_i} + r_{23}q_{y_i} - s\left(r_{11}M_{x_i} + r_{12}M_{y_i}\right)r_{13} - s\left(r_{21}M_{x_i} + r_{22}M_{y_i}\right)r_{23}}{s\left(r_{13}{}^2 + r_{23}{}^2\right)} . \qquad \text{(A1.2)}$$

Since

$$\begin{aligned} r_{11}r_{13} + r_{21}r_{23} + r_{31}r_{33} &= 0 \\ r_{12}r_{13} + r_{22}r_{23} + r_{32}r_{33} &= 0 \end{aligned} \qquad \text{(A1.3)}$$

(A1.2) can be rewritten as follows:

$$M_{z_i} = \frac{r_{13}q_{x_i} + r_{23}q_{y_i} + sr_{33}\left(r_{31}M_{x_i} + r_{32}M_{y_i}\right)}{s\left(r_{13}{}^2 + r_{23}{}^2\right)} . \qquad \text{(A1.4)}$$

For simplicity's sake, we rewrite (A1.4) as follows:

$$M_{z_i} = \frac{m}{s} + n\,. \qquad \text{(A1.5)}$$

where $m = \frac{r_{13} q_{x_i} + r_{23} q_{y_i}}{s\left(r_{13}{}^2 + r_{23}{}^2\right)}$ and $n = \frac{r_{33}\left(r_{31} M_{x_i} + r_{32} M_{y_i}\right)}{r_{13}{}^2 + r_{23}{}^2}$

Let $\mathrm{r1} = \left[r_{13} \quad r_{23}\right]$, $\mathrm{r2} = \left[r_{31} \quad r_{32}\right]$, $\mathrm{r3} = \left[r_{11} \quad r_{12}\right]$ and $\mathrm{r4} = \left[r_{21} \quad r_{22}\right]$, and $\mathbf{X_M}$, $\mathbf{Y_M}$ and $\mathbf{Z_M}$ are the three $n\times1$ matrices, which represent the *x*-, *y*- and *z*-coordinates of the centered feature points in the adapted face model. Let $\mathbf{M_{xy}} = [\mathbf{X_M}, \mathbf{Y_M}]^T$, and $\mathbf{q}$ be the $2\times n$ matrix, which represents the centered image point set. Then, rewrite (A1.4) into matrix form as follows:

$$\mathrm{Z_M}{}^T = \frac{\mathrm{r1}\cdot\mathrm{q} + s\cdot r_{33}\cdot\mathrm{r2}\cdot\mathrm{M}_{xy}}{s\cdot\mathrm{r1}\cdot\mathrm{r1}^T}\,. \qquad \text{(A1.6)}$$

Substituting (A1.5) into (A1.1), we have

$$D^2 = \left\| q_{x_i} - r_{13}m - s\left(r_{11}M_{x_i} + r_{12}M_{y_i} + r_{13}n\right)\right\|^2 + \left\| q_{y_i} - r_{23}m - s\left(r_{21}M_{x_i} + r_{22}M_{y_i} + r_{23}n\right)\right\|^2, \qquad \text{(A1.7)}$$

and then differentiating (A1.1) with respect to *s*

$$\begin{aligned}\frac{\partial D^2}{\partial s} &= \left[q_{x_i} - r_{13}m - s\left(r_{11}M_{x_i} + r_{12}M_{y_i} + r_{13}n\right)\right]\left(r_{11}M_{x_i} + r_{12}M_{y_i} + r_{13}n\right)\\ &\quad + \left[q_{y_i} - r_{23}m - s\left(r_{21}M_{x_i} + r_{22}M_{y_i} + r_{23}n\right)\right]\left(r_{21}M_{x_i} + r_{22}M_{y_i} + r_{23}n\right)\\ &= 0,\end{aligned}$$

$$\text{i.e.}\quad s = \frac{\left(q_{x_i} - r_{13}m\right)\left(r_{11}M_{x_i} + r_{12}M_{y_i} + r_{13}n\right) + \left(q_{y_i} - r_{23}m\right)\left(r_{21}M_{x_i} + r_{22}M_{y_i} + r_{23}n\right)}{\left(r_{11}M_{x_i} + r_{12}M_{y_i} + r_{13}n\right) + \left(r_{21}M_{x_i} + r_{22}M_{y_i} + r_{23}n\right)}\,. \qquad \text{(A1.8)}$$

We can also write *s* in matrix form, let

$$\mathrm{a}_i = \mathrm{r3}_i\cdot\mathrm{M}_{xy} + r_{i_{13}}\cdot\frac{r_{i_{33}}\cdot\mathrm{r2}_i\cdot\mathrm{M}_{xy}}{\mathrm{r1}_i\cdot\mathrm{r1}_i{}^T} \quad\text{and}\quad \mathrm{b}_i = \mathrm{r4}_i\cdot\mathrm{M}_{xy} + r_{i_{23}}\cdot\frac{r_{i_{33}}\cdot\mathrm{r2}_i\cdot\mathrm{M}_{xy}}{\mathrm{r1}_i\cdot\mathrm{r1}_i{}^T}\,,$$

which are $1\times n$ matrices, then

$$s_i = \frac{tr\left[\mathrm{q}_i \cdot \begin{bmatrix}\mathrm{a}_i \\ \mathrm{b}_i\end{bmatrix}^T\right]}{\mathrm{a}_i \cdot \mathrm{a}_i^T + \mathrm{b}_i \cdot \mathrm{b}_i^T} \tag{A1.9}$$

where *tr*[] of a matrix is the sum of its diagonal elements.

Appendix 2: The Levenberg-Marquardt method

The Levenberg-Marquardt (LM) method searches the parameters **x**, which will minimize (3.9), where $\mathrm{x} = [\theta \quad \varphi \quad \phi \quad s]^T$. Let

$$\mathbf{f}(\mathrm{x}) = \begin{bmatrix} q_{x_1} - s\left(r_{11}M_{x_1} + r_{12}M_{y_1} + r_{13}M_{z_1}\right) \\ \vdots \\ q_{x_n} - s\left(r_{11}M_{x_n} + r_{12}M_{y_n} + r_{13}M_{z_n}\right) \\ q_{y_1} - s\left(r_{21}M_{x_1} + r_{22}M_{y_1} + r_{23}M_{z_1}\right) \\ \vdots \\ q_{y_n} - s\left(r_{21}M_{x_n} + r_{22}M_{y_n} + r_{23}M_{z_n}\right) \end{bmatrix} \tag{A2.1}$$

and (3.9) can be rewritten as follows:

$$F(\mathrm{x}) = \frac{1}{n}\mathbf{f}(\mathrm{x})^{\mathrm{T}}\mathbf{f}(\mathrm{x}). \tag{A2.2}$$

We will compute **x** such that $\mathrm{x}^* = \mathrm{argmin}_{\mathbf{x}}\{F(\mathrm{x})\}$, (A2.3)

i.e. to minimize the (A2.2).

To find the solution of (A2.3), Levenberg and Marquardt suggested using a damped Gauss-Newton method. Assume **J** is the Jacobian of f(**x**), which is a matrix containing the first partial derivatives of f(**x**), i.e. $(\mathrm{J}(\mathrm{x}))_{ij} = \frac{\partial f_i}{\partial x_j}(\mathrm{x})$.

Then, solve

$$(\mathbf{J}^{\mathrm{T}}\mathbf{J}+\mu\mathbf{I})\mathbf{h}_{\mathrm{lm}} = -\mathbf{J}^{\mathrm{T}}\mathbf{f}, \tag{A2.4}$$

where **J** = J(**x**) and **f** = f(**x**), μ is the damping parameter and $\mathbf{h}_{\mathrm{lm}}$ is a descent direction.

The steps in the Levenberg-Marquardt method are shown as follows:

1. Initialize the damping parameter μ related to the size of the elements $\mathbf{A_0}$ = $\mathbf{J}(\mathbf{x_0})^T\mathbf{J}(\mathbf{x_0})$. For example, let $\mu_0 = 10^{-3} \cdot \max_i \{a_{ii}^{(0)}\}$,

2. Solve (A2.4) to find $\mathbf{h}_{lm}$

3. $\mathbf{x}_{new}=\mathbf{x}+\mathbf{h}_{lm}$ (A2.5)

4. Substitute (A2.5) back to (A2.4)

5. During iteration, the size of μ is controlled by the gain ratio

$$\varsigma = \frac{F(\mathrm{x}) - F(\mathrm{x} + \mathrm{h}_{\mathrm{lm}})}{\frac{1}{2}\mathrm{h}_{\mathrm{lm}}{}^{T}\left(\mu \mathrm{h}_{\mathrm{lm}} - \mathrm{J}^{T}\mathrm{f}\right)} \tag{A2.6}$$

The stopping criteria indicate that, at a global minimizer, F'(**x***) = g(**x***) = 0, so $\left\|\mathbf{J}^T\mathbf{f}\right\|_\infty \leq \varepsilon_1$. Another relevant stopping criterion is that the change in **x** is small, i.e. $\|\mathrm{x}_{\mathrm{new}} - \mathrm{x}\| \leq \varepsilon_2(\|\mathrm{x}\| + \varepsilon_2)$. In our algorithm, ε_1 and ε_1 are both set at 10^{-8}.

References

[1]. A. N. Ansari and M. Abdel-Mottaleb, “Automatic facial feature extraction and 3D face modeling using two orthogonal views with application to 3D face recognition,” *Pattern Recognition*, Vol. 38, No. 12, pp. 2549-2563, 2005.

[2]. A. Pentland and T. Choudhury, “Face recognition for smart environments,” *Computer*, Vol. 33, No. 2, pp. 50-55, 2000.

[3]. A. R. Chowdhury, R. Chellappa, S. Krishnamurthy and T. Vo, “3D face reconstruction from video using a generic model,” *Proceedings of IEEE International Conference on Multimedia and Expo*, Vol. 1, pp. 449- 452, 2002.

[4]. B. Gokberk, L. Akarun and E. Alpaydyn, “Feature Selection for Pose Invariant Face Recognition,” *16th International Conference on Pattern Recognition*, Vol. 4, pp. 40306, 2002.

[5]. B. Gökberk, M. Okan İrfanoğlu, L. Akarun and E. Alpaydın, “Learning the best subset of local features for face recognition,” *Pattern Recognition*, Available online, 2006.

[6]. B. Scholkopf, S. Mika, C. J. C. Burges, P. Knirsch, K. -R. Muller, G. Ratsch and A. J. Smola, “Input space versus feature space in kernel-based methods,” *IEEE Transactions on Neural Networks*, Vol. 10, No. 5, pp. 1000-1017, 1999.

[7]. B. Triggs, “Factorization methods for projective structure and motion,” *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 845-851, 1996.

[8]. C. J. Poelman and T. Kanade, “A paraperspective factorization method for shape and motion recovery,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 3, pp. 206-218, 1997.

[9]. C. Kotropoulos, A. Tefas and I. Pitas, “Frontal face authentication using morphological elastic graph matching,” *IEEE Transactions on Image Processing*, Vol. 9, No. 4, pp. 555-560, 2000.

[10]. C. Liu and H. Wechsler, “Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition,” *IEEE Transactions on Image Processing*, Vol. 11, No. 4, pp. 467-476, 2002.

[11]. C. Liu and H. Wechsler, “Independent component analysis of Gabor features for face recognition,” *IEEE Transactions on Neural Networks*, Vol. 14, No. 4, pp. 919-928, 2003.

[12]. C. Liu, “Gabor-based kernel PCA with fractional power polynomial models for face recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 5, pp. 572-581, 2004.

[13]. C. P. Jerian and R. Jain, "Structure from motion - a critical analysis of methods," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 21, No. 3, pp. 572-588, 1991.

[14]. C. S. Chua, F. Han and Y. K. Ho, "3D Human Face Recognition Using Point Signature," *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 233-239, 2000.

[15]. C. Tomasi and T. Kanade, "Shape and Motion from Image Streams under Orthography: a Factorization Method," *International Journal of Computer Vision*, Vol. 9, No. 2, pp. 137-154, 1992.

[16]. C. Zeller and O. Faugeras, "Camera Self-Calibration from Video Sequences: the Kruppa Equation Revisited," *Research Report 2793*, INRIA, 1996.

[17]. C. Zhang and F. S. Cohen, "3-D face structure extraction and recognition from images using 3-D morphing and distance mapping," *IEEE Transactions on Image Processing*, Vol. 11, No. 11, pp. 1249- 1259, 2001.

[18]. D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," *Addison-Wesley*, Reading, MA, 1989.

[19]. D. Gong, Q. Yang, X. Tang and J. H. Lu, "Extracting micro-structural gabor features for face recognition," *IEEE International Conference on Image Processing*, Vol. 2, pp. 942-945, 2005.

[20]. D. Gonzalez-Jimenez and J. L. Alba-Castro, "Shape contexts and Gabor features for face description and authentication," *IEEE International Conference on Image Processing*, Vol. 2, pp. 962-965, 2005.

[21]. D. H. Liu, K. M. Lam and L. S. Shen, "Optimal sampling of Gabor features for face recognition," *Pattern Recognition Letter*, Vol. 25, No. 2, pp. 267-276, 2004.

[22]. D. L. Jiang, Y. X. Hu, S. C. Yan, L. Zhang, H. J. Zhang and W. Gao, "Efficient 3D reconstruction for face recognition," *Pattern Recognition*, Vol. 38, No. 6, pp. 787-798, 2005.

[23]. D. Liebowitz and A. Zisserman, "Metric rectification for perspective images of planes," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 482-488, 1998.

[24]. D. Marquardt, "An Algorithm for Least Squares Estimation on Nonlinear Parameters," *SIAM Journal of the Society for Industrial and Applied Mathematics*, Vol. 11, No. 2, pp.431-441, 1963.

[25]. D. Nandy and J. Ben-Arie, "Shape from recognition: a novel approach for 3-D face shape recovery," *IEEE Transactions on Image Processing*, Vol. 10, No. 2, pp. 206-217, 2001.

[26]. E. Trucco and A. Verri, "Introductory Techniques for 3-D Computer Vision," *Prentice Hall*, 1998.

[27]. F. Galton, “Personal identification and description,” *Nature*, pp. 173-188, 1888.

[28]. F. L. Li and K. X. Xu, “Optimal Gabor Kernel's Scale and orientation selection for face classification,” *Optics & Laser Technology*, Vol. 39, No. 4, pp. 852-857, 2006.

[29]. G. Dai and Y. Qian, “A Gabor direct fractional-step LDA algorithm for face recognition,” *IEEE International Conference on Multimedia and Expo*, Vol. 1, pp. 61-64, 2004.

[30]. G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, “Classifying facial actions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, pp. 974-989, 1999.

[31]. G. G. Gordon, “Face recognition based on depth and curvature features,” *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 808-810, 1992.

[32]. G. P. Stein and A. Shashua, “Direct estimation of motion and extended scene structure from a moving stereo rig,” *Proceedings of IEEE Conference on Computer Society Conference,* pp. 211-218, 1998.

[33]. H. Tanaka, M. Ikeda and H. Chiaki, “Curvature-based face surface recognition using spherical correlation - Principal directions for curved object recognition,” *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 372-377, 1998.

[34]. I. S. Bruner and R. Tagiuri, “The Perception of People,” *Handbook of Social Psychology*, Vol.2, G. Lindzey, Ed., Addison-Wesley, Reading, MA, pp. 634-654, 1954.

[35]. J. Ahlberg, “CANDIDE-3 - Updated Parameterised Face,” Linkoping University, Lysator LiTH-ISY-R-2325, 2001.

[36]. J. G. Daugman, “Two-dimensional spectral analysis of cortical receptive field profiles,” *Vis. Res.*, Vol. 20, pp. 847-856, 1980.

[37]. J. G. Daugman, “Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters,” *Journal of the Optical Society of America A - Optics, Image Science, and Vision*, Vol. 2, No. 7, pp. 1160-1169, 1985.

[38]. J. Jones and L. Palmer, “An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex,” *J. Neurophys.*, pp. 1233-1258, 1987.

[39]. J. Karhunen, E. Oja, L. Wang, R. Vigario, and J. Joutsensalo, “A class of neural networks for independent component analysis,” *IEEE Transactions on Neural Networks*, Vol. 8, pp. 486-504, 1997.

[40]. J. Oliensis, “A Multi-Frame Structure-from-Motion Algorithm under Perspective Projection,” *International Journal of Computer Vision*, Vol. 34, No. 2-3, pp. 163-192, 1999.

[41]. J. W. Gu, J. H. Han, “3D reconstruction in a constrained camera system,” *Pattern Recognition Letters,* Vol. 23, No. 11, pp. 1337-1347, 2002.

[42]. J. W. Lu, K. N. Plataniotis and A. N. Venetsanopoulos, “Face recognition using kernel direct discriminant analysis algorithms,” *IEEE Transactions on Neural Networks*, Vol. 14, No. 1, pp. 117-126, 2003.

[43]. J. W. Yi and J. H. Oh, “Recursive resolving algorithm for multiple stereo and motion matches,” *Image and Vision Computing*, Vo. 15, No. 3, pp. 181-196, 1997.

[44]. J. Weng, T. S. Huang and N. Ahuja, “Motion and structure from two perspective views: Algorithms, error analysis, and error estimation,” *IEEE Transactions on Pattern Analysis and Machine Vision*, Vol. 11, No. 5, pp.451-476, 1989.

[45]. K. C. Chung, S. C. Kee and S. R. Kim, “Face Recognition Using Principal Component Analysis of Gabor Filter Responses,” *International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pp. 53-58, 1999.

[46]. K. Levenberg, “A Method for the Solution of Certain Problems in Least Squares,” *Quart. Appl. Math.* Vol. 2, pp. 164-168, 1944.

[47]. K. Messer, J. Kittler, M. Sadeghi, et al., “Face authentication test on the BANCA database,” *Proceedings of 17th International Conference on Pattern Recognition*, Vol. 4, pp. 523-532, 2004.

[48]. K. Okajima, “Two-dimensional Gabor-type receptive field as derived by mutual information maximization,” *Neural Networks*, Vol. 11, No. 3, pp. 441-447, 1998.

[49]. K. Sengupta and C. C. Ko, “Scanning face models with desktop cameras,” *IEEE Transactions on Industrial Electronics*, Vol. 48, No. 5, pp. 904-912, 2001.

[50]. K. Sengupta and P. Burman, “A curve fitting problem and its application in modeling objects inmonocular image sequences,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, pp. 674-686, 2002.

[51]. L. Ayinde and Y. H. Yang, “Face recognition approach based on rank correlation of Gabor-filtered images”, *Pattern Recognition*, Vol. 35, pp. 1275-1289, 2002.

[52]. L. Hajder, “An Iterative Improvement of the Tomasi-Kanade Factorization,” *Third Hungarian Conference on Computer Graphics and Geometry*, 2005.

[53]. L. L. Shen and L. Bai, “AdaBoost Gabor feature selection for classification,” *Proceeding of Image and Vision Computing Conference,* pp. 77-83, 2004.

[54]. L. L. Shen and L. Bai, "Gabor wavelets and kernel direct discriminant analysis for face recognition," Proceedings of the 17th International Conference on Pattern Recognition, Vol. 1, pp. 284-287, 2004.

[55]. L. L. Shen and L. Bai, "Information Theory for Gabor Feature Selection for Face Recognition," *Journal on Applied Signal Processing*, Vol. 2006, Article ID. 30274, pp. 1-11, 2006.

[56]. L. L. Shen and L. Bai, "MutualBoost learning for selecting Gabor features for face recognition," *Pattern Recognition Letters*, Vol. 27, No. 15, pp. 1758-1767, 2006.

[57]. L. L. Shen, L. Bai and M. Fairhurst, "Gabor wavelets and General Discriminant Analysis for face identification and verification," *Image and Vision Computing*, Available online 30 June 2006.

[58]. L. Nanni and D. Maio, "Weighted Sub-Gabor for face recognition," *Pattern Recognition Letters*, Available online 7 November 2006.

[59]. L. S. Shapiro, A. Zisserman and M. Brady, "3D Motion Recovery via Affine Epipolar Geometry," *International Journal of Computer Vision*, Vol. 16, pp. 147-182, 1995.

[60]. L. Shen and L. Bai, "Gabor feature based face recognition using Kernel methods," *Proceedings of Sixth IEEE International Conference Automatic Face and Gesture Recognition*, pp.170-176, 2004.

[61]. L. Wiskott, "Phantom faces for face analysis," *Pattern Recognition*, Vol. 30, No. 6, pp.837-846, 1997.

[62]. L. Wiskott, J. M. Fellous, N. Kruger and C. vonder Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 775-779, 1997.

[63]. L. Xi, "3D orthographic reconstruction based on robust factorization method with outliers," *International Conference on Image Processing*, Vol. 3, pp. 1927-1930, 2004.

[64]. M. D. Kelly, "Visual Identification of People by Computer," *Technical Report* AI-130, Stanford AI Proj., Stanford, CA, 1970.

[65]. M. Han and T. Kanade, "Reconstruction of a scene with multiple linearly moving objects," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol.2, pp. 542-549, 2000.

[66]. M. J. Lyons, J. Budynek and S. Akamastsu, "Automatic Classification of Single Facial Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 12, pp. 1357-1362, 1999.

[67]. M. J. Lyons, J. Budynek and S. Akamatsu, “Automatic classification of single facial images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 12, pp. 1357-1362, 1999.

[68]. M. Lades, J. C. Vorbruggen, J. Buhmann, et al., “Distortion invariant object recognition in the dynamic link architecture,” *IEEE Transactions on Computers*, Vol. 42, No. 3, pp. 300-311, 1993.

[69]. M. Lades, J. C. Vorbruggen, J. Buhmann, et al., “Distortion invariant object recognition in the dynamic link architecture,” *IEEE Transactions on Computers*, Vol. 42, No. 3, pp.300-311, 1993.

[70]. M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel and J. Movellan, “Recognizing facial expression: machine learning and application to spontaneous behaviour,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 568- 573, 2005.

[71]. M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel and J. Movellan, “Machine learning methods for fully automatic recognition of facial expressions and facial actions,” *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 1, pp. 592- 597, 2004.

[72]. M. S. Bartlett, J. R. Movellan and T. J. Sejnowski, “Face recognition by independent component analysis,” *IEEE Transactions on Neural Networks*, Vol. 13, No. 6, pp. 1450-1464, 2002.

[73]. M. Sainz, N. Bagherzadeh and A. Susin, “Recovering 3D metric structure and motion from multiple uncalibrated cameras,” *Proceedings of International Conference on Information Technology: Coding and Computing*, pp. 268-273, 2002.

[74]. M. W. Lee and S. Ranganath, “Pose-invariant face recognition using a 3D deformable model,” *Pattern Recognition*, Vol. 36, No 8, pp. 1835-1846, 2003.

[75]. M. Werman and D. Weinshall, “Similarity and affine invariant distances between 2D point sets,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 8, pp. 810-814, 1995.

[76]. M. Zucchelli, “Optical Flow Based Structure from Motion,” *Ph.D. Thesis*, Royal Institute of Technology, 2002.

[77]. N. Krüger, M. Pötzsch and C. von der Malsburg, Determination of face position and pose with a learned representation based on labelled graphs,” *Image and Vision Computing*, Vol. 15, No. 8, pp. 665-673, 1997.

[78]. N. W. Campbell and B. T. Thomas, “Automatic selection of Gabor filters for pixel classification,” *Sixth International Conference on Image Processing and Its Applications*, Vol. 2, pp. 761-765, 1997.

[79]. O. Ayinde and Y. H. Yang, “Face recognition approach based on rank correlation of Gabor-filtered images,” *Pattern Recognition*, Vol. 35, No. 6, pp. 1275-1289, 2002.

[80]. O. Faugeras and R. Keriven, “Variational Principles, Surface Evolution, PDE's Level Set Methods, and the Stereo Problem,” *IEEE Transactions on Image Processing*, Vol. 7, pp. 336-344, 1998.

[81]. P. J. Phillips, Moon Hyeonjoon, S. A. Rizvi and P. J. Rauss, “The FERET evaluation methodology for face-recognition algorithms,” *IEEE Transactions Pattern Analysis and Machine Intelligence,* Vol. 22, No. 10, pp. 1090-1104, 2000.

[82]. P. Kalocsai, C. von der Malsburg and J. Horn, “Face recognition by statistical analysis of feature detectors,” *Image and Vision Computing*, Vol. 18, No. 4, pp. 273-278, 2000.

[83]. P. Sturm and B. Triggs, “A Factorization Based Algorithm for multi-Image Projective Structure and Motion,” *Proceedings of the 4th European Conference on Computer Vision*, Vol. 1065, pp. 709-720, 1996.

[84]. P. Sturm, “A case against Kruppa's equations for camera self-calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 10, pp. 1199-1204, 2000.

[85]. P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 511-518, 2001.

[86]. P. Yang, S. G. Shan, W. Gao, S. Z. Li and D. Zhang, “Face recognition using Ada-Boosted Gabor features,” *Proceedings. Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 356-361, 2004.

[87]. R. Alterson and M. Spetsakis, “Object recognition with adaptive Gabor features,” *Image and Vision Computing*, Vol. 22, No. 12, pp. 1007-1014, 2004.

[88]. R. Chellappa, C.L. Wilson and S. Sirohey, “Human and machine recognition of faces: a survey,” *Proceedings of the IEEE*, Vol. 83, No. 5, pp. 705-741, 1995.

[89]. R. Lengagne, P. Fua and O. Monga, “3D stereo reconstruction of human faces driven by differential constraints,” Image and Vision Computing, Vol. 18, No. 4, pp. 337-343, 2000.

[90]. R. M. Haralick, H. Joo, C. Lee, X. Zhuang, V. G. Vaidya and M. B. Kim, “Pose estimation from corresponding point data,” *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 19, No. 6, pp. 1426-1446, 1989.

[91]. R. Szeliski and S. B. Kang, “Recovering 3D shape and motion from image streams using nonlinear least squares” *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 752-753, 1993.

[92]. S. Arca, P. Campadelli and R. Lanzarotti, “A face recognition system based on automatically determined facial fiducial points,” *Pattern Recognition*, Vol. 39, No. 3, pp. 432-443, 2006.

[93]. S. B. Kang and M. Jones, “Appearance-Based Structure from Motion Using Linear Classes of 3-D Models,” *International Journal of Computer Vision*, Vol. 49, No. 1, pp. 5-22, 2002.

[94]. S. G. Kong, J. Heo, B. R. Abidi, J. Paik and M. A. Abidi, “Recent advances in visual and infrared face recognition - a review,” *Computer Vision and Image Understanding*, Vol. 97, No. 1, pp. 103-135, 2005.

[95]. S. Mahamud and M. Hebert, “Iterative projective reconstruction from multiple views,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 430-437, 2000.

[96]. S. Malassiotis and M. G. Strintzis, “Robust Face Recognition Using 2D and 3D Data: Pose and Illumination Compensation,” *Pattern Recognition*, Vol. 38, No. 12, pp. 2537-2548, 2005.

[97]. S. Pankanti, R. M. Bolle and A. Jain, “Guest editors’ introduction: Biometrics-the future of identification,” *Computer*, Vol. 33, No. 2, pp. 46-49, 2000.

[98]. S. Ullman, “The Interpretation of Visual Motion,” Cambridge, Mass.: MIT Press, 1979.

[99]. T. Heseltine, N. Pears and J. Austin, “Three-dimensional face recognition: an eigensurface approach,” *International Conference on Image Processing*, Vol. 2, pp. 1421-1424, 2004.

[100]. T. Kanade, “Computer Recognition of Human Faces,” Birkhauser, Basel and Stuttgart, 1977.

[101]. T. Morita and T. Kanade, “A sequential factorization method for recovering shape and motion from image streams,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 8, pp. 858-867, 1997.

[102]. T. S. Huang and A. N. Netravali, “Motion and structure from feature correspondences: A review,” *Proceedings of the IEEE*, Vol. 82, No. 2, pp. 252-268, 1994.

[103]. T. S. Huang and C. H. Lee, “Motion and structure from orthographic projections,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 5, pp. 536-540, 1989.

[104]. 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.

[105]. V. Blanz and T. Vetter, “A Morphable model for the synthesis of 3D faces,” *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 187-194, 1999.

[106]. V. Blanz and T. Vetter, “Face recognition based on fitting a 3D morphable model,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 9, pp. 1063-1074, 2003.

[107]. W. E. L. Grimson, “Computational experiments with a feature based stereo algorithm,” *IEEE Transaction on Pattern Analysis and Machine Vision*, Vol. 7, No. 1, pp. 17-34, 1985.

[108]. W. Polzleitner, “Defect detection on wooden surface using Gabor filters with evolutionary algorithm design,” *International Joint Conference on Neural Networks*, Vol. 1, pp. 750-755, 2001.

[109]. W. W. Bledsoe, “The Model Method in Facial Recognition,” *Panoramic Research Inc., Technical Report* PRI: 15, Palo Alto, CA, 1964.

[110]. W. Zhao, R. Chellappa, P. J. Phillips and A. Rosenfeld, “Face Recognition: A Literature Survey,” *ACM Computing Survey*, Vol. 35, No. 4, pp. 399-458, 2003.

[111]. X. Armangué and J. Salvi, “Overall view regarding fundamental matrix estimation,” *Image and Vision Computing*, Vol. 21, No. 2, pp. 205-220, 2003.

[112]. X. D. Xie and K. M. Lam, “Gabor-based kernel PCA with doubly nonlinear mapping for face recognition with a single face image,” *IEEE Transactions on Image Processing*, Vol. 15, No. 9, pp. 2481-2492, 2006.

[113]. X. F. He, S. C. Yan, Y. X. Hu, P. Niyogi and H. J. Zhang, “Face recognition using Laplacianfaces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 3, pp. 328-340, 2005.

[114]. X. L. Wang and H. R. Qi, “Face Recognition Using Optimal Non-orthogonal Wavelet Basis Evaluated by Information Complexity,” *16th International Conference on Pattern Recognition*, Vol. 1, pp. 164-167, 2002.

[115]. X. Lu, D. Colbry and A. K. Jain, “Three-dimensional model based face recognition,” *International Conference on Pattern Recognition*, Vol. 1, pp. 362-366, 2004.

[116]. X. P. Hu and Narendra Ahuja, “Motion estimation under orthographic projection,” *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 6, pp. 848-853, 1991.

[117]. Y. B. Joo and J. Jin, Content-based image retrieval for not-well-framed images usingmultiresolutional eigen-features,” IEEE International Conference on Multimedia and Expo, Vol. 2, pp. 665-668, 2000.

[118]. Y. M. Zhang, X. M. Zhang and Y. C. Guo, “Face recognition base on low dimension Gabor feature using direct fractional-step LDA,” *International Conference on Computer Graphics, Imaging and Vision: New Trends*, pp. 103-108. 2005.

[119]. Y. Shan, Z. C. Liu and Z. Zhang, “Model-based bundle adjustment with application to face modelling,” *Proceedings of IEEE International Conference on Computer Vision*, Vol.2, pp. 644-651, 2001.

[120]. Y. Su, S. Shan, X. Chen and W. Gao, “Hierarchical Ensemble of Gabor Fisher Classifier for Face Recognition,” *7th International Conference on Automatic Face and Gesture Recognition*, pp. 91-96, 2006.

[121]. Y. Xirouhakis and A. Delopoulos, “Least squares estimation of 3D shape and motion of rigid objects from their orthographic projections,” *IEEE Transaction on Pattern Analysis and Machine Intelligence,* Vol. 22, No. 4, pp. 393-399, 2000.

[122]. Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 11, pp. 1330-1334, 2000.

[123]. Z. Zhang, “Determining the Epipolar Geometry and its Uncertainty: A Review,” *International Journal of Computer Vision*, Vol. 27, No. 2, pp. 161-195, 1998.

[124]. Z. Zhang, “On the optimization criteria used in two-view motion analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 7, pp. 717-729, 1998

[125]. Z. Zhang, Z. Liu, D. Adler, M. F. Cohen, E. Hanson and Y. Shan, “Robust and rapid generation of animated faces from video images: A model-based modeling approach,” *International Journal of Computer Vision*, Vol. 58, No. 2, pp. 93-119, 2004.

[126]. S. Z. Li, and Z. Q. Zhang, “FloatBoost learning and statistical face detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 9, pp. 1112-1123, 2004.

[127]. M. H. Yang, D. J. Kriegman and N. Ahuja, “Detecting faces in images: a survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 1, pp. 34-58, 2002.

[128]. A. U. Batur and M. H. Hayes, “Adaptive active appearance models,” *IEEE Transactions on Image Processing*, Vol. 14, No. 11, pp. 1707-1721, 2005.

[129]. Z. Y. Xiong, Y. Q. Chen, R. Wang and T. S. Huang, “A real time automatic access control system based on face and eye corners detection, face recognition

and speaker identification," *Proceedings. 2003 International Conference on Multimedia and Expo*, Vol. 3, pp. 233-236, 2003.

[130]. V. Pahor and S. Carrato, "A fuzzy approach to mouth corner detection," *Proceedings. 1999 International Conference on Image Processing*, Vol. 1, pp. 667-671, 1999.

[131]. M. S. Su, C. Y. Chen and K. Y. Cheng, "An automatic construction of a person's face model from the person's two orthogonal views," *Proceedings of Geometric Modeling and Processing*, pp. 179-186, 2002.

[132]. P. Viola and M. J. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 511-514, 2001.

[133]. R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," Proceedings. International Conference on Image Processing, Vol. 1, pp. I-900 - I-903, 2002.

[134]. A. M. Martinex and R. Benavente, "The AR face database," *CVC Technical Report #24*, 1998.

[135]. J. Vleugels and Remco C. Veltkamp, "Efficient image retrieval through vantage objects," *Pattern Recognition*, Vol. 35, No. 1, pp. 69-80, 2002.

[136]. T. Riklin-Raviv and A. Shashua, "The quotient image: Class based recognition and synthesis under varying illumination conditions," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 23-25, 1999.

[137]. W. Gao; S. Shan; X. Chai and X. Fu, "Virtual face image generation for illumination and pose insensitive face recognition," *Proceedings. International Conference on Multimedia and Expo*, Vol. 3, pp. III-149-52, 2003.

[138]. S. Shan; W. Gao; B. Cao and D. Zhao, "Illumination normalization for robust face recognition against varying lighting conditions," *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pp. 157-164, 2003