Abstract—Face Recognition (FR) has a wide range of applications, such as face-based video indexing, human-computer interaction, and surveillance system. The recognition rate can reach more than 90% with the frontal face, while dealing with faces are oriented or rotated, the accuracy just achieves 40% or 50%. Hence, an algorithm is proposed in this paper to deal with the problem. The face pose is first determined and then a frontal face pose is synthesized by wire frame model (WFM) according to the face pose parameter. In the FR system, Gabor filters are adopted for feature extraction, while the subspace based algorithms, principal component analysis (PCA) and linear discriminant analysis (LDA), are used for dimension reduction. The 2-norm distance is applied for similarity at last. With synthesized frontal face pose information, it can be seen from experimental results that the recognition rate is greatly improved.

Keywords—face recognition; wire frame model; principal component analysis; linear discriminant analysis;

I. INTRODUCTION

Face Recognition (FR) has a wide range of applications, such as face-based video indexing, human-computer interaction, and surveillance system. The face recognition system provides a direct way for person identification. Various algorithms for face recognition have been proposed, such as [1]-[6]. However, the performance of face recognition system varies greatly while dealing with the frontal poses and random poses. The verification accuracy for various faces is just 40% or 50%, while the accuracy is more than 90% for the frontal face pose [7]. It can be seen that the conventional algorithms get better recognition results when the face pose is frontal. However, in dynamic recognition system such as surveillance system, most of the faces captured from sequence images are in various face poses. Hence, in order to improve the accuracy of face recognition system under different face poses, the pose estimation and synthesis is a very important step in face recognition system.

The estimation of head pose including location and orientation is an important task for applications in several areas, such as human-robot interaction [8], visual attention of focus recognition [9][10], face recognition [11][12], chin curve estimation [13], and face replacement system [14]. For face recognition, it is known that the differences caused by changes of pose are much larger than the differences caused by the face appearances [15]. The performance of face recognition drops dramatically when large pose variations are presented in the input face images, especially when training face images of various face poses are not sufficient. When it comes to human machine interface applications, face pose estimation is an important cue of where the person is directing attention [16]. It is also a principal component of driver inattention monitoring system [17][18]. Hence, it can be seen that most applications require pose estimation which is robust to large pose.

Since facial features extraction is an important and primary step for systems such as interaction systems and identification systems, lots of algorithms and methods have been proposed for extracting the facial features. There are also many existing methods for pose estimation. Hence, the position of facial features such as eyes, lips, nose, and chin contour are supposed to be known as prior information. Most work on face pose estimation can be divided into two main categories: appearance-based methods and feature-based methods. In the appearance-based methods, the pose estimation problem is viewed as a pattern classification problem by using linear subspace decomposition and other non-linear variants to model appearances of different face poses. Fisher manifold learning is presented to estimate the head pose between two training views, which is a nonlinear interpolation method [19]. In [20], the independent component analysis (ICA) based approach is used to learn subspace representations of different face poses. Appearance-based methods attempt to recover the relationship between face poses and its appearance by statistical learning algorithms [21][22]. It can be used for large pose angles with sufficient training images for each face pose angle. On the other hand, feature-based methods focus on the pose-specific properties of certain points on the face, such as eyes, nose tips, and lips. These facial features are represented as feature templates [23] and can be connected to form pose-specific attribute graphs or bunch graphs [24]. After facial feature points are detected, the face pose parameters under
orthographic projection can be recovered [25]. Symmetry of the face has also been used to compute the rotation angles [26]. These geometry-based methods have some advantages, such as more intuitive and more accurate. The relation between the 3D face pose and the 2D face image has been clearly proved by projection model. Besides, with enough facial feature points, the noises will be reduced. Hence, the proposed method in this paper is based on the geometry-based method.

Gabor filters have been successfully used in image representation [27] and face recognition [28-30]. The Gabor transformed face images exhibit strong characteristics of spatial locality, scale and orientation selectivity. The most important properties are related to invariance to illumination, rotation, scale, and translation. These properties are based on the fact that they are all parameters of Gabor filters themselves [31]. The Gabor wavelet representation, therefore, captures salient visual properties. The 2-D Gabor filters are good models of the simple cells in the mammalian visual system [32]. This fact has promoted the use of Gabor filters for several years and is still mentioned in almost every article on the subject [33]. In this paper, Gabor filters based features are also adopted for face recognition.

The Principal component analysis (PCA) and linear discriminant analysis (LDA) are two powerful tools for feature extraction and data representation [34]. Among various FR methods used, the most popular are the so-called appearance based approaches. These methods include Eigenfaces [35] and Fisherfaces [36]. These approaches are either extension or combination of PCA and LDA, and have been widely used in a variety of areas, such as pattern recognition, computer vision, machine learning, and neural networks. With focusing on statistical feature extraction, the appearance-based approaches generally operate directly on appearance of face images and process them as two-dimensional holistic patterns to avoid difficulties, such as shape or landmark detection [37].

This paper is organized as follows: Sec. II shows the algorithm of face pose determination. Face recognition, which is based on Gabor features and PCA-LDA features, will be introduced in Sec. III. The results will be showed in Sec. IV and a conclusion will be given in Sec. V.

II. FACE POSE ESTIMATION

Face pose is an important pre-processing step in many face analysis problems, such as face recognition. In this section, a face pose estimation method based on 3D virtual face model is proposed. For an input image containing target face, 3 different rotation degrees, namely roll angle, tilt angle, and yaw angle, can be determined. There are three main stages in the proposed method. In the first stage, the transformation matrix is found by analyzing the virtual 3D face models and corresponding 2D projection face images in the database. By using the information of the coordinates of facial feature points in input face image, roll angle can be firstly obtained from the sloping degree of two eyes or the lips. Then, rotate the input image so that the line crossing eyes and the line crossing lips are parallel to x-axis. This normalization task is done in the second stage. In the last stage, the tilt angle and yaw angle are derived. We approximate the 3D face data of real input face image by the virtual face model with a combination of shift, scale, and rotation. The geometric ratio between 2D input real face image and simulated virtual face image is used to estimate the scale factor. With the scale factor, the rotation matrix derived by yaw angle and tilt angle can be solved. Hence, for an input face image with known facial features, these three rotation angles can be found.

A. Modeling

In this section, the model of projecting 3D points to 2D plane is introduced. The projection model derived from perspective projection projects 3D points onto the image plane along the lines that radiate from the projection center. The transformation matrix can be obtained by acquiring the parameters of camera and the geometric relationship between the 3D object and the projection center. However, the parameters of camera and the geometric relationship are not given in general case and are unable to be obtained from a single image. Hence, the transformation matrix cannot be obtained. Since the transformation matrix contains important information about the rotation angles, in this paper we propose another way to approximate the transformation matrix.

The three dimensional projection technique maps 3D points to 2D plane, which includes projection and transformation. Because most of the current displaying data are based on 2D media, the use of projection is widely spread. There are two types of projection: orthographic projection and perspective projection [38]. While orthographic projections are often used to show precise and detail measurements of 3D object, the perspective projection treats the 2D projection as being viewed through a camera viewfinder. The camera's position, orientation, and field of view control the behavior of the projection transformation. By utilizing the OpenGl application programming interface, the transformation matrix can be solved by corresponding 3D and 2D facial feature points. Because virtual 3D face data is much easier to manipulate than real face data, we will facilitate virtual 3D face data and corresponding 2D projection data to find out the transformation matrix in this section. This transformation matrix will be used to approximate the transformation matrix of camera later.

With the data of 3D virtual face and corresponding 2D face image, for each of the point \( x \) on 3D virtual face, there is a matrix \( M \) that projects and transforms the 3D points to corresponding point \( y \) on 2D plane:

\[
y = Mx,
\]

where

\[
y = \begin{bmatrix} x_{h_0} \\ y_{h_0} \\ z_{h_0} \\ 1 \end{bmatrix}, \quad x = \begin{bmatrix} x_{h_0} \\ y_{h_0} \\ z_{h_0} \\ 1 \end{bmatrix}, \quad y = \begin{bmatrix} x_{h_0} \\ y_{h_0} \\ z_{h_0} \\ 1 \end{bmatrix},
\]

are corresponding 3D and 2D points described in homogeneous coordinate system, and \( M \) is a \( 3 \times 4 \) matrix with certain parameters, such as camera parameters and relative geometric relationship. Each projected 2D face pose from a same face model can be viewed as a combination of scale, self-
orientation, translation, and projection. Hence, the matrix $M$ can be approximately decomposed by
\[
M = PTR_5R_4R_3S,
\] (3)
where P defined by
\[
P = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix},
\] (4)
is the $3 \times 4$ projection matrix which projects the 3D points to 2D plane. $T$ is a $4 \times 4$ transformation matrix which includes the relative geometric relationship between 3D points and view center, $R_5$, $R_4$, and $R_3$ are the rotation matrices functioned on the 3D points with three rotation angles, namely roll angle, tilt angle, and yaw angle respectively, and $S$ is a $4 \times 4$ scale matrix which works on 3D points. In this paper, we use six facial feature points, which are the two tips of two eyes and lips, in frontal face pose images and fewer facial feature points in profile face pose images to find the relationship between corresponding 3D and 2D points. To find the basic transformation matrix, the rotation matrices $R_5R_4R_3$ and the scale matrices are all set to be identity matrix $I_{4 \times 4}$. The corresponding basic transformation matrix $M_b$ is therefore described by
\[
M_b = PT.
\] (5)

When the face is frontal, six facial feature points are used to solve the basic transformation matrix as shown in Fig.1(a). When the face is profile, some of the facial feature points would be overlapped and therefore fewer feature points are available. Hence, two facial feature points, nose tip and chin tip, are added as facial features. The nose tip holds the property that it lays between the eyes and the lips in $y$-axis and is a corner of the face boundary, while chin tip is also a corner of the face boundary with lowest position in $y$-axis. These two facial feature points are added to ensure that there are at least four facial features to solve the transformation matrix, as shown in Fig.1(b).

![Figure 1](image1)

Figure 1. (a) The six facial feature points. (b) Two added facial feature points, nose tip $x_7$ and chin tip $x_8$, are used in profile face pose.

B. Face image normalization

The purpose of face image normalization is to rotate the face image so that the face is centrally located with zero roll angle, as shown in Fig.2. The roll angle in this paper is determined according to the number of eyes. When two eyes are found, the roll angle is defined as the included angle $\theta_{\text{eye}}$ between $x$-axis and the extended line across two eyes,
\[
\theta_{\text{eye}} = \tan^{-1}\left(\frac{y_{\text{eye}} - y_{\text{eye}}'}{x_{\text{eye}} - x_{\text{eye}}'}\right),
\] (6)
where $y_{\text{eye}}$ and $x_{\text{eye}}$ are the $y$-coordinate and $x$-coordinate of the right eye, while $y_{\text{eye}}'$ and $x_{\text{eye}}'$ are the $y$-coordinate and $x$-coordinate of the left eye. When there is only one eye in the face image, the roll angle is defined as the included angle between $x$-axis and the extended line across the right and left tips of the lips,
\[
\theta_{\text{lip}} = \tan^{-1}\left(\frac{y_{\text{lip}} - y_{\text{lip}}'}{x_{\text{lip}} - x_{\text{lip}}'}\right),
\] (7)
where $y_{\text{lip}}$ and $x_{\text{lip}}$ are the $y$-coordinate and $x$-coordinate of the right tip of lips, while $y_{\text{lip}}'$ and $x_{\text{lip}}'$ are the $y$-coordinate and $x$-coordinate of the left tip of lips. After the roll angle is retrieved, the face can be normalized to be horizontal by executing rigid transformation. The rigid transformation matrix $M_{\text{rig}}$ is defined by
\[
M_{\text{rig}} = \begin{bmatrix}
\cos \theta & -\sin \theta & 0 \\
\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{bmatrix}.
\] (8)

Bilinear interpolation using the four nearest neighbors of a point is adopted here to modify the rotated image. Let the point $(x', y')$ denote the coordinate of interpolated pixel and $v(x', y')$ denote the RGB value assigned to it. For bilinear interpolation, the assigned RGB value is given by
\[
v(x', y') = ax' + by' + cx'y' + d,
\] (9)
where four coefficients, $a$, $b$, $c$, and $d$, are determined from the four equations in four unknowns that can be written using the four nearest neighbors of point $(x', y')[39]$. 

![Figure 2](image2)

Figure 2. (a) Original image. (b) Rotated image with zero roll angle after normalization.
C. Rotation Model

After the roll angle is obtained and the input 2D face image is normalized, the yaw angle and tilt angle are going to be discussed here. There is a basic assumption that all the rotation angles are the relative face pose to view center. We do not consider in which way the body fronts and how the face is connected to the neck. This means, the face pose is decided and determined only by the way how the face looks relative to the view center. Hence, if a person is facing front with 0° yaw angle and a camera is on his right, the face pose would be determined with yaw angle 90° even though his face pose is frontal, as shown in Fig. 3. Under this assumption, the real face in 3D plane can be approximated as a combination of three stages: self-rotation, scale, and shift transformation on 2D image plane. Since the face is far away from the camera, the facial features such as eyes and lips on the frontal face surface are supposed to be on the same plane. Then, the distance between all facial features and camera can be viewed as a scale factor. Along with the assists of transformation matrix and the coordinates of facial feature points obtained from virtual 3D face data, the yaw angle and tilt angle can be solved.

As a result, the transformation matrix which transforms a 3D face into a face image in 2D plane can be viewed as a combination of stages. The real 3D facial feature points $x_r$ are approximated by different yaw angle, tilt angle, and scale:

$$x_r = \left( \frac{S_r}{S_{sf}} \right) R_x R_y R_z x_v,$$

$$y_r = s + PT x_v = s + \left( \frac{S_r}{S_{sf}} \right) M_b R_x R_y R_z x_v,$$

where $x_v$ are the virtual 3D facial feature points without any rotation and scale, $x_r$ are the real 3D facial feature points and $y_r$ are the corresponding facial feature points on 2D input face image, $R_x$ and $R_y$ are rotation matrix derived from tilt angle and yaw angle respectively, $M_b$ is the basic transformation matrix defined and obtained in (5), $S_r$ and $S_{sf}$ are the scale factor and basic scale factor respectively, and $s$ is a shift offset on 2D image plane owing to different geometric relationship between camera and face. Though the relative scale factor between 3D virtual face data and real 3D face is unknown, we can approximate the scale factor by the 2D geometric relationship between simulated virtual 2D face image and real 2D input face image.

Since the resolution and size may be different from each other, a scale factor $S_f$ is defined here to represent the relative size of face image or face region by comparing with the face image in data base. Though the skin color region is known as prior information, the size of skin color region is not used here to represent the size of face image since skin color region extracted by skin color may contain neck region. Besides, the hair may cover some face region when the face of input image is with long hair. Hence, the size of skin color region is not a proper parameter for face size. When the face pose is frontal, or two eyes are found, the scale factor $S_f$ is defined by

$$S_f = \sqrt{D_{e-e}^2 + D_{o-o}^2} = \sqrt{\|E_o - E_e\|^2 + \|P_e - L_c\|^2},$$

where parameter $D_{e-e}$ is the distance between two eyes, $E_e$ and $E_o$, and $D_{o-o}$ is the vertical distance from the centroid of the lips $L_c$ to the point $P_e$, which is the intersection of the line segment of two eyes and the vertical line crossing the centroid of the lips. These feature points are illustrated in Fig.4(a). From the statistical experiment results, the value of these two parameters, $D_{e-e}$ and $D_{o-o}$, are approximately proportional to the face size and nearly equal to each other when the face is frontal without turning left or right and looking up or down, while $D_{e-e}$ varies with the turning angle and $D_{o-o}$ varies when looking up or down. Hence, the scale factor $S_f$ is defined by using these two parameters. When the face pose is profile, or one eye is found, $S_f$ is defined with lesser facial information as

$$S_f = D_{e-e}^2 = \|E_o - L_c\|,$$

where $D_{e-e}$ is the distance between the only eye, $E_o$, and the centroid of lips, $L_c$, as shown in Fig.4(b).

Since the scale factor is estimated, the shift offset $s$ can be eliminated by
The unit vector in the direction of yaw angle, defined by

$$\mathbf{y}_e = s + \frac{S_i}{S_j} \mathbf{R}_i \mathbf{R}_j \mathbf{R}_i \mathbf{x}_e,$$

$$\mathbf{y}_r = s + \frac{S_j}{S_i} \mathbf{R}_j \mathbf{R}_i \mathbf{x}_r,$$

(13)

$$\left( \mathbf{y}_e - \mathbf{y}_r \right) = \frac{S_j}{S_i} \mathbf{R}_i \mathbf{R}_j \mathbf{R}_i \mathbf{x}_e - \mathbf{x}_r, \quad i \neq j.$$  

Hence, the rotation matrix is estimated by

$$\mathbf{M}_i \mathbf{R}_i \mathbf{R}_j \mathbf{R}_i = \left( \frac{S_j}{S_i} \right)^{i-1} \mathbf{Y}_o \mathbf{X}_i^T \left[ \mathbf{X}_i \mathbf{X}_j^T \right]^{-1},$$

where \( \mathbf{Y}_o \) and \( \mathbf{X}_o \) are the sets of distance between facial feature points on real 2D face image and virtual 3D face data, defined by

$$\mathbf{Y}_o = \left[ \mathbf{y}_o \right], \quad \mathbf{X}_o = \left[ \mathbf{x}_o \right], \quad i \neq j,$$

with

$$\mathbf{y}_o = \mathbf{y}_e - \mathbf{y}_r,$$

$$\mathbf{x}_o = \mathbf{x}_i - \mathbf{x}_j.$$

Suppose the face pose is decomposed of yaw angle and tilt angle in sequence, the rotation matrix \( \mathbf{R}_i \) with yaw angle \( \theta_i \) is defined by

$$\mathbf{R}_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

(17)

and \( \mathbf{R}_i \) is the rotation matrix with tilt angle \( \theta_2 \) about an axis in the direction of yaw angle, defined by

$$\mathbf{R}_2 = \begin{bmatrix} u_i^2 + (1-u_i^2)c & u_i u_j (1-c)-u_j s & u_i u_j (1-c)+u_j s & 0 \\ u_i u_j (1-c)+u_j s & u_i^2 + (1-u_i^2)c & u_i u_j (1-c)-u_j s & 0 \\ u_i u_j (1-c)-u_j s & u_i u_j (1-c)+u_j s & u_i^2 + (1-u_i^2)c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

(18)

where \( s \) and \( c \) denotes \( \sin \theta \) and \( \cos \theta \) respectively, and \( \mathbf{u} \) is the unit vector in the direction of yaw angle, defined by

$$\mathbf{u} = \begin{bmatrix} u \sin \theta_i \\ u \cos \theta_i \\ u \end{bmatrix} = \begin{bmatrix} \cos \theta_i \\ \sin \theta_i \\ 0 \end{bmatrix}.$$  

Hence, without using the homogeneous coordinate system, the relationship equation can be re-wrote by

$$\mathbf{M}_i \mathbf{R}_i \mathbf{R}_j \mathbf{R}_i = \frac{S_j}{S_i} \mathbf{Y}_o \mathbf{X}_i^T \mathbf{X}_i \mathbf{X}_j^T \mathbf{X}_j^{-1} \mathbf{X}_j,$$

where \( \mathbf{Y}_o \) and \( \mathbf{X}_o \) are the sets of distance between facial feature points on real 2D face image and virtual 3D face data, defined by

$$\mathbf{Y}_o = \left[ \mathbf{y}_o \right], \quad \mathbf{X}_o = \left[ \mathbf{x}_o \right], \quad i \neq j,$$

with

$$\mathbf{y}_o = \mathbf{y}_e - \mathbf{y}_r,$$

$$\mathbf{x}_o = \mathbf{x}_i - \mathbf{x}_j.$$

Suppose the face pose is decomposed of yaw angle and tilt angle in sequence, the rotation matrix \( \mathbf{R}_i \) with yaw angle \( \theta_i \) is defined by

$$\mathbf{R}_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

(17)

and \( \mathbf{R}_i \) is the rotation matrix with tilt angle \( \theta_2 \) about an axis in the direction of yaw angle, defined by

$$\mathbf{R}_2 = \begin{bmatrix} u_i^2 + (1-u_i^2)c & u_i u_j (1-c)-u_j s & u_i u_j (1-c)+u_j s & 0 \\ u_i u_j (1-c)+u_j s & u_i^2 + (1-u_i^2)c & u_i u_j (1-c)-u_j s & 0 \\ u_i u_j (1-c)-u_j s & u_i u_j (1-c)+u_j s & u_i^2 + (1-u_i^2)c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

(18)

where \( s \) and \( c \) denotes \( \sin \theta \) and \( \cos \theta \) respectively, and \( \mathbf{u} \) is the unit vector in the direction of yaw angle, defined by

$$\mathbf{u} = \begin{bmatrix} u \sin \theta_i \\ u \cos \theta_i \\ u \end{bmatrix} = \begin{bmatrix} \cos \theta_i \\ \sin \theta_i \\ 0 \end{bmatrix}.$$  

Hence, without using the homogeneous coordinate system, the relationship equation can be re-wrote by

$$\mathbf{M}_i \mathbf{R}_i \mathbf{R}_j \mathbf{R}_i = \frac{S_j}{S_i} \mathbf{Y}_o \mathbf{X}_i^T \mathbf{X}_i \mathbf{X}_j^T \mathbf{X}_j^{-1} \mathbf{X}_j,$$

where \( \mathbf{Y}_o \) and \( \mathbf{X}_o \) are the sets of distance between facial feature points on real 2D face image and virtual 3D face data, defined by

$$\mathbf{Y}_o = \left[ \mathbf{y}_o \right], \quad \mathbf{X}_o = \left[ \mathbf{x}_o \right], \quad i \neq j,$$

with

$$\mathbf{y}_o = \mathbf{y}_e - \mathbf{y}_r,$$

$$\mathbf{x}_o = \mathbf{x}_i - \mathbf{x}_j.$$
the trial solutions satisfies the Metropolis criteria, the trial solution replaces the current optimal solution. After several iterations, SA [41] slowly focuses on the most promising area. While SA acquires the optimal solution, the system satisfies the iteration limits, the algorithm will be terminated.

III. FACE POSE SYNTHESIS AND FACE RECOGNITION

It is known that the face pose is one of the most important factors to the face appearance. To overcome this problem, a synthesized frontal face pose image is needed for the recognition system. In this section, the face pose synthesis algorithm is introduced, and the Gabor features, PCA, and LDA feature extraction method are also detailed.

A. Face Pose Synthesis

There are two main steps in face synthesis. First, all the corresponding facial feature point pairs are located. Then, image warping method is adopted in every corresponding triangular mesh with wire frame model (WFM). In the first stage, suppose a 3D frontal face model is composed of \( N \) triangles and each of the triangular is consisted of 3 vertices which are also facial feature points, these facial feature points can be represented by a combination of rotation, scale, shift, and projection after being projected to 2D image plane, as shown in Fig. 6.

These projected facial feature points on 2D image plane can be viewed as the facial feature points in an input face image. Hence, all the corresponding facial feature points in an input image can be retrieved if these parameters in (10) are known. These parameters including rotation matrix, scale matrix, shift matrix, and projection matrix are all detailed and solved in section II. Therefore, we do not need to label all the facial feature points either in training phase or in testing phase. After the corresponding facial feature points are located, the image warping is exhibited in each triangular. The input image is warped to be frontal face pose by the affine transformation. The affine transformation is subject to geometric distortion introduced by perspective irregularities. By specifying to where 3 coordinate pairs from the source image re-map in the target image, the affine transformation in \( i \)-th triangular can be defined by

\[
\mathbf{T}_0 = \mathbf{M}_i \mathbf{S}_i, \quad \forall i = 1, 2, ..., N,
\]

where \( \mathbf{S}_i \) is the set of pixels within the \( i \)-th triangular in the source image (input image), \( \mathbf{S}_i \) is the set of pixels within the \( i \)-th triangular in the target image (the projected image of 3D frontal face model in the database), and \( \mathbf{M}_i \) is the affine transformation matrix of \( i \)-th triangular, which can be estimated by the 3 coordinate pairs of vertices

\[
\mathbf{M}_i = \mathbf{T}_s \mathbf{S}_i \mathbf{S}_s^{-1}, \quad \forall i = 1, 2, ..., N,
\]

where \( \mathbf{T}_s \in \mathbb{R}^3 \) and \( \mathbf{S}_s \in \mathbb{R}^3 \) are the set of 3 vertex coordinates of \( i \)-th triangular in target image and source image respectively. The value of pixels in target face is therefore determined by the value of input pixel or by the bilinear interpolation of neighboring four pixels.

B. Gabor Features

Gabor features are extracted by using Gabor filters and they include image’s local and discriminating information. The Gabor wavelets which capture the properties of spatial localization, orientation selectivity and spatial selectivity seem to be a good feature extractor for face recognition. The Gabor filters can be defined as \([42][43]\):

\[
\Psi_{\mu,\nu}(z) = \frac{k_{\mu,\nu}}{\sigma} \int e^{-\frac{1}{2\sigma^2} \sum_{k=1}^{2} \left( x_k - z_k \right)^2} e^{-\frac{\phi_{\mu,\nu}}{2\sigma^2} - \frac{\phi_{\mu,\nu}}{2\sigma^2}} dz,
\]

where \( \mu \) and \( \nu \) represent the orientation and scale of the Gabor kernels, \( z = (x, y) \), \( \| \| \) denotes the norm operator, and \( k_{\mu,\nu} \) is the wave vector and defined by

\[
k_{\mu,\nu} = k_x e^{i \phi_{\mu,\nu}},
\]

where \( k_x = k_{max} / f \) and \( \phi_{\mu} = \pi \mu / 8 \) with \( f \) is the spacing factor between kernels in the frequency domain. For the parameters of Gabor filters, four scales \( \nu \in \{0, 1, 2, 3\} \) and four orientations \( \mu \in \{0, 1, 2, 3\} \) are used in this paper, as shown in Fig. 7.

C. Principal Component Analysis (PCA)

Principal component analysis (PCA) have been very successful in image representation, analysis, compression, modeling, and recognition. This method performs dimensional reduction to the input dataset while retaining characteristics of
the dataset that contribute most to its variance by eliminating the insignificant principal components. The goal of PCA is to extract features that can best approximate the data. Suppose the training data is \( A = \{ x_1, x_2, \ldots, x_N \}, x_i \in \mathbb{R}^d \), PCA algorithm takes \( A \) as input for the sample covariance matrix \([44]\):

\[
S = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T ,
\]

where

\[
\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i ,
\]

is the mean of all sample vectors. A \( m \)-dimensional subspace is constructed by the eigenvectors of \( S \) corresponding to the first \( m \) highest eigenvalues. This linear subspace achieves least reconstruction error of \( A \) among all \( m \)-dimensional subspaces.

D. Linear Discriminant Analysis (LDA)

Linear discriminant analysis (LDA) has been one of the popular techniques used for face recognition. The basic idea is to calculate the optimal discriminant vectors such that the ratio of the between-class scatter and the within-class scatter is maximized \([45]\). The goal of the LDA is also to find an efficient way to represent the face vector space. PCA constructs the face subspaces by using the whole face training data as a whole without using the face class information. However, LDA uses class information which best discriminates among classes \([46]\).

Suppose the between class scatter matrix \( S_b \) is defined as:

\[
S_b = \sum_{i=1}^{k} N_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T ,
\]

and the within class scatter matrix \( S_w \) is defined as follow:

\[
S_w = \sum_{i=1}^{k} \sum_{j=1}^{N_i} (x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)^T ,
\]

where \( x_{ij} \) is the \( n \)-dimensional pattern \( j \) from class \( i \), \( N_i \) is the number of training patterns from class \( i \), and \( g \) is the total number of classes. The vector \( \bar{x}_i \) is unbiased sample mean from class \( i \). The mean vector \( \bar{x} \) is defined by:

\[
\bar{x} = \frac{1}{N} \sum_{i=1}^{k} \sum_{j=1}^{N_i} x_{ij} ,
\]

where \( N \) is the total number of samples \([47]\).

The main objective of LDA is to find a projection matrix \( P_{opt} \) to maximize the ratio of the determinant of the between class scatter matrix to the determinant of the within class scatter matrix:

\[
P_{opt} = \arg \max_p \left| \frac{P^T S_p P}{P^T S_w P} \right| .
\]

IV. EXPERIMENTAL RESULTS

In this section, we will demonstrate the improvement and validity of proposed methods in this paper. By estimating the face pose with large face pose variations, a better face can be easily synthesized. This face image is going to be used for face recognition, and the face recognition rate is referenced as the correctness and validity of the proposed system. We will compare the proposed method with pure face recognition system based on PCA-LDA and face recognition system using synthesized frontal face pose image which is based on active appearance model (AAM). The face recognition system contains three stages: feature extraction stage with Gabor features, dimension reduction stage with PCA-LDA method, and classification stage by calculating the similarity.

The input face images contain three types of face images. The first kind of face images is composed of frontal face pose images with slight pose and expression variations. This kind of images is used to illustrate the best face recognition rate. The second kind of face images contains frontal face pose and some certain types of profile face poses, which are in yaw direction only. The importance of synthesized frontal face pose image can be seen from the experimental results. The last kind of face images involves various kinds of face poses, including frontal and profile face pose in yaw direction, looking up and down face pose in tilt direction, clockwise and counter-clockwise face pose in roll direction, and a combination of those face poses.

A. Frontal Face Pose Images

For the first kind of input face images, the frontal face pose images with slight face pose and expression variations are considered. The experimental results of these images give us an ideal and best recognition rate. The input images come from internet and are in two different classes. The images in the first class are frontal face images with unobservable changes in face pose and expressions. There are 51 individuals in the first class and each of them contains 10 images, which are 510 images in total. Hence, 5 images are used for training and the other 5 images are used as testing for each individual. These images are represented in gray level and the size of each image is 64×64. The average face recognition rate of this class is 96.8%.

The images in the second class are frontal face pose images with slight face pose variations less than 20 degrees. There are 40 individuals in this class and each of them contains 10 images. In the training stage, 5 images including pose variations are used, while 5 images are used in the testing stage. These images are represented in gray level and the size of each image is 92×112. The average face recognition rate of this class is 88%.

B. Profile Face Pose Images

For profile face pose images, the CMU-PIE database of human face images are used as input profile face pose images \([48]\). The synchronized multi-camera imaging system, namely the CMU 3D room, can capture multiple images of each person simultaneously from multiple fixed poses. With these cameras, 13 poses of each person can be obtained simultaneously. Among these cameras, 9 of the 13 cameras are positioned at roughly head height in an arc from approximately a full left
profile to a full right profile. Each neighboring pair of these 9 cameras is therefore approximately 22.5° apart. Of the remaining 4 cameras, 2 are placed above and below the frontal camera, and 2 are placed in the corners of the room. Some of the face images including frontal face pose images and profile face pose images are illustrated in Fig.8 and these face images are arranged according to the position of camera.

![Face images](image1)

**Figure 8.** The corresponding face images according to the position of cameras.

There are 70 individuals in the CMU-PIE database and each of them contains 13 images. Some of them are randomly picked for training, while the rest of them are used for testing. The face recognition rates of pure face recognition system, face recognition based on synthesized frontal face pose from AAM, and from the proposed face pose estimation method in this paper are compared. The number of training images varies from 5 to 10. These training images are in RGB color space and the size of each image is 360×360. We first consider 5 random face images for training as before. The comparison of recognition rate is shown in Fig.9. From the experimental results, it can be seen that different face poses greatly influence the recognition rate. Without a synthesized frontal face pose image, a poor recognition result, about 38.5% in average, will be obtained when using traditional face recognition system. A synthesized frontal face pose image can offer a direct way for recognition instead of being influenced by various face pose conditions. As it can be seen from the experimental results, the synthesized frontal face pose image based face recognition systems hold better recognition rate, such as 71.5% and 79.5% in average for AAM and proposed method respectively. On the other hand, the number of training images also influences the recognition result. A higher recognition rate is reached if there are sufficient training images. Because the proposed method in this paper uses the wire frame model with face pose parameters, the recognition rate is still acceptable when large face pose variation is presented, as shown in Fig. 9.

![Face recognition results](image2)

**Figure 9.** The face recognition rate under different yaw degrees when using 5 training images.

When the training images are few, from the experimental results we can see that the AAM based face pose synthesis system failed to approach face pose precisely, while the proposed method based on 3D face model which is not restricted by the number of training face images provides satisfying results. The recognition rate of different number of training images is shown in Fig.10 and Fig.11, and the comparison is shown in Fig.12. It can be seen that the recognition rate is improved as the number of training images increase. The average recognition rate is 78.2% and 81.6% for AAM based face recognition when 8 and 10 training images are used respectively, while the proposed method holds a better face recognition rate, 81% and 83%. Hence, from the experimental results, it can be seen that the proposed method offers an acceptable face recognition rate even when training images are not sufficient and large face pose variation are presented.

![Face recognition results](image3)

**Figure 10.** The face recognition rate under different yaw degrees when using 8 training images.

![Face recognition results](image4)

**Figure 11.** The face recognition rate under different yaw degrees when using 10 training images.
In this paper, we propose a face pose estimation and face pose synthesis based face recognition system, which is robust to large face pose variations. Besides, when the training samples are few, the proposed system can still maintain an acceptable average recognition rate. This is because the face pose synthesis is done by wire frame model and 3D face model transformation, while the traditional face pose synthesis methods rely on the training samples to approach the real face pose. Hence, it can be seen from the experimental results that the proposed system is more reliable and robust when large face pose variation is represented and training samples are few.

V. CONCLUSIONS

In this paper, we propose a face pose estimation and face pose synthesis based face recognition system, which is robust to large face pose variations. Besides, when the training samples are few, the proposed system can still maintain an acceptable average recognition rate. This is because the face pose synthesis is done by wire frame model and 3D face model transformation, while the traditional face pose synthesis methods rely on the training samples to approach the real face pose. Hence, it can be seen from the experimental results that the proposed system is more reliable and robust when large face pose variation is represented and training samples are few.

REFERENCES


