

國立交通大學

資訊科學與工程研究所

碩士論文

利用自動化影像形變與邊緣偵測於人臉辨識  
**Automatic Morphing and Edge Map for Face  
Recognition**



研究生：古蕙嬪

指導教授：李素瑛

中華民國九十六年六月

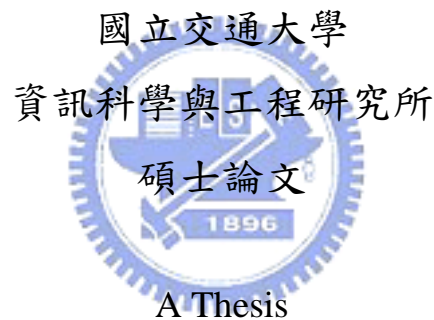
利用自動化影像形變與邊緣偵測於人臉辨識  
**Automatic Morphing and Edge Map for Face  
Recognition**

研究生：古蕙嬪

Student：Hui-Zhen Gu

指導教授：李素瑛

Advisor：Suh-Yin Lee



Submitted to Institute of Computer Science and Engineering  
College of Computer Science  
National Chiao Tung University  
in Partial Fulfillment of the Requirements  
for the Degree of Master in  
Computer Science  
June 2007  
Hsinchu, Taiwan, Republic of China

中華民國 九十六 年 六月


# 利用自動化影像形變與邊緣偵測於人臉辨識

## Automatic Morphing and Edge Map for Face Recognition

學生：古蕙嬪 指導教授：李素瑛教授

國立交通大學資訊科學與工程研究所

### 摘要



人臉辨識是近年來很受矚目的議題。主向量分析是眾多成功的人臉辨識方法之一，但是當有明顯的光線與姿態變化時，主向量分析的辨識率不夠準確。許多相關研究提出利用多張訓練樣本來解決光線與姿態變化的問題。本論文提出一套以主向量分析方法為基礎的人臉辨識系統，稱作自動化形變與邊緣偵測辨識器，讓我們可以在有光線以及姿勢變化的情況下，只用一張訓練樣本就可以更正確地辨識人臉的身份。我們的構想是重新描繪一個不受人臉姿勢影響的標準參考模型，並使用不易受光線變化影響的人臉邊緣影像，來作人臉的辨識。我們使用ORL資料庫來驗證系統的有效性。實驗結果證實，只使用一張訓練樣本，利用主向量分析方法，此自動化形變與邊緣偵測辨識器的確可以在有光線與姿勢變化的情況下，得到更佳的辨識效果。

檢索詞：形變、自動化對應、邊緣偵測、人臉辨識

# Automatic Morphing and Edge Map for Face Recognition

STUDENT: Hui-Zhen Gu

ADVISOR: Prof. Suh-Yin Lee

Institute of Computer Science and Information Engineering

National Chiao-Tung University

## ABSTRACT

Face recognition has received much attention during the past several years. Principal component analysis (PCA) is one of the most successful methods for face recognition but it is not highly accurate when the illumination and pose of the facial images vary considerably. Many researches have discussed some solutions to solve the illumination and pose problems, but most of them need multiple training images. This paper presents a novel face recognition system based on PCA, named Automatic Pose normalization and Edge map face Recognizer (APER). The idea is to automatically re-render a pose invariant reference model to accommodate varying pose of the images. Face edge images, which are insensitive to illumination changes, are incorporated. The APER requires only a single face image for training per person. The APER and the PCA method are evaluated using ORL database. The experimental results demonstrate that the APER can improve the performance of conventional PCA approach under varying pose and illumination with single training image.

**Keyword: morphing, automatic correspondence, edge detection, face recognition**

# ACKNOWLEDGEMENT

I sincerely appreciate the kind guidance of my advisor, Prof. Suh-Yin Lee. She encouraged me in exploiting research topics freely and enthusiastically helped me. Without her graceful suggestions and encouragement, I cannot complete this thesis. Besides, thanks are extended to all the members in the Information System Laboratory, especially Mr. Ming-Ho Hsiao, Mr. Yi-Wen Chen, Mr. Hua-Tsung Chen and Mr. Hsuan-Sheng Chen. They gave me a lot of suggestions and shared their experiences. Finally, I want to express my appreciation to my parents for their support. This thesis is dedicated to them.



## Table of Contents

摘要.....	i
<b>ABSTRACT .....</b>	<b>ii</b>
<b>ACKNOWLEDGEMENT.....</b>	<b>iii</b>
<b>Table of Contents .....</b>	<b>iv</b>
<b>Lists of Figures .....</b>	<b>v</b>
<b>Lists of Tables .....</b>	<b>vi</b>
<b>Chapter 1. Introduction .....</b>	<b>1</b>
1.1. Background.....	1
1.2. Research objectives and contributions.....	2
1.3. System overview .....	3
1.4. Organization .....	4
<b>Chapter 2. Related Work.....</b>	<b>5</b>
2.1 Review of the PCA method.....	5
2.2 Pose problem approaches.....	7
2.3 Illumination problem approaches.....	9
<b>Chapter 3. Pose Normalization.....</b>	<b>11</b>
3.1. View angle problem .....	11
3.2. Image metamorphosis .....	12
3.3. Pose normalization .....	14
<b>Chapter 4. Edge Map.....</b>	<b>16</b>
4.1. Edge map advantage .....	16
4.2. Sobel edge detector .....	17
4.3. Slight shift problem.....	18
4.4. Level mask .....	19
<b>Chapter 5. Automatic Correspondence Methodology .....</b>	<b>21</b>
5.1 Image registration .....	21
5.2 Automatic correspondence methodology.....	23
5.3 Connected component correspondence .....	24
5.4 Control point correspondence .....	31
5.4.1 Control point extracting.....	31
5.4.2 Vector difference method.....	33
<b>Chapter 6. Experimental Results .....</b>	<b>37</b>
6.1 Recognition results of slight shift problem .....	38
6.2 Recognition results under varying pose .....	39
6.3 Recognition results under varying pose and lighting.....	41
<b>Chapter 7. Conclusions and Future Work .....</b>	<b>45</b>
<b>Reference .....</b>	<b>47</b>

## Lists of Figures

<b>Fig. 1-1</b> The system overview.....	4
<b>Fig. 2-1</b> A simplified version of face space.....	7
<b>Fig. 3-1</b> Pose variation: (a) out-of-plane variation (b) in-plane variation.....	12
<b>Fig. 3-2</b> An example of image metamorphosis with interpolation parameter $t$ .....	13
<b>Fig. 3-3</b> A coordinate mapping with single line pair.....	13
<b>Fig. 3-4</b> Pose normalization generation.....	15
<b>Fig. 4-1</b> The same face appears differently under different illuminations.....	16
<b>Fig. 4-2</b> The convolution kernel $G_x$ and $G_y$ of the Sobel edge detector on vertical and horizontal orientations.....	18
<b>Fig. 4-3</b> The slight shift of the edge.....	19
<b>Fig. 4-4</b> An example of the Level Mask when $k=3$ .....	20
<b>Fig. 4-5</b> Edge detection generation.....	20
<b>Fig. 5-1</b> An image mosaic example.....	22
<b>Fig. 5-2</b> A correspondence example performed manually.....	23
<b>Fig. 5-3</b> Automatic correspondence flowchart.....	24
<b>Fig. 5-4</b> Corresponding component algorithm.....	25
<b>Fig. 5-5</b> A corresponding component algorithm example.....	27
<b>Fig. 5-6</b> A vector difference method example.....	34
<b>Fig. 5-7</b> Corresponding component set.....	35
<b>Fig. 5-8</b> Automatic morphing results.....	36
<b>Fig. 6-1</b> Recognition curves based on a PNR, APNR, and PCA.....	41
<b>Fig. 6-2</b> Illumination modification examples.....	41
<b>Fig. 6-3</b> Recognition curves based on PER, APNR, APER, and PCA.....	44

## Lists of Tables

<b>Table 2.1</b>	The comparison of related works on face problem approaches.....	10
<b>Table 5.1</b>	The components sorting results $S_X$ , $S_Y$ , $S'_X$ , and $S'_Y$ .....	28
<b>Table 5.2</b>	The corresponding candidate when $p=1$ .....	29
<b>Table 5.3</b>	The distributions of component 15 in $I_1$ and component 12 in $I_2$ .....	29
<b>Table 5.4</b>	The distributions of component 15 in $I_1$ and component 13 in $I_2$ .....	30
<b>Table 5.5</b>	The distributions of component 15 in $I_1$ and component 18 in $I_2$ .....	30
<b>Table 5.6</b>	.The corresponding component algorithm result of Fig. 5-5.....	31
<b>Table 6.1</b>	Subsystem table.....	37
<b>Table 6.2</b>	Recognition result of slight shift problem with single training image...38	
<b>Table 6.3</b>	Recognition results under varying pose with single training image.....	39
<b>Table 6.4</b>	Recognition results under varying pose with two training images.....	40
<b>Table 6.5</b>	Recognition results under varying pose with three training images .....	40
<b>Table 6.6</b>	Recognition results under varying pose with four training images .....	40
<b>Table 6.7</b>	Recognition results under varying pose with five training images .....	40
<b>Table 6.8</b>	Recognition results under varying pose and illumination with single training image. ....	42
<b>Table 6.9</b>	Recognition results under varying pose and illumination with two training images.....	42
<b>Table 6.10</b>	Recognition results under varying pose and illumination with three training images. ....	43
<b>Table 6.11</b>	Recognition results under varying pose and illumination with four training images.. ....	43
<b>Table 6.12</b>	Recognition results under varying pose and illumination with five training images.....	43
<b>Table 7.1</b>	The comparison of Related works and APER .....	45



# Chapter 1

## Introduction

### 1.1. Background

Face recognition has recently received much attention, and it is used to identify one or more persons from still images or a video sequence by comparing input images with faces stored in a database [1]. Human can easily recognize the difference between two similar faces, but it is a significant challenge for computer. Typical applications of face recognition include commercial and law-enforcement surveillance systems. Although very reliable methods of biometric personal identification exist, e.g., fingerprint analysis and retinal or iris scans, these methods rely on the cooperation of the participants, whereas a face recognition system is often effective without the participant's cooperation or knowledge [2]. Therefore, face recognition is a more natural biometric identification method and easier to be accepted by human.

Principal component analysis (PCA) is a statistical approach, which has been widely used in applications such as face recognition and image compression. In PCA approach, face images are expressed as eigenvectors, and hence the eigenvectors are called eigenfaces [3]. However, PCA recognition method is not highly accurate when the illumination and pose of the facial images vary considerably [4]. The two important research issues of face recognition are illumination and pose problems [1]. The illumination variation problem is that the changes induced by illumination are larger than the differences between individuals, causing systems to misclassify input images [5]. The pose problem means that the performance of face recognition system drops

significantly when pose variations are present in the input images. When illumination variation and pose variation are both present, the task of face recognition becomes even more difficult. In order to solve the illumination and pose problems, many researches [6, 7, 8] gather multiple training images under varying illumination and pose conditions. However, in many cases, there is only one training image available per person. In this situation, the performance of these face recognition methods is not satisfactory.

## **1.2. Research objectives and contributions**

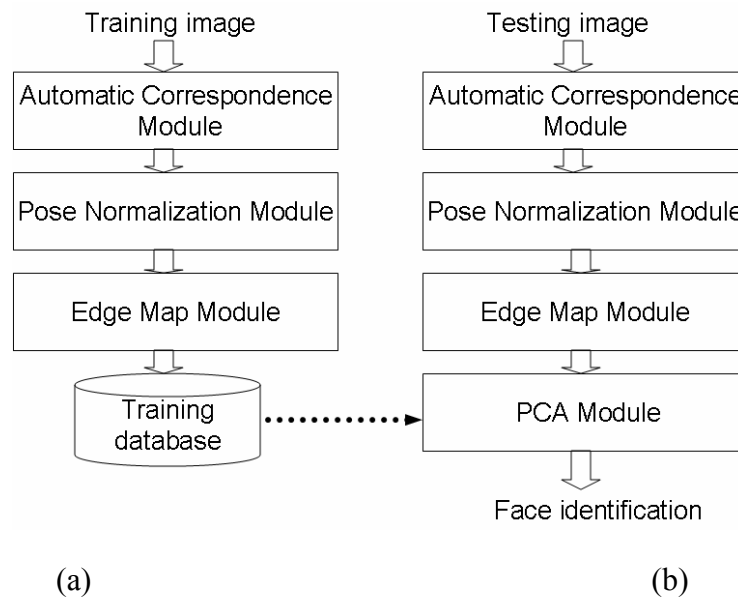
Although there have been a large number of approaches designed for face recognition, investigating accurate recognition under varying pose and illumination is still very challenging. Furthermore, if only a single training image is available, the task of face recognition becomes more difficult. This paper intends to achieve accurate recognition under varying pose and illumination with single training image. Hence, for each person, we transform a single training image into a standard reference model, which is an edge-based approximated frontal face of the person. For a testing or a query image, it is also transformed by the same way into a normalized image for recognition against the face image database. Based on the standardized reference model images, we can recognize the identifications of querying images accurately under not well-controlled environment.

We propose a system named Automatic Pose Normalization and Edge Map Face Recognizer (APER). The APER requires only a single training image per person to construct the standard reference model. Because human face has symmetric property, we can predict a symmetric view of the single face image by mirror transform. With these two symmetric view images, the reference model can be constructed. The APER contains three modules: automatic correspondence module, pose normalization module

and edge map module. The aim of the automatic correspondence module is to describe the correspondence between the two symmetric view images. The pose normalization module synthesizes two different perspective view images by interpolation into an approximated frontal face. Finally, the edge map module uses edges to describe the contour of face without complexion. Complexion is sensitive to light change, but edge is not. Therefore, the APER accommodates the effect of pose and illumination variation with single training image.

### **1.3. System overview**

There are two phases in the system, the training phase and the testing phase, as depicted in Fig. 1-1. For the single training image per person, a model frontal face image is created by the pose normalization module, which is proposed to overcome the aspect angle problem effectively. Then an edge map image of this model frontal face is acquired by edge map module. Edge map is less sensitive in illumination variation condition and more computation efficiency. Because the morphing technique requires control lines, which is often extracted manually, an automatic correspondence algorithm has been derived to enhance the efficiency the pose normalization process. Furthermore, the edge map image is stored into the training database. For an input query face image, the normalized face is acquired and its corresponding map is captured in the same way. Then the edge map image of the query image is fed into PCA module for recognition against the training database.



**Fig. 1-1** The system overview (a) Training process (b) Recognition process

## 1.4 Organization

This paper is organized as follows. In Chapter 2, we discuss the related works of pose problem approaches, illumination problem approaches and the PCA method. Chapter 3 presents the pose normalization module using image metamorphosis technique to re-render a normalized face for each input image. Chapter 4 illustrates the edge map module to acquire the edge map image for PCA recognition module. Chapter 5 proposes the automatic correspondence methodology to extract the corresponding control points between the input image and its reference image automatically. Finally, the experimental results of the proposed system and conclusions are drawn in Chapter 6 and Chapter 7.

# Chapter 2

## Related Work

In Chapter 2, we introduce the background knowledge for principal component analysis (PCA) method and some previous related works in pose and illumination problem approaches. In Section 2.1, we present the PCA method, which has been used for representing, detecting, and recognizing faces, called *eigenfaces* [3]. In the Section 2.2 and Section 2.3, some related works to overcome pose problem and illumination problem are described.

### 2.1 Review of the PCA method

This section introduces the PCA method [9], which has been widely used in applications such as face recognition and image compression. PCA is a common technique for finding patterns in data, and expressing the data as eigenvector to highlight the similarities and differences between different data. The following steps summarize the PCA process.

1. Let  $\{D_1, D_2, \dots, D_M\}$  be the training data set. The average Avg is defined by:

$$Avg = \frac{1}{M} \sum_{i=1}^M D_i \quad (1)$$

2. Each element in the training data set differs from Avg by the vector  $Y_i = D_i - Avg$ .

The covariance matrix Cov is obtained as:

$$Cov = \frac{1}{M} \sum_{i=1}^M Y_i \cdot Y_i^T \quad (2)$$

Since the covariance matrix Cov is square, we can calculate the eigenvectors and eigenvalues for this matrix.

3. Choose  $M'$  significant eigenvectors of Cov as  $E_K$ 's, and compute the weight vectors  $W_{ik}$  for each element in the training data set, where  $k$  varies from 1 to  $M'$ .

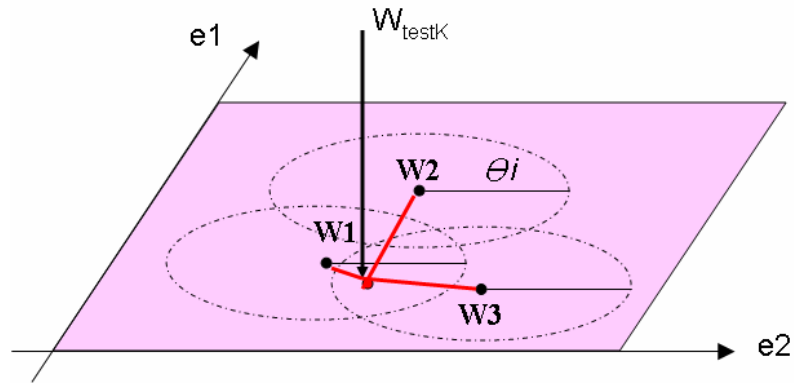
$$W_{ik} = E_k^T \cdot (D_i - Avg), \quad \forall i, k \quad (3)$$

Based on PCA, many face recognition techniques have been developed, such as *eigenfaces* [3]. The following steps summarize the *eigenface* recognition process:

1. Initialization: Acquire the training set of face images  $I_1, I_2, \dots, I_M$ . Calculate each face difference vector from the average face Avg by Eq.(1), and the covariance matrix Cov is obtained by Eq.(2). Then compute the eigenvectors  $E_k$  of Cov, which define the face space. Finally, compute the weights  $W_{ik}$  by Eq.(3) for each image in the training set.
2. Input querying: When a new testing face image is encountered, calculate a set of weights  $W_{testK}$  depending on the same steps mentioned above. The weights  $W_{testk}$  forming a vector  $T_p = [w_1, w_2, \dots, w_{M'}]^T$  describes the contribution of each *eigenface* in representing the input face image
3. Recognition: A simplest technique to classify the weight pattern is to compute the minimum distance of  $W_{testK}$  from  $T_p$ . It means that the test image can be classified to be in class  $p$  when  $\min(D_p) < \theta_i$ , where  $D_p = \|W_{testK} - T_p\|$  and  $\theta_i$  is the threshold.

Fig. 2-1 shows a simplified version of face space to illustrate the projecting results of three training face images  $W_1, W_2, W_3$  and a testing image  $W_{testk}$ . We can recognize  $W_{testk}$  as one of the three known individuals  $W_1, W_2$  and  $W_3$  by the projecting distance between  $W_{testk}$  with each training images. In this case, there are two *eigenfaces*  $e_1, e_2$  to construct the face space. The distance between  $W_{testk}$  and  $W_2$  is larger than the

threshold  $\Theta_i$ , they are not considered to be the same person consequently. Furthermore, the projecting location of  $W_{\text{test}k}$  in the face space is more close to the projecting location of  $W_1$  than  $W_2$ . Therefore, we believe that  $W_{\text{test}k}$  and  $W_1$  are the same person.



**Fig. 2-1** A simplified version of face space

## 2.2 Pose problem approaches

Over the last few years, numerous algorithms have been proposed for face recognition including geometric feature-based methods [10, 11, 12, 13], and holistic matching methods [14, 15, 16, 17]. Geometric feature-based methods use geometric properties and relations (e.g., distances and angles) between facial features such as eyes, mouth, nose, and chin to perform recognition. Holistic matching methods use the whole face region as the raw input to a recognition system. Many of these successful face recognition methods suffer from an important drawback: these methods are successful in terms of recognition performance only under well-controlled environment. However, face recognition in an uncontrolled environment is still very challenging. The difficulty has been documented in the FERET test and FRVT test reports [18][19] : pose variation problem and illumination variation problem. The problem of recognizing a human face from a general view remains largely unsolved, because transformations such as position, orientation, and scale cause the face's appearance to vary substantially. Researchers

have proposed various methods for handling the pose problem. We illustrate the most common pose problem approaches below.

A view-based eigenspace method [20] is proposed for face recognition under variable pose. The method defines a small number of face classes for each known person corresponding to characteristic views. The system attempts to recognize many slightly different views, at least one of which is likely to fall close to one of the characteristic views. This method explicitly codes the pose information by constructing an individual eigenface for each pose. The view-based formulation allows for recognition under varying head orientations and the modular description allows for the incorporation of important facial features such as eyes, nose and mouth. These extensions account for variations in face pose and lead to a more robust recognition system under varying pose condition.

An active appearance model (AAM) [21] uses a statistical model of object to capture the shape and appearance of an image. The method demonstrates a small number of 2D statistical models matched by the active appearance model from any viewpoint. The models can be used to estimate head pose, to track faces through large angles of head rotation and to synthesize faces from unseen viewpoints. These appearance models are trained on multiple example images labeled with sets of landmarks.

By these pose problem approaches, when we encounter a testing image with variable pose condition, we can select the most suitable model and choose the one which achieves the best match. Once a model is selected and matched, we can estimate the head pose, and thus track the face, switching to a new model if the head pose varies significantly. However, these pose problem approaches have two common drawbacks:



(1) they need many example images to cover the range of possible views; (2) the illumination problem is not explicitly addressed.

### **2.3 Illumination problem approaches**

The effect of variation in illumination is often larger than the variation between individuals, causing systems based on image comparison to misclassify input images. Significant illumination change can seriously degrade the performance of face recognition [18, 19]. Hence it is necessary to seek methods that compensate for illumination changes. In general, the illumination problem is quite difficult and has received consistent attention recently. In the case of face recognition, many approaches to this problem have been proposed and the two typical illumination approaches: Heuristic methods [14, 22, 16] and Illumination cone [23, 8] are introduced.

Many existing systems [14, 22] use heuristic methods to compensate for lighting changes. Contrast normalization is utilized to preprocess the detected faces, and then histogram equalization is applied. Histogram equalization [24] is an image processing operation to compensate for imaging effects due to changes in illumination brightness and differences in camera response curves. Through this adjustment, the intensities can be better distributed on the histogram and the recognition works well for images under different lighting conditions [16]. The method is useful in images with backgrounds and foregrounds that are both bright or both dark.

The other illumination problem approach is illumination cone [23, 8], which has been proposed as an effective method of handling complex illumination variations, including shadowing and multiple light sources. This method is an extension of the 3D linear-subspace method and also requires at least three aligned training images acquired under different lighting conditions. Using multiple training images of each face taken

with different lighting directions, the shape and albedo of the face can be reconstructed. Recently, image synthesis based on illumination cones has been proposed to handle both pose and illumination problems in face recognition. To handle variations due to rotation, it needs to completely resolve the GBR (generalized-bas-relief) ambiguity and then reconstruct the Euclidean 3D shape [23]. The drawback of the illumination cone method is that it requires multiple training images and a lot of computation. The related works comparison of pose and illumination problem approaches is shown in Table 2.1.

**Table 2.1** The comparison of related works on face problem approaches

Face problem approaches	Accommodate pose problem	Accommodate illumination problem	Single training image
View-based eigenface	✓	X	X
Active appearance model	✓	X	X
Heuristic method	X	✓	✓
Illumination cone	✓	✓	X

# Chapter 3

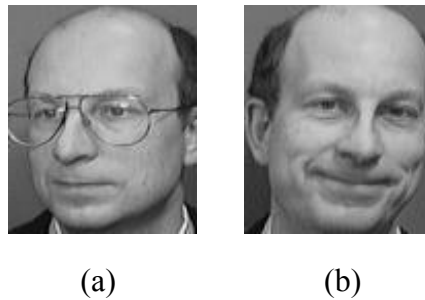
## Pose Normalization

Pose normalization is important in face recognition. We utilize the image based rendering technique: image metamorphosis [25] to accommodate view angle problem in face recognition. In this chapter, we introduce the view angle problem in Section 3.1 and the image metamorphosis is presented in Section 3.2. In Section 3.3, we propose the pose normalization procedure and results.

### 3.1 View angle problem

The PCA method for face recognition is not very effective under the condition of varying pose. The performance of face recognition system drops significantly when pose variations are present in the input images. The difficulty has been documented in the FERET test and FRVT test reports [18, 19] and suggested as a major research issue. The pose variations are divided into out-of-plane rotation problem and in-plane rotation problem as shown in Fig. 4-1. The in-plane rotation is the poses in which the face is rotated to the left or right only without much change in the plane of sight. Out-of-plane rotation represents the poses in which the face is rotated out of the plane of sight looking left or right [26]. Over the past few years, a considerable number of studies have been made on the rotation problem. Numerous algorithms of class-based approaches have been proposed. For each person, some view classes corresponding to characteristic views are defined first, and then the tested image closest to the characteristic view is recognized [6]. This solution requires many example images to cover the range of possible views. In addition, no matter how many view classes are

defined, there always exists the possibility that at least one image does not belong to any view class.



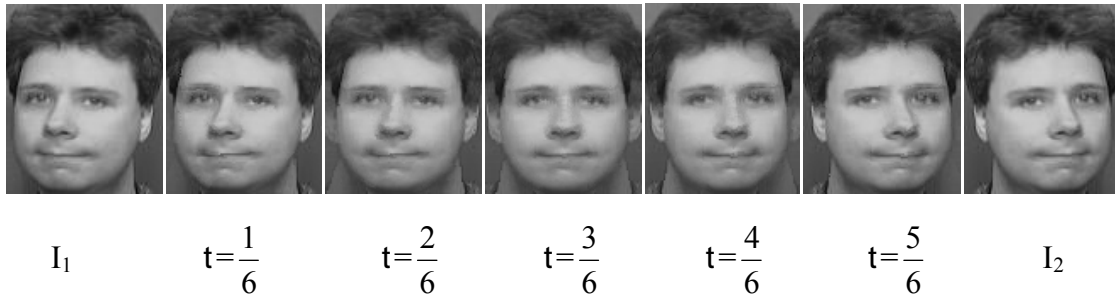
**Fig. 3-1** Pose variation: (a) out-of-plane variation (b) in-plane variation

We use image metamorphosis technique to re-render a pose invariant reference model to accommodate both in-plane and out-of-plane pose problem. The advantage of pose normalization is that the images, training or querying are normalized, at the central line. After the normalization processing, all of the images are treated with exactly the same pose. Hence, the recognition rate of pose-invariant PCA method can be improved. However, the extreme case of pose (for example, almost some features of one side of face lost) is not our focus in this paper.

### 3.2 Image metamorphosis

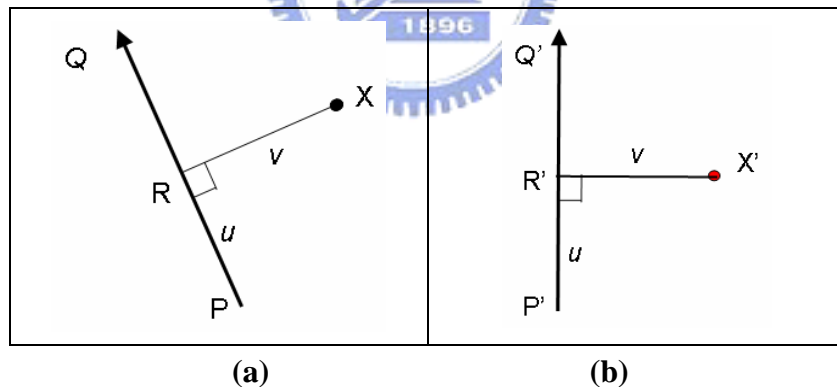
Metamorphosis is an image processing technique typically used as an animation tool to specify a warp from the first image  $I_1$  to the second  $I_2$ . For at least a decade, this approach is called “morphing” and we use “morphing” for the remainder of this paper. Morphing is the process of continuously transforming a source image  $I_1$  into a target image  $I_2$  in terms of different interpolation parameter  $t$ , as illustrated in Fig. 3-2. As the morphing proceeds, it changes one digital image into another by cross-dissolve between them. Cross-dissolve is a weighted combination of two images controlled by a single interpolation parameter  $t$  as in Eq.(4):

$$I = t \times I_1 + (1-t) \times I_2 \quad (4)$$



**Fig. 3-2** An example of image metamorphosis with interpolation parameter  $t$

A morphing operation can blend two images: the source image  $I_1$ , and the destination image  $I_2$  and create a new image  $I$ . To cross-dissolve the corresponding position accurately, we need to acquire corresponding feature lines in  $I_1$  and  $I_2$ . Each corresponding line is used to interpolate the new position between  $I_1$  and  $I_2$  by Beier-Neely's Field Morphing Technique [25]. We take a coordinate mapping example with single line pair shown in Fig. 3-3 to illustrate the technique.



**Fig. 3-3** A coordinate mapping from the (b) destination image pixel coordinate to the (a) source image pixel coordinate with single line pair.

A pair of corresponding lines  $PQ$  and  $P'Q'$  in the source and destination images respectively defines a coordinate mapping from the destination image pixel coordinate  $I_2$  to the source image pixel coordinate  $I_1$ .  $X'$  is the location to sample the destination image for the pixel at  $X$  in the source image. The distance from  $X$  to line  $PQ$  is  $v$ . The nearest point to  $X$  along the line  $PQ$  is  $R$ , and  $u$  is defined to be the distance from  $P$  to  $R$ .

For each  $X$  in the source image,  $X'$  is the point with  $v$  pixels away from  $P'Q'$ , and is at the same side of  $X$  to  $PQ$ . Similarly, the nearest point  $R'$  to  $X'$  along  $P'Q'$  is the point with distance  $u$  from  $P'$  to  $R'$ .  $X'$  can be defined in Eq. (5):

$$X' = P' + \frac{u \cdot (Q' - P')}{\|Q' - P'\|} + \frac{v \cdot \text{Perpendicular}(Q' - P')}{\|Q' - P'\|} \quad (5)$$

where  $\text{Perpendicular}(V)$  returns the vector perpendicular to the input vector  $V$ , and of the same length as the input vector.

When multiple pairs of lines are considered, the weight of the coordinate transformations for each corresponding line is calculated. The weight is determined by the distance from  $X$  to the line  $PQ$ . The closer the pixel to the line, the higher the weight is. In other words, the farther the pixel to the line, the lower the weight is. After the above processes, a new image where shape and texture lied in between  $I_1$  and  $I_2$  is created as shown in Fig. 3-2.



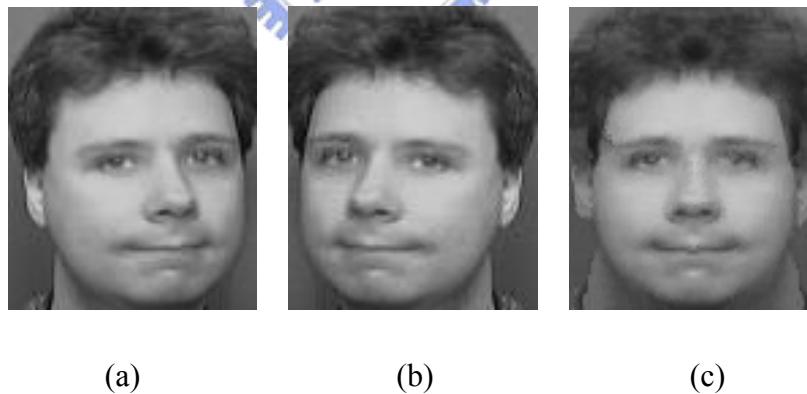
### 3.3 Pose normalization

For pose normalization, we aim to create an approximated frontal face image as a standard reference image for each person. So, we adopt a single input image  $I_1$  as the source image, and its mirrored image  $I_2$  as the destination image. An example of the source image and the expected destination image is shown in Fig. 3-4(a) and Fig. 3-4(b). It is supposed that human faces are symmetric in horizontal orientation but asymmetric in vertical orientation. The benefit of horizontal mirroring is that only one single image as input is needed to obtain the symmetric destination image.

The pose normalization module synthesizes two different perspective view images by interpolation into an approximated frontal face. Using the input image and the

mirrored image, we take the interpolation with parameter  $t = 0.5$  in Eq. (4). The new synthesized image by morphing will be an approximated frontal face. An example of morphing result is shown in Fig. 3-4(c). Face photos usually have slight displacement or rotation. By morphing technique we can normalize the face to the central line of the input image. Based on the normalization, we can recognize the identifications of querying images with pose variation. Therefore, the pose problem can be accommodated by image morphing.

The pose normalization module in the proposed system can be used both in training phase and testing phase. Many other face recognition systems [20, 21, 22] artificially select the image, which is closest to frontal view to be training image. In our system, we no longer have the frontal restriction on choosing the training images. Because every input image will be normalized, we can randomly choose an image as training image, except the extreme cases.



**Fig. 3-4.** Pose normalization generation (a) the original image (b) the mirror image of the original image (c) the normalized face image in frontal view using morphing.

# Chapter 4

## Edge Map

The edge map uses edges to describe the contour of face without complexion. Complexion is sensitive to light change, but edge is not. In this chapter, we introduce the advantage of edge map in Section 4.1, and show how edge map accommodates the general case of illumination problem. The edge detector used in the APER is Sobel edge detector which is presented in Section 4.2. However, we find that the edge image acquired by the Sobel edge detector arises slight shift problem, In Section 4.3, we propose the Level Mask designed for making up the deficiency of the Sobel edge detector.



### 4.1 Edge map advantage

Besides the pose problem, the PCA method is still not highly accurate when the illumination of the facial images varies considerably [4]. The illumination problem is illustrated in Fig. 4.1, where the same face appears different due to a change in lighting.



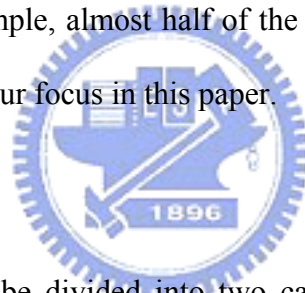
**Fig. 4-1** The same faces appear differently under different illuminations (from the ORL face database with illumination modification)



Among the many approaches to this problem, class-based methods [14, 16, 22] gather multiple images of the same face in a fix pose but under different lighting conditions to construct a low-dimensional representation of images. The *illumination cone* method [8, 23] models the complete set of images of an object with *Lambertian reflectance* under an arbitrary combination of point light sources at infinity.

However, in many cases, we cannot acquire multiple training images under different illumination. The idea in our APER system adopts edge map in a single image to cover the illumination problem instead of collecting multiple images. Edge image describes the contour of face, and the edge is less sensitive to the lighting variation. However, our method is under the assumption that edges can be found. The extreme case of illumination (for example, almost half of the face is under the shadow and the edge cannot be found) is not our focus in this paper.

## 4.2 Sobel edge detector



The edge detectors can be divided into two categories [27]: gradient filter and Laplacian filter. In our system, we choose a common gradient filter Sobel edge detector to find edges. Because an edge is normally defined as an abrupt change in intensity, the Sobel Edge Detector uses two simple *convolution kernel*  $G_x$  and  $G_y$ , shown in Fig. 4-2 to detect the intensity change on vertical and horizontal orientations. If a pixel value and its eight neighbor pixel values are the same, then the sum of the convolution kernel masks  $\sum G_x$  and  $\sum G_y$  are both zero, which implies there is no edge on the pixel. So, if  $\sum G_x \neq 0$ , the Sobel edge detector detects an edge point on vertical orientation. Similarly, if  $\sum G_y \neq 0$ , the Sobel edge detector detects an edge on horizontal orientation. After the process of Sobel edge detector, the image such as Fig. 4-5(a) will be transformed into Fig. 4-5(b).

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

**G<sub>x</sub>**

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

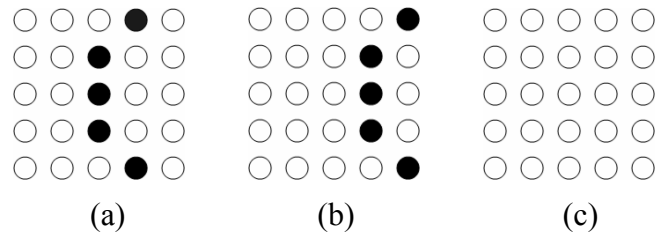
**G<sub>y</sub>**

**Fig. 4-2** The convolution kernel G<sub>x</sub> and G<sub>y</sub> of the Sobel edge detector on vertical and horizontal orientations.

### 4.3 Slight shift problem

The Sobel edge detector indicates the edge position of the face image and creates a Sobel edge image which is not influenced by varying illumination. However, when we feed the Sobel edge image directly into the PCA module for recognition, the Sobel edge image still cannot be recognized correctly. This is because even two similar images of the same person have a very slight pose shift. Fig. 4-3 depicts an example of the slight pose shift. In Fig. 4-3, a circle represents a pixel point on the Sobel edge image. Black circle designates the pixel point which is on the edge; otherwise it is not on the edge. Compare the three figures shown in the Fig. 4-3. Most of us would accept that (a) is more similar to (b) than to (c). However, the difference between Fig. 4-3 (a) and (b) is 10 pixels which is larger than the difference between Fig. 4-3 (a) and (c), which is 5 pixels. This example suffices to show that the recognition result of Sobel edge image will be influenced by the amount of edges. Testing images will be recognized to the training image with fewest edges. When the pose on the face image takes a pixel shift to the right, the whole contour edge will together shift a pixel to the right. It is a very slight change on the image which is hardly perceived. Because PCA is a low-dimensional representation, this kind of slight pose shift will dramatically influence the PCA recognition result. We call this kind of problem the slight shift

problem. Obviously, the PCA recognition method is too sensitive to the slight shift problem on Sobel edge image.



**Fig. 4-3** Slight shift of the edge (a) the original position of the edge, (b) the new position of the edge after a slight shift (c) image without edge

#### 4.4 Level mask

In order to use edge information to recognize face correctly, we design a mask called Level-Mask to modify the Sobel edge image into Level-Masked Sobel (LMS) image. The Level-Mask is used to blur the Sobel edge image. We re-set the k-nearest pixel values of each edge point on the edge image. Level k means the distance between an edge point and its neighbor pixels. A suitable level k can be chosen. For an edge pixel at position  $a$ , each neighboring pixel  $b$ , if it is not on an edge, is re-set to the value  $f(a, b)$  as defined in Eq.(6):

$$f(a, b) = \begin{cases} \frac{k - dist(a, b)}{k} \times value(a), & \text{if } value(b) = 0 \\ value(b), & \text{if } value(b) \neq 0 \end{cases} \quad (6)$$

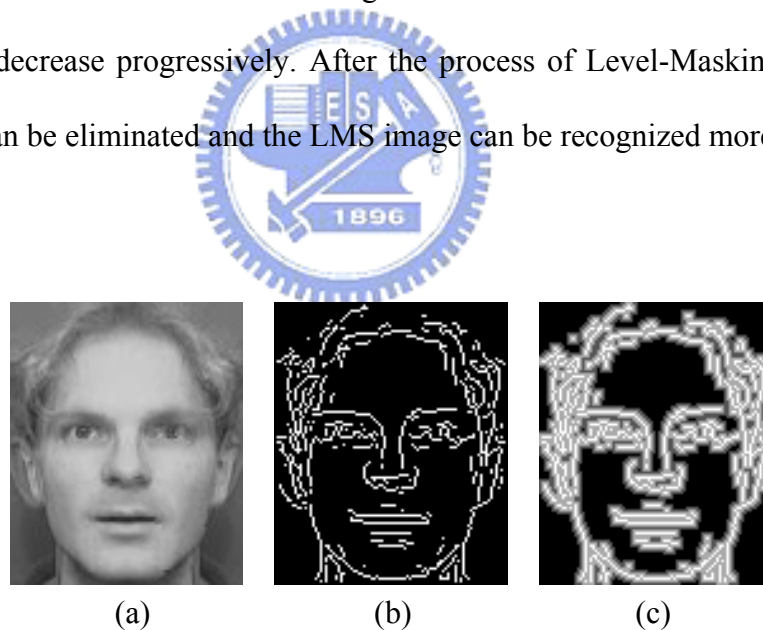
$dist(a, b)$  is the distance between  $a$  and  $b$ , and  $value(b)$  is the pixel value of pixel  $b$ . After transforming original image into the edge image, each edge pixel value is either 0 or 255. If the pixel  $b$  is on the edge whose  $value(b)$  is 255, we do not re-set the  $value(b)$ . Otherwise, the pixel  $b$  is not on the edge and we will re-set the  $value(b)$  according to the distance to the edge point  $a$ . When the level number  $k$  is 3 and the neighbor pixel values are 0, the neighbor pixels on the first level are re-set to  $((3-1)/3) \times 255$  and the sixteen neighbor pixels on the second level are reset to  $((3-2)/3) \times 255$ . The 3-level mask

is shown in Fig. 4-4. Using the Level-Mask, we can briefly calculate the neighbors on the first level as  $(2/3) \times 255$ , and  $(1/3) \times 255$  on the second level, if the neighbor pixel value is 0. If the neighbor pixel value is not 0, then its pixel value is remained.

$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
$\frac{1}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{3}$
$\frac{1}{3}$	$\frac{2}{3}$	<b>1</b>	$\frac{2}{3}$	$\frac{1}{3}$
$\frac{1}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{3}$
$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$

**Fig. 4-4.** An example of the Level Mask when  $k=3$

Fig. 4-5 (c) shows the Level Mask Sobel (LMS) image after using Sobel edge detector and Level-Mask when  $k = 3$ . Edge will be thickened and the value of its neighbors will decrease progressively. After the process of Level-Masking, the slight shift problem can be eliminated and the LMS image can be recognized more correctly.



**Fig. 4-5** Edge detection generation; (a) original intensity input image (b) edge image by Sobel edge detector (c) LMS image by Level Mask

## Chapter 5

### Automatic Correspondence Methodology

As we mention before, pose normalization by morphing is proposed to overcome the aspect angle problem effectively. But it requires control lines, which are usually hand-marked to describe the correspondence between an input face image and its reference image. This will lead us further into the consideration of automatically extracting corresponding control lines. This kind of image processing operation is often called *image registration*. In this chapter, we introduce the related work of image registration in Section 5.1. We present our automatic correspondence methodology for face images in Section 5.2. Our methodology falls into two phases: connected components correspondence and control points correspondence, which are illustrated in Section 5.3 and Section 5.4 respectively.

#### 5.1 Image registration

In computer vision, corresponding points mapping is required for motion tracking, object recognition, shape reconstruction, and image mosaic. Fig. 5-1 shows an example of image mosaic. Before we mosaic two images together correctly, we need to make the two photographs in the same coordinate with correspondence position. Recently, the task of automatic extraction of intrinsic geometric information between an input face image and its reference image has received a great deal of attention. This image processing operation is called *image registration* [28].



**Fig. 5-1** An image mosaic example

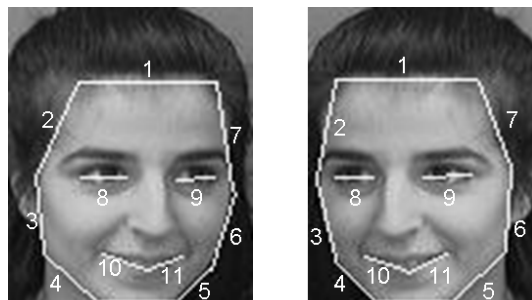
In the APER, we utilize the automatic correspondence module to enhance the efficiency of the pose normalization module. When the number of images in the database is large, pose normalization module takes a lot of time to find out the control lines manually. Therefore, we develop an algorithm to extract the corresponding control lines automatically. It makes the system totally automatic and more suitable on large database. Control point (CP) and intensity are the two basic features used for image registration in the literature [29]. Control points are pixel points, such as corners, intersections, points of locally maximum curvature, and centers of gravity of close-boundary regions in an image. In gray-level image, it is difficult to use intensity value to identify features, because the difference of the intensity is small. Hence, we choose control points to find out the correspondence in our automatic correspondence module.

For face images, feature detection is the most common approach on image registration. For example, a lot of eye localization algorithms mostly based on iris detection have been developed [30, 31, 32]. Besides, Wiskott et al. [33] used Gabor wavelets to find accurate correspondences between facial features in face recognition. The drawback of those approaches is that the feature detection has to define feature models per feature, and each feature extraction has its own correctness rate. If the non-suitable features are used, the chance of finding correspondence correctly is low. In

fact, our goal is to obtain the correspondences of images, and it doesn't matter what feature a correspondence belongs to. Therefore, we create our automatic correspondence algorithm which uses regular structure relationship between contours and facial features instead of feature modeling.

## 5.2 Automatic correspondence methodology

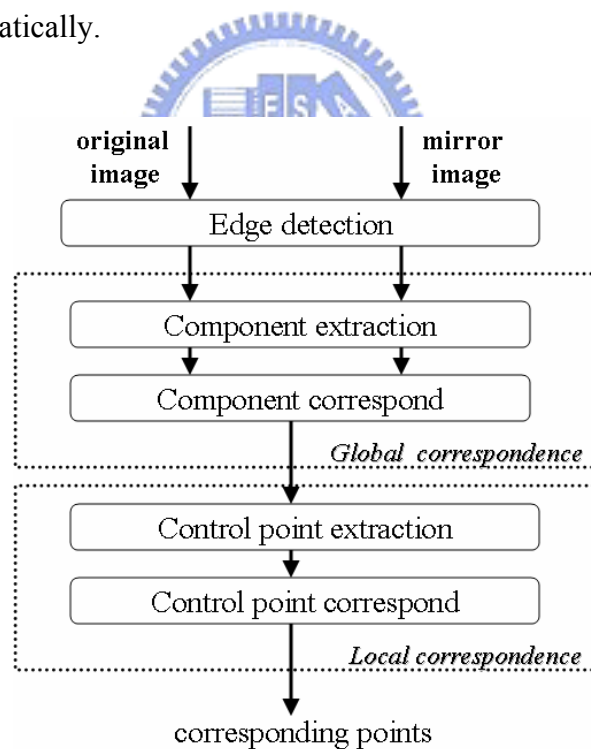
Fig. 5-2 shows a correspondence example which is performed manually. It is observed that human usually looks for the obvious points on contours and facial features, for example, forehead, jaw, eyes and mouth, as corresponding points. It is a difficult task for computer to provide corresponding points as correctly as human. Nevertheless, human face belongs to a special class, which possesses regular structure relationship. For example, eyes are below brows; nose is definitely above mouth. One other important characteristic for face images is that the structure of human face is almost symmetric in horizontal orientation. By these regular structure rules of human face, we can develop an automatic correspondence algorithm for face images.



**Fig. 5-2** A correspondence example performed manually, The same numbers on the two face images indicate the same feature lines.

We propose an automatic correspondence algorithm which extracts the corresponding lines between a single face image and its reference image. The algorithm contains five steps as depicted in Fig. 5-3. First of all, we detect the edge image of the input face image by Sobel edge detector. The Sobel edge detector is a discrete

differentiation operator, which computes an approximation of the gradient of the image intensity function [27]. After transforming the input face image into a Sobel edge image, we acquire the reference image of the Sobel edge image by mirroring ( $I_2(i,j) = I_1(\text{ImageWidth}-i, j)$ ). The Sobel edge image  $I_1$  of input face and its mirror image  $I_2$  are used as the inputs of the two important steps in the algorithm: connected component correspondence and control points correspondence which are illustrated in section 5.3 and section 5.4. After we find out the corresponding control points, we connect the corresponding control points to be corresponding lines. Finally, we obtain the corresponding lines between an input face image and its mirror image and the corresponding lines are fed into the next module in the proposed system for pose normalization automatically.



**Fig. 5-3** Automatic correspondence methodology flowchart

### 5.3 Component correspondence

To find out the corresponding contour and facial features between the face image



and its mirror image with regular structure relation, we first extract all connected components of the two Sobel edge images. For each pixel on the edges, if any of its eight neighbors is also on the edges, then this pixel and the current pixel are considered to be in the same connected component. After we scan the whole image, we can extract all connected components of the two images. We design a corresponding component algorithm as shown in the Fig. 5-4 which can find out the corresponding component of the input image and its reference image.

```

Initial:   Sx = sortX (Com); S'x = sortX (Com')
          Sy = sortY (Com); S'y = sortY (Com')
          Min_difference = ∞
For i=1,2,...,2M
  Acquire the index i', where Sy[i'] == Sy[i] and i' ≠ i
  Acquire the first index j, where Sx[j] == Sy[i]
  Acquire the second index j', where Sx[j'] == Sy[i]
  Select the set {S'y[i-p], ..., S'y[i+p]} as the candidate of corresponding component of Sy[i]
  For t=-p, -p+1, ..., p
    k=i+t
    Acquire the index k' which S'y[k'] == S'y[k] and k' ≠ k
    Acquire the first index l which S'x[l] == S'y[k]
    Acquire the second index l' which S'x[l'] == S'y[k]
    Difference_y = | k-i | + | k'-i' |
    Difference_x = | l-j | + | l'-j' |
    Difference_score = (difference_y)2 +(difference_x)2
    If(Difference_score < min_difference && difference_score < threshold)
      min_component1 = Sy[i]
      min_component2 = S'y[k]
      min_difference = difference_score
    end
  end
end
min_component1 and min_component2 are regarded as correspond components
end

```

**Fig. 5-4** Corresponding component algorithm

We have defined some notations for the component correspondence algorithm:

$$\mathbf{P} = \{(x,y) \mid 1 \leq x \leq \text{ImageHeight}, 1 \leq y \leq \text{Width}\}$$

$$X(p) = x, \text{ which is the } x \text{ value of point } p, p \in \mathbf{P}$$

$$Y(p) = y, \text{ which is the } y \text{ value of point } p, p \in \mathbf{P}$$

$$\mathbf{Com} = \{\text{com}_1, \text{com}_2, \dots, \text{com}_n\}; \mathbf{Com}' = \{\text{com}'_1, \text{com}'_2, \dots, \text{com}'_m\}$$

$$\text{com}_i = \{(p_{i_1}, p_{i_2}, \dots, p_{i_u}) \mid p_{i_1}, p_{i_2}, \dots, p_{i_u} \in \mathbf{P}\}$$

$$\text{com}'_j = \{(p'_{j_1}, p'_{j_2}, \dots, p'_{j_v}) \mid p'_{j_1}, p'_{j_2}, \dots, p'_{j_v} \in \mathbf{P}\}$$

$$M_{i\_xmax} = \max(X(p_{i_1}), X(p_{i_2}), \dots, X(p_{i_u})), \text{ where } (p_{i_1}, p_{i_2}, \dots, p_{i_u}) = \text{com}_i$$

$$M_{i\_ymax} = \max(Y(p_{i_1}), Y(p_{i_2}), \dots, Y(p_{i_u})), \text{ where } (p_{i_1}, p_{i_2}, \dots, p_{i_u}) = \text{com}_i$$

$$M_{i\_xmin} = \min(X(p_{i_1}), X(p_{i_2}), \dots, X(p_{i_u})), \text{ where } \{p_{i_1}, p_{i_2}, \dots, p_{i_u}\} = \text{com}_i$$

$$M_{i\_ymin} = \min(Y(p_{i_1}), Y(p_{i_2}), \dots, Y(p_{i_u})), \text{ where } \{p_{i_1}, p_{i_2}, \dots, p_{i_u}\} = \text{com}_i$$

$$M'_{j\_xmax} = \max(X(p'_{j_1}), X(p'_{j_2}), \dots, X(p'_{j_v})), \text{ where } \{p'_{j_1}, p'_{j_2}, \dots, p'_{j_v}\} = \text{com}'_j$$

$$M'_{j\_ymax} = \max(Y(p'_{j_1}), Y(p'_{j_2}), \dots, Y(p'_{j_v})), \text{ where } \{p'_{j_1}, p'_{j_2}, \dots, p'_{j_v}\} = \text{com}'_j$$

$$M'_{j\_xmin} = \min(X(p'_{j_1}), X(p'_{j_2}), \dots, X(p'_{j_v})), \text{ where } \{p'_{j_1}, p'_{j_2}, \dots, p'_{j_v}\} = \text{com}'_j$$

$$M'_{j\_ymin} = \min(Y(p'_{j_1}), Y(p'_{j_2}), \dots, Y(p'_{j_v})), \text{ where } \{p'_{j_1}, p'_{j_2}, \dots, p'_{j_v}\} = \text{com}'_j$$

$\text{sortX}(\mathbf{Com}) = (i_1, i_2, \dots, i_{2u})$ , which is the component number of the sorted result of

$$(M_{1\_xmax}, M_{1\_xmin}, M_{2\_xmax}, M_{2\_xmin}, \dots, M_{u\_xmax}, M_{u\_xmin})$$

$\text{sortX}(\mathbf{Com}') = (j_1, j_2, \dots, j_{2v})$ , which is the component number of the sorted result of

$$(M'_{1\_xmax}, M'_{1\_xmin}, M'_{2\_xmax}, M'_{2\_xmin}, \dots, M'_{v\_xmax}, M'_{v\_xmin})$$

$\text{sortY}(\mathbf{Com}) = (i_1, i_2, \dots, i_{2u})$ , which is the component number of the sorted result of

$$(M_{1\_ymax}, M_{1\_ymin}, M_{2\_ymax}, M_{2\_ymin}, \dots, M_{u\_ymax}, M_{u\_ymin})$$

$\text{sortY}(\mathbf{Com}') = (j_1, j_2, \dots, j_{2v})$ , which is the component number of the sorted result of

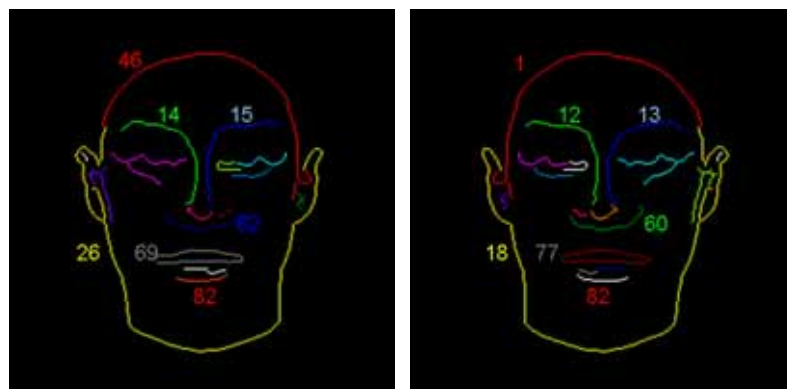
$$(M'_{1\_ymax}, M'_{1\_ymin}, M'_{2\_ymax}, M'_{2\_ymin}, \dots, M'_{v\_ymax}, M'_{v\_ymin})$$

Set  $\mathbf{P}$  contains all of the points in the image. Function  $X$  and function  $Y$  are used to get the  $x$  value and  $y$  value of point  $p$ . The connected components of the input image and its reference image are  $\mathbf{Com}$  and  $\mathbf{Com}'$ . Each of the  $\mathbf{Com}$  and  $\mathbf{Com}'$  contains a set of points. In addition, for each  $\text{com}_i$  and  $\text{com}'_j$ , we find the maximum and minimum value of both  $x$  value and  $y$  value of points in this component.

First of all, the component correspondence algorithm sorts the connected components of the Sobel edge image  $I_1$  and its mirrored image  $I_2$  in horizontal orientation to  $S_X$  and  $S'_X$ , and sorts them in vertical orientation to  $S_Y$  and  $S'_Y$ . For each component number in  $S_Y$ , we select several neighbor corresponding components in  $S'_Y$  as corresponding candidates and computes the difference score of these corresponding candidates. The difference score is the square sum of the index distances in vertical and horizontal orientations. The lower the difference score is, the more similar the structure relation is between the two corresponding candidates. It means that the two connected components very likely are the corresponding connected components. On the other hand, if the difference score is high, the two components cannot be corresponded. We acquire the corresponding candidates with lowest difference score and treat them as corresponding components. The corresponding components  $Corr$  are defined as:

$$Corr = \{ corr_1, corr_2, \dots, corr_n \}$$

$$corr_k = \{ (com_i, com'_j) \mid com_i \in Com, com'_j \in Com' \}$$



(a)

(b)

**Fig. 5-5** A corresponding component algorithm example

We take a simple example to illustrate the execution process of the corresponding component algorithm. In Fig. 5-5 (a) and (b), there are both seven connected

components of the two Sobel edge images  $I_1$  and  $I_2$ . First of all, we derive the four inputs  $S_X$ ,  $S_Y$ ,  $S'_X$ , and  $S'_Y$  shown in Table 5.1 from these connected components. For the connected component 15 in  $S_X$ , we can select three neighbor corresponding components 12, 13 and 18 in  $S'_X$  marked in Table 5.2 as corresponding candidates. Second, we compare the distribution of the three candidates 12, 13, 18 in  $I_2$  with the connected component 15 in  $I_1$ , shown in Table 5.3, 5.4 and 5.5 respectively. The difference between the distributions is based on the difference score, which is defined in corresponding algorithm shown in Fig. 5-4. The difference score of the candidate 12 is  $1^2+0^2+5^2+5^2=51$ , and the difference score of the candidate 13 is  $0^2+1^2+0^2+0^2=1$ , and the difference score of the candidate 18 is  $1^2+7^2+7^2+2^2=103$ . Finally, we compare the three difference scores; the difference score of the connected component 13 is much smaller than those of other candidates. This is because the distribution of the candidate 12 in  $I_2$  is most similar to the distribution of the connected component 15 in  $I_1$ . We take the connected component 15 in  $I_1$  and the connected component 13 as corresponding connected components consequently.

**Table 5.1** The components sorting results  $S_X$ ,  $S_Y$ ,  $S'_X$ , and  $S'_Y$

index	component image1		component image2	
	$S_Y$	$S_X$	$S'_Y$	$S'_X$
1	46	26	1	18
2	14	46	12	1
3	15	14	13	12
4	26	69	18	77
5	46	62	1	60
6	14	82	13	82
7	15	14	12	12
8	62	15	60	13
9	62	82	60	82
10	69	62	77	60
11	69	69	77	77
12	82	15	82	13
13	82	46	82	1
14	26	26	18	18

**Table 5.2** The corresponding candidate when  $p=1$

index	component image1		component image2	
	$S_Y$	$S_X$	$S_Y$	$S_X$
1	46	26	1	18
2	14	46	12	1
3	15	14	13	12
4	26	69	18	77
5	46	62	1	60
6	14	82	13	82
7	15	14	12	12
8	62	15	60	13
9	62	82	60	82
10	69	62	77	60
11	69	69	77	77
12	82	15	82	13
13	82	46	82	1
14	26	26	18	18

**Table 5.3** The distributions of component 15 in  $I_1$  and component 12 in  $I_2$

index	component image1		component image2	
	$S_Y$	$S_X$	$S_Y$	$S_X$
1	46	26	1	18
2	14	46	12	1
3	15	14	13	12
4	26	69	18	77
5	46	62	1	60
6	14	82	13	82
7	15	14	12	12
8	62	15	60	13
9	62	82	60	82
10	69	62	77	60
11	69	69	77	77
12	82	15	82	13
13	82	46	82	1
14	26	26	18	18

**Table 5.4** The distributions of component 15 in  $I_1$  and component 13 in  $I_2$

index	component image1		component image2	
	$S_Y$	$S_X$	$S_Y$	$S_X$
1	46	26	1	18
2	14	46	12	1
3	15	14	13	12
4	26	69	18	77
5	46	62	1	60
6	14	82	13	82
7	15	14	12	12
8	62	15	60	13
9	62	82	60	82
10	69	62	77	60
11	69	69	77	77
12	82	15	82	13
13	82	46	82	1
14	26	26	18	18



**Table 5.5** The distributions of component 15 in  $I_1$  and component 18 in  $I_2$

index	component image1		component image2	
	$S_Y$	$S_X$	$S_Y$	$S_X$
1	46	26	1	18
2	14	46	12	1
3	15	14	13	12
4	26	69	18	77
5	46	62	1	60
6	14	82	13	82
7	15	14	12	12
8	62	15	60	13
9	62	82	60	82
10	69	62	77	60
11	69	69	77	77
12	82	15	82	13
13	82	46	82	1
14	26	26	18	18

After the processing of the corresponding algorithm, the corresponding results are shown in the Table 5.6.

**Table 5.6** The corresponding component algorithm result of Fig 5-5

Correspond components table	
component image1	component image2
<b>46</b>	<b>1</b>
<b>14</b>	<b>12</b>
<b>15</b>	<b>13</b>
<b>26</b>	<b>18</b>
<b>62</b>	<b>60</b>
<b>69</b>	<b>77</b>
<b>82</b>	<b>82</b>

## 5.4 Control points correspondence

Control point (CP) and intensity are the two basic features used for image registration in the literature [29]. In gray-level image, it is difficult to use intensity value to identify features, because the difference of the intensity is small. Control points such as corners, road intersections, points of locally maximum curvature, and centers of gravity of close-boundary regions in an image can be used to find out the correspondence in gray-level image more accurately.

### 5.4.1 Control points extracting

In our automatic correspondence algorithm, we use end points and corners as control points to find out the correspondence between two face edge images  $I_1$  and  $I_2$ . In image processing and computer vision, corner detection has drawn a lot of attention in the past twenty years. Many corner detectors have been reported [34, 35, 36]. Because the total curvature of the gray level image is proportional to the second order

directional derivative in the direction of edge normal [37], the corner points of an image are defined as points where image edges have the maximum of absolute curvature. The trajectory of edge at any point  $(x,y)$  is described by the equation below:

$$B_1(x,y) = \alpha_1 \vec{x} + \beta_1 \vec{y} \quad (7)$$

$$B_2(x,y) = \alpha_2 \vec{x} + \beta_2 \vec{y} \quad (8)$$

$B_1, B_2$  are the motion vector defined by parameters  $\alpha_1, \beta_1, \alpha_2, \beta_2$  which are the relative distance from the last trajectory location and the next trajectory location of the edge point  $(x,y)$ .

A corner point is the one at which both the gradient magnitude and the rate of change of gradient direction are high. The algorithm looks for places where there are large curvatures along the edge. In other words, the algorithm finds the location  $(x,y)$  where the edge changes direction rapidly. Therefore, a corner response (CR) function is defined as:

$$CR(x,y) = B_2(x,y) - B_1(x,y) \quad (9)$$

If CR is greater than a predefined threshold, the pixel is a candidate corner that should be retained for processing in the subsequent steps. However, since the CR function is sensitive to noise, we will find some corners with local maximum but not global maximum of the CR function. Since noise increases the difficulty of locating corners, the CR function must be smoothed by using smoothing filter or low-pass filter in the frequency domain. We design a simple low-pass filter to pass the pixel points with low frequencies but to attenuate frequencies higher than the cutoff frequency. The low-pass filter designed provides a smoother form of a signal which removes the short-term oscillations of the orientation change, and keeps only the long-term trend



orientation change. Therefore, the CR function can be rewritten as follows:

$$CR'(x, y) = CR(w_1(x, y)) + 2CR(w_2(x, y)) + CR(w_3(x, y)) \quad (10)$$

$$w_1(x, y) = (x, y) - B_1(x, y)$$

$$w_2(x, y) = (x, y)$$

$$w_3(x, y) = (x, y) + B_2(x, y)$$

With the CR function, the control points can be selected out. For each control point  $cp_i$ , it belongs to a connected component. On the other hand, a connected component may contain several control points.

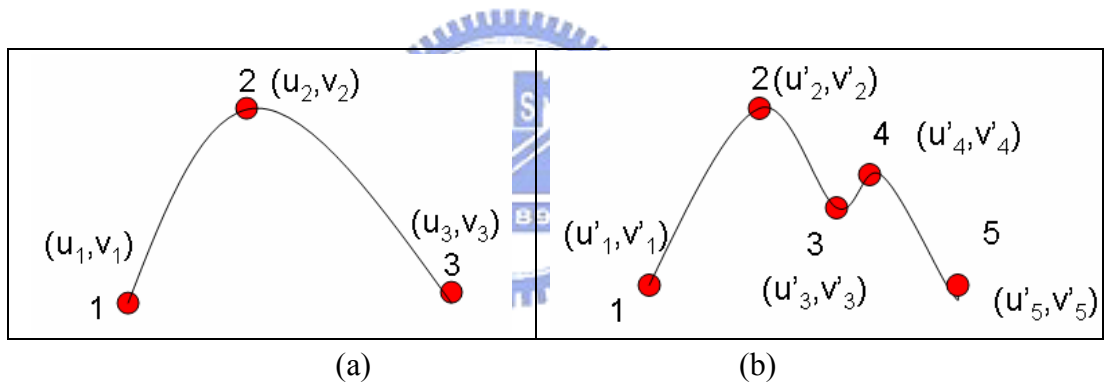
#### 5.4.2 Vector difference method

So far, we have seen how to extract control points. After that, we use vector difference method to determine the corresponding relationship between these control points. Because the relative distance is more important than absolute distance between an input face image and its mirrored image, the vectors from each control point to the standard point of each component are computed. After that, the vector length differences  $DiffSet_w$  of a pair of corresponding components  $corr_w$  can be defined as follows:

$$DiffSet_w = \{ \text{vector\_length}(cp_i, s_w) - \text{vector\_length}(cp_j, s'_w) \mid cp_i \in com_p, cp_j \in com'_q, \\ (com_p, com'_q) = corr_w, s_w \text{ is the standard point of } com_p, s'_w \text{ is the standard} \\ \text{point of } com'_p \}$$

$$diff_{i_j} = \text{vector\_length}(cp_i, s_w) - \text{vector\_length}(cp_j, s'_w), \text{ where } cp_i \in com_p, cp_j \in com'_q, \\ (com_p, com'_q) = corr_k, s_w \text{ is the standard point of } Com_p, s'_w \text{ is the standard} \\ \text{point of } com'_p$$

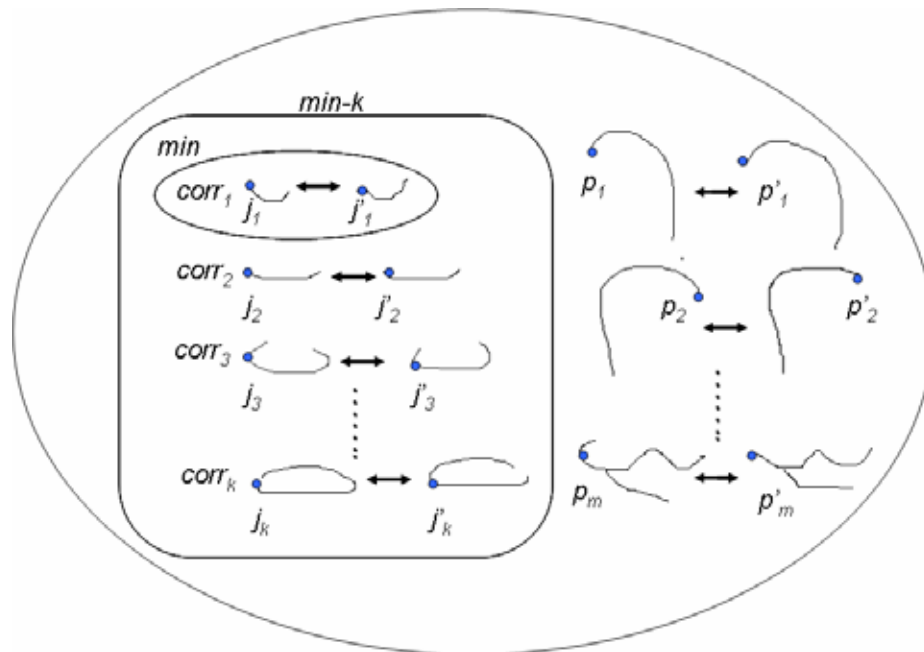
$DiffSet_w$  is a set of differences. If the minimum vector length difference in  $DiffSet_w$  is smaller than a predefined threshold, in other words, they are relatively close, the two control points  $cp_i$  and  $cp_j$  are considered as corresponding control points. Otherwise, they are not considered as corresponding control points. We introduce an example of the vector difference method shown in Fig. 5-6. The vector length difference  $diff_{2_3}$ ,  $diff_{2_4}$ ,  $diff_{2_5}$  are large and the vector length difference  $diff_{2_2}$   $diff_{3_5}$  are smaller. So that the control points 2 in Fig. 5-6 (a) with 2 in Fig. 5-6 (b) and the control points 3 in Fig. 5-6 (a) with 5 in Fig. 5-6 (b) are regarded as corresponding control points, but others are not. By the vector difference method, we can eliminate those control points without correspondence.



**Fig. 5-6** A vector difference method example (a) component 1, (b) component 2

By the vector difference method, we adopt the pair of left-most points of each pair of corresponding components as standard points. The correctness of standard point is very important, therefore, we design a feedback mechanism: voting, to check the correctness. In voting mechanism, we select the first  $k$  minimum corresponding components  $corr_1, corr_2, \dots, corr_k$  and the left-most points of each component in these  $k$  corresponding components to be the *judges*. Each *judge* consists of two points:  $(j_k, j'_k)$  shown in Fig. 5-7. For each corresponding component, we calculate the vector

difference between *judges*  $(j_1, j'_1), (j_2, j'_2), \dots, (j_k, j'_k)$  and left-most points  $(p_1, p'_1), (p_2, p'_2), \dots, (p_m, p'_m)$ . If the vector difference is smaller than the predefined threshold, the judge vote for this pair of corresponding components, otherwise, it votes against this pair of corresponding components. If over half of the  $k$  judges vote for a pair of corresponding components, we adopt the left-most points of the components of this pair of corresponding components as the standard points. Otherwise, we adopt the left-most points of the minimum pair of corresponding components as the standard points instead for this pair of corresponding components.



**Fig. 5-7** Corresponding component set

In the pose normalization module, we use morphing technique to accommodate the view angle problem. However, morphing requires corresponding control lines which are usually hand-marked. In this chapter, the automatic correspondence methodology extracts the corresponding features automatically and the methodology can help the morphing technique to be automatic, called automatic morphing. Here we show an automatic morphing result per subject in the ORL database in Fig. 5-8. The

corresponding features are produced by automatic correspondence methodology. The face images after pose normalization eliminate the slight displacement and rotation of the original images. But all of the morphing images are maintain original complexion, expression and contour. They can be correctly recognized.



**Fig. 5-8** Automatic morphing results

# Chapter 6

## Experimental Results

We demonstrate the performance of the edge-based face recognition system using a benchmark database: the ORL database [38]. The ORL database contains 40 distinct persons with ten images per person. The images were taken by varying the pose, lighting, and facial expressions (open/closed eyes, smiling/not smiling). The dimension of each image is  $92 \times 112$  pixels.

In order to emphasize the functionality of different modules, we examine different subsystems of APER shown in Table 6-1. The subsystem PCA means that each testing image is fed into the original PCA method without any processing. The subsystem PER is similar to APER, and the only difference between them is that PER gathers the correspondence of face and mirrored face manually. The subsystem APNR normalizes the pose of face images manually and performs the recognition by the PCA module without the processing of edge map module. Finally, the PNR is similar to APNR, but gathers the correspondence of face and mirrored face manually.

**Table 6.1** Subsystem table

PCA	Principal Component Analysis
PNR	Pose Normalization Recognizer
APNR	Automatic Pose Normalization Recognizer
PER	Pose normalization and Edge map Recognizer
APER	Automatic Pose normalization and Edge map Recognizer

Here we describe the experiments designed. First of all, we show the recognition results of the shift problem and compare the recognition results before and after using Level-mask in Section 6.1. Next, we compare the recognition rate of the PCA method and the improvement by each subsystem of APER. In this chapter, we analyze the recognition results under varying pose in Section 6.2 and the recognition results under varying pose and illumination in Section 6.3. The recognition rate is defined as the amount of correctly recognized testing images divided by the amount of all the testing images. In all of the experiments, we randomly choose the training images per subject, and calculate the average recognition rate by the original PCA method, the APER, the PER, the APNR, and the PNR improvement respectively for twenty times.

## 6.1 Recognition results of slight shift problem

When we feed the Sobel edge image directly into the PCA module for recognition, the Sobel edge image still cannot be recognized correctly due to the slight shift problem as mentioned in Section 4.3. Hence, we design a Level-Mask as illustrated in Section 4.4 to modify the Sobel edge image into Level-Masked Sobel (LMS) image. We show the recognition results of the Sobel image and LMS image with single training image in Table 6.2. In Table 6.2, the recognition rate of the Sobel image is 6.50% which is much lower than that of the LMS image. In other words, we should transform Sobel image into LMS image before recognition for better performance. .

**Table 6.2.** Recognition result of slight shift problem with single training image

	Number of correctly recognized images	Number of incorrectly recognized images	Recognition rate
Sobel image	22	338	6.50%
LMS image	279	81	77.50%

## 6.2 Recognition results under varying pose

This experiment demonstrates the face recognition results of the original PCA method, the APNR and the PNR under varying pose condition. Table 6.3 to Table 6.7 show the respective recognition rates of the subsystems using from one to five training images. For each testing image, the face recognition accuracy of the PCA method is 73.61% shown in Table 6.3. In the APNR(Automatic Pose Normalization recognizer) and PNR(Pose Normalization Recognizer) system, the testing images are transformed to approximated frontal face automatically and manually respectively, before the PCA recognition. The face recognition accuracy is improved to 79.72% and 84.17% with single training image. Hence, the improvement of PCA is 8.3% by APNR system and 14.35% by PNR system. In Table 6.4, the recognition rate of the PCA method rises to 80.63% because there are more training images and the recognition rate of the APNR and PNR also rises to 89.38% and 92.81%. Table 6.5, 6.6, and 6.7 show the three methods each with three, four and five training images, and the recognition rates of our proposed APNR and PNR method are always higher than those of the PCA method. Hence, the recognition results confirm that the pose normalization can accommodate the pose problem.

**Table 6.3.** Recognition result under varying pose with single training image

	Number of correctly recognized images	Number of incorrectly recognized images	Recognition rate
PCA	265	95	73.61%
APNR	287	73	79.72%
PNR	303	57	84.17%

**Table 6.4** Recognition result under varying pose with two training images

	Number of correctly recognized images	Number of incorrectly recognized images	Recognition rate
PCA	258	62	80.63%
APNR	286	34	89.38%
PNR	297	23	92.81%

**Table 6.5** Recognition result under varying pose with three training images

	Number of correctly recognized images	Number of incorrectly recognized images	Recognition rate
PCA	248	32	88.57%
APNR	263	17	93.93%
PNR	272	8	97.14%

**Table 6.6** Recognition result under varying pose with four training images

	Number of correctly recognized images	Number of incorrectly recognized images	Recognition rate
PCA	226	14	94.17%
APNR	231	9	96.25%
PNR	235	5	97.92%

**Table 6.7** Recognition result under varying pose with five training images

	Number of correctly recognized images	Number of incorrectly recognized images	Recognition rate
PCA	194	6	97.00%
APNR	195	5	97.50%
PNR	198	2	99.00%

Fig. 6-1 shows the recognition curves of the three methods compared in this experiment with one to five training images. Our APNR and PNR system significantly outperform the PCA method with few training images. Besides, we can find that the recognition rate of the PNR is always higher than the APNR, because human is more intelligent than computer. The correspondence is indeed more precise by human than by our automatic methodology. The improvement of the PNR system is the approximate upper bound of the APNR method theoretically, and it is the limitation in



the performance enhancement of our automatic correspondence module of APNR.

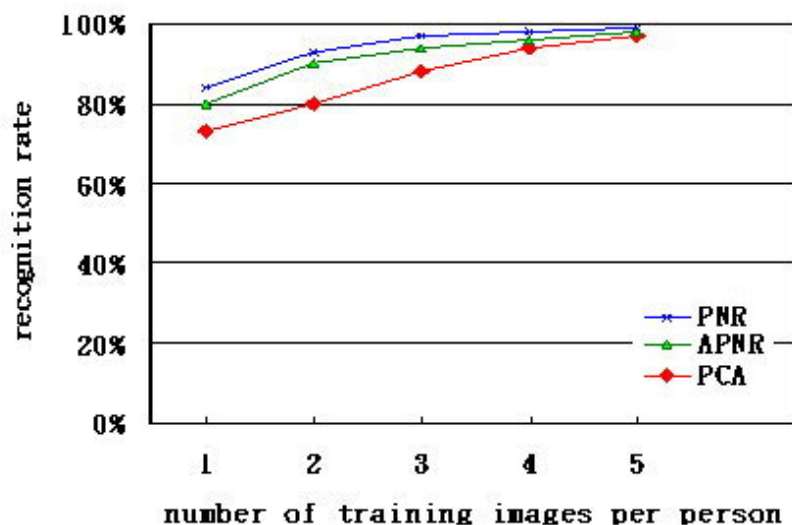


Fig. 6-1 Recognition curves based on PNR, APNR, and PCA

### 6.3 Recognition results under varying pose and lighting

This experiment shows the recognition results of the original PCA method, the PER, APER, and PNR subsystems under varying pose and lighting condition. Because the illumination variation of pictures in ORL database is not very obvious, we have done some modifications on luminance to emphasize the effect of illumination. The second row of Fig. 6-2 shows the modified face images of a person in the ORL database.

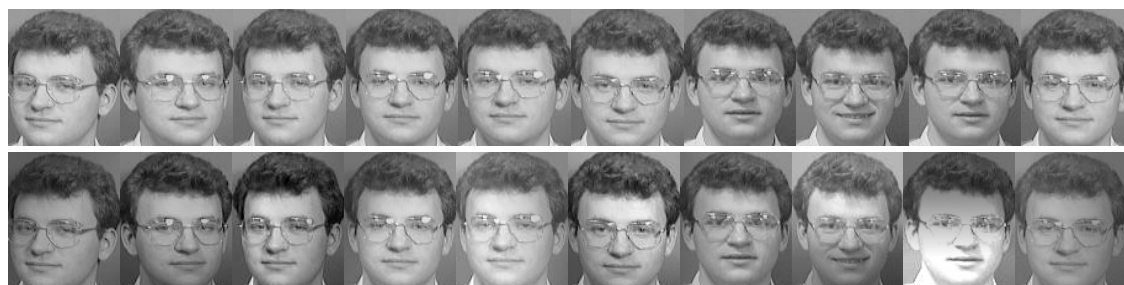


Fig. 6-2 The first row lists some images in ORL database; the second row lists the corresponding images modified on luminance of the first row.

Table 6.8 to Table 6.12 show the recognition rates of the original PCA method, the

APNR system, the APER(Automatic Pose normalization and Edge map Recognizer) and the PER(Pose normalization and Edge map Recognizer) with one to five training images. The recognition rates of our proposed APNR, APER and PER method are all higher than that of the PCA method. Hence the edge map and pose normalization are useful for the face recognition approaches under the pose and illumination variation. Fig. 6-3 shows the recognition curves of the four systems compared in this experiment with one to five training images. The APNR, APER and PER systems significantly outperform the PCA method with few training images.

**Table 6.8** Recognition result under varying pose and illumination with single training image

	Number of correctly recognized images	Number of incorrectly recognized images	Recognition rate
PCA	166	194	46.11%
APNR	171	189	47.50%
APER	251	109	69.72%
PER	276	84	76.67%

**Table 6.9** Recognition result under varying pose and illumination with two training images

	Number of correctly recognized images	Number of incorrectly recognized images	Recognition rate
PCA	169	151	52.81%
APNR	188	132	58.75%
APER	243	77	75.94%
PER	268	92	83.75%

**Table 6.10** Recognition result under varying pose and illumination with three training images

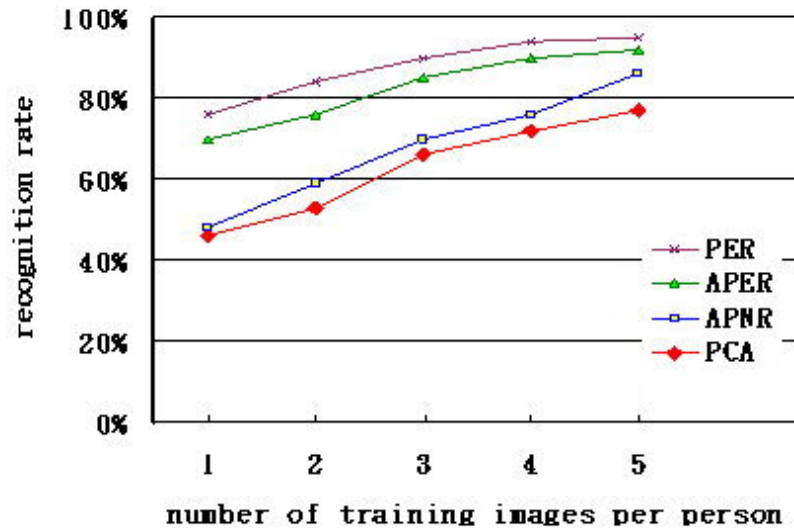
	Number of correctly recognized images	Number of incorrectly recognized images	Recognition rate
PCA	186	94	66.43%
APNR	197	83	70.36%
APER	239	41	85.36%
PER	252	28	90.00%

**Table 6.11** Recognition result under varying pose and illumination with four training images

	Number of correctly recognized images	Number of incorrectly recognized images	Recognition rate
PCA	173	67	72.08%
APNR	183	57	76.25%
APER	217	23	90.42%
PER	226	14	94.17%

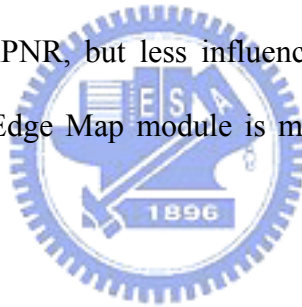
**Table 6.12** Recognition result under varying pose and illumination with five training images

	Number of correctly recognized images	Number of incorrectly recognized images	Recognition rate
PCA	154	46	77.00%
APNR	165	35	82.50%
APER	185	15	92.50%
PER	189	11	94.50%



**Fig. 6-3** Recognition curves based on PER, APNR, APER, and PCA.

It is obvious that, the illumination problem has a very great influence on accuracy rate of PCA approach and APNR, but less influence on APER and PER system. It shows that the System with Edge Map module is more resistant of illumination than without this module.



# Chapter 7

## Conclusions and Future Work

The face recognition accuracy degrades under varying illumination and pose. Many researches have discussed solutions to solve the two problems, but most of them need multiple training images. Unfortunately, this requirement cannot always be satisfied. In this thesis, we have proposed a system for face recognition called automatic pose normalization and edge map recognizer (APER). The APER employs pose normalization and edge map for face recognition with single training image. Pose normalization by morphing before PCA recognition is proposed to overcome the aspect angle problem effectively and an automatic corresponding algorithm has been derived to enhance the efficiency of the pose normalization process. Edge map with level-mask process used in face recognition is less sensitive in illumination variation condition and more is more efficient in computation. In conclusion, the APER can improve the performance of conventional PCA approach under varying pose and illumination with single training image. The comparison of the APER and the related works is shown in Table 7.1.

**Table 7.1** The comparison of related works and APER

Face recognition Approaches	accommodate pose problem	accommodate illumination problem	single training image
View-based eigenface	✓	✗	✗
Active appearance model	✓	✗	✗
Heuristic method	✗	✓	✓
Illumination cone	✓	✓	✗
<b>APER</b>	<b>✓</b>	<b>✓</b>	<b>✓</b>

To improve the performance and the robustness of the APER, some enhancements can be done in the future:

- (a) We are currently extending the system based on other recognition approaches. We believe that pose normalization and edge map are also useful for other face recognition approaches, not only for PCA.
- (b) Edge map can also be extended to derive the automatic correspondence algorithm for pose normalization, and the edge map can be morphed for face recognition directly. Morphing by the edge map will make the system more efficient and we will prove it in the future.
- (c) The automatic correspondence algorithm sometimes extracts error correspondence, and the error correspondence degrades the performance of face recognition. Therefore, the recognition rate of APER is lower than PER. We will improve our automatic correspondence algorithm as accurate as hand-marked.
- (d) In this thesis, we focus on intensity images but not color images. Pose normalization, edge map and automatic correspondence can directly be adopted on color images, but PCA recognition method have to be modified. Such modification for color images will be implemented in the future.



# Reference

- [1] W. Zhao, R. Chellappa, and A. Rosenfeld, P.J. Phillips, "Face Recognition: A Literature Survey," *ACM Computing Surveys*, 399-458, 2003
- [2] P. J. Phillips, R. M. McCabe, and R. Chellappa. "Biometric image processing and recognition." *Proc. of the European Signal Processing Conference*, 1998.
- [3] Matthew A. Turk and Alex P. Pentland, "Face Recognition Using Eigenfaces," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 586-591, 1991
- [4] Rajkiran Gottumukkal , and Vijayan K. Asari, "An improved face recognition technique based on modular PCA approach," *Pattern Recognition Letters*, Vol.25 No.4, 429-436,2004
- [5] Y. Adini, Y. Moses, and S. Ullman, "Face recognition: The problem of compensating for changes in illumination direction." *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol.19, 721-732, 1997
- [6] Y.li, S. Gong, and H. Liddell, "Support vector regression and classification based multi-view face detection and recognition," *Proc. of Automatic Face and Gesture Recognition Conference, Grenoble*, 300-305, 2000.
- [7] T. Riklin-Raviv and A. Shashua, "The quotient image: Class based rerendering and recognition with varying illuminations." *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 566-571, 1999
- [8] Georghiadis, A. S., Kriegman, D. J., and Belhumeur, P. N., "Illumination cones for recognition under variable lighting: Faces" , *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 52-58, 1998
- [9] Fukunaga, Keinosuke, "Introduction to Statistical Pattern Recognition," *Elsevier*, 1990
- [10] B. Manjunath, R. Chellappa, and C. Malsburg, "A feature based approach to face recognition," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 373-378, 1992
- [11] Cox, J. Ghosn, and P. Yianilos, "Feature-based face recognition using mixture-distance," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 209-216, 1996
- [12] F. Shamaria and S. Young, "HMM based architecture for face identification," *Image and Vision Computing*, Vol.12, 537-583, 1994

- [13] A. Nefian and M. Hayes, "Hidden Markov models for face recognition," *Proc. of International Conference on Acoustics, Speech and Signal Processing*, 2721-2724, 1998
- [14] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.19, 696-710,1997
- [15] D. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, 831-836, 1996
- [16] P. N. Belhumeur, J. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.19, 711-720, 1997
- [17] W. Zhao, R. Chellappa, and A. Krishnaswamy, "Discriminant analysis of principal components for face recognition," *Proc. of International Conference on Automatic Face and Gesture Recognition*, 336-341, 1998
- [18] P. Phillips, P. Rauss, and S. Der. Feret (face recognition technology) recognition algorithm development and test report. Technical Report ARL 995, U.S. Army Research Laboratory, 1996
- [19] P.J. Phillips, P. Grother, R. Micheals, D. Blackburn, E. Tabassi, and J. Bone. Face recognition vendor test 2002: Evaluation report. NISTIR 6965, National Institute of Standards and Technology <http://www.frvt.org>.
- [20] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 84-91,1994
- [21] T. Cootes, K. Walker, and C. Taylor, "View-based active appearance models," *Proc. of International Conference on Automatic Face and Gesture Recognition*, 227-238, 2000
- [22] K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.19, 775-779, 1997
- [23] A. Georghiades, and P. Belhumeur, "Few to many : Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.23, No. 6, 643~660, 2001
- [24] Acharya and Ray, "Image Processing: Principles and Applications," Wiley-Interscience, 2005
- [25] Thaddeus Beier and Shawn Neely, "Feature-Based Image Metamorphosis,"



- Pro. of Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. Vol. 26, 35-42, 1992 .
- [26] Alam, M.S.; Al-Samman, A.R.; “Ultrafast invariant face recognition.” *Proc. of IEEE. National Aerospace and Electronics Conference*, 486-493 ,2000
- [27] James Matthews. “An introduction to edge detection: The Sobel edge detector.” Available at <http://www.generation5.org/content/2002/im01.asp>, 2002.
- [28] Angus M. K. Siu and Rynson W. H. Lau, “Image Registration for Image-Based Rendering”, *IEEE Transactions on Image Processing*, Vol. 14, No. 2, 241-252, 2005
- [29] Winston Li and Henry Leung, “A Maximum Likelihood Approach for Image Registration Using Control Point And Intensity”, *IEEE Transactions on Image Processing*, Vol. 13, No. 8, 1115- 1127, 2004
- [30] Huang, J, and Wechsler, H.; “Visual routines for eye location using learning and evolution”, *IEEE Transactions on Evolutionary Computation*, Vol. 4, No. 1, 73 – 82, 2000
- [31] Paperno, E, and Semyonov, D; ”A new method for eye location tracking”, *IEEE Transactions on Biomedical Engineering*, Vol 50, No. 10, 1174 – 1179, 2003
- [32] Jung-Il Choi, Che-Woo La, Phill-Kyu Rhee, and Yong-Lai Bae; “Face and eye location algorithms for visual user interface”, *IEEE First Workshop on Multimedia Signal Processing*, 239 – 244, 1997
- [33] Wiskott, L, Fellous, J.-M, Kuiger, N, and Malsburg, C “Face recognition by elastic bunch graph matching”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, 775 – 779, 1997
- [34] Mokhtarian, F, and Suomela, R. ”Curvature scale space for robust image corner detection”, *Proc. of International Conference on Pattern Recognition*, Vol. 2, 1819-1821, 1998
- [35] P. Saeedi, D. Lowe, and P. Lawrence, “An efficient binary corner detector,” *Proc. of International Conference on Control, Automation, Robotics And Vision*, Vol. 1, 338- 343, 2002,
- [36] S. Alkaabi, and F. Deravi , “Iterative Corner Extraction and Matching for Mosaic Construction”, *Proc. of the Canadian Conference on Computer and Robot Vision*, 468- 475, 2005
- [37] Han Wang and Brady, M. “A practical solution to corner detection”, *Proc. of IEEE Conference on Image Processing*, Vol. 1, 919-923, 1994
- [38] ORL face database, AT&T Laboratories, Cambridge, U.K. [Online]. Available at <http://www.cam-orlco.uk/facedatabase.html>