國立交通大學

電機學院　電機與控制學程

碩 士 論 文

基於主動外觀模型及彈性束圖匹配之人臉辨識系統

Face Recognition System based on Active Appearance Model and Elastic Bunch Graph Matching

研 究 生：周榮宗

指導教授：林進燈　教授

中 華 民 國 一〇一 年 八 月

基於主動外觀模型及彈性束圖匹配之人臉辨識系統

Face Recognition System based on Active Appearance Model and Elastic Bunch Graph Matching

研　究　生：周榮宗　　　　Student：Jung-Tsung Chou

指導教授：林進燈　　　　Advisor：Chin-Teng Lin

國　立　交　通　大　學

電機學院　電機與控制學程

碩　士　論　文

A Thesis
Submitted to College of Electrical and Computer Engineering
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master of Science
in
Electrical and Control Engineering
August 2012

Hsinchu, Taiwan, Republic of China

中華民國一Ｏ一年八月

基於主動外觀模型及彈性束圖匹配之人臉辨識系統

學生：周榮宗　　　　　　　　　　　　　　　　指導教授：林進燈 教授

國立交通大學　電機學院　電機與控制學程 碩士班

摘要

早期的人臉辨識方法常利用簡單的幾何特徵來做辨識，例如顏色、形狀、位置等特徵，而現今的人臉辨識技術已成熟至使用數學的描述及比對方法。人臉辨識技術可用於人臉的偵測與識別，人臉辨識的方法主要可分成兩類：第一種是以特徵為基礎(feature-base)，第二種是以圖像為基礎(view-base)。在眾多的人臉辨識技術中，最廣泛應用的方法為主成分分析(Principal Components Analysis, PCA)的相關演算法，然而 PCA 基本上對姿勢及表情的變化非常地敏感，因此本論文提出一種創新的 feature- base 演算法，它不易受姿勢及表情的變化所影響，以此為基礎的人臉辨識系統使用主動外觀模型(Active Appearance Model, AAM)來偵測人臉的特徵點，並且採用彈性束圖匹配(Elastic Bunch Graph Matching, EBGM)來識別偵測到的人臉是否為系統成員，其中我們利用 5 組不同大小及 8 組不同方向共 40 張 Gabor

filter 來描述由 AAM 所得到的特徵點，並經由分析組內差異(within-class deviation)及組間差異(between-class deviation)的方法從58 個特徵點中挑選出最具代表性的 14 個特徵點來做人臉辨識，此外我們也以不同大小的AR 人臉資料庫來分析特徵點的選擇與資料庫大小的關係，最後並將實驗結果與文獻中使用 PCA 及 LDA 的實驗結果做比較以檢驗系統的性能。

Face Recognition System based on Active Appearance Model and Elastic Bunch Graph Matching

student：Jung-Tsung Chou Advisors：Dr. Chin-Teng Lin

Degree Program of Electrical and Computer Engineering
National Chiao Tung University

ABSTRACT

Early face recognition algorithms used simple geometric features, like feature color, shape, and position etc., but the recognition process has now matured into a mathematical representations and matching processes. Face recognition can be used for both verification and identification. There are two predominant approaches to the face recognition problem: geometric (feature-base) and photometric (view-base). Among many different algorithms, Principal Components Analysis (PCA) related algorithms are the most popular method in face recognition field. However, PCA basically is very sensitive to pose, expression change. In this thesis, we propose a novel feature-base algorithm which is more robust to pose and expression influence. The face recognition system uses Active Appearance Model (AAM) to detect feature points on the face and adopts Elastic Bunch Graph Matching (EBGM) to identify whether the face is a member or not. Firstly, we introduce 40 Gabor filters in 5 different sizes and 8 different directions to describe the feature points obtained from AAM. Besides, we also analyze within-class and between-class deviation of 58 feature points and find out the optimized 14 feature points for face recognition. Furthermore, we use different sizes of AR database to analyze the relationship between feature points selection and database size. Finally, our experiment result is compared with that of PCA and LDA to examine our system performance.

誌謝

又到了寫序文的時候，這個時候代表學業又告了一個段落，此刻的心情是既興奮又充滿感觸，距離第一個碩士畢業已經有十年了，從原本的力學領域到後來接觸的光學領域，歷經數年的職場磨練後深感所學的不足及對電機資訊領域的渴望，於是在經過一番努力準備之後，我如願進入交大電機資訊學院碩士在職專班就讀，並有幸進入林進燈教授所領導的超視覺實驗室，而我就這樣一腳踏入奧妙的影像處理領域，有別於職場上只著重時效而不求甚解，在這裡就像一個大家庭，學長姐不吝於給予協助，同學學弟妹們也勇於表達自己的意見，徜徉在這種開放的研究環境中使我枯竭已久的心靈重新感受到學習、研究的樂趣。

在這段期間裡，我完成了人生大事，擁有了自己的家庭及心愛的寶貝兒子，不管是在家庭還是在工作上也負擔起更大的責任，有時工作、學業忙碌起來常讓我無法分身，感覺自己像是蠟燭一樣地兩頭燒，時有想要放棄的念頭，但在關鍵時刻總能轉變心境咬緊牙根將問題逐一解決，最終得以完成此篇論文，本論文雖未能臻至完善，卻也是這一段研究歷程的菁華呈現，而這一段時間的煎熬歷練也使得畢業的果實更加甜美。

本論文研究得以完成，首先要感謝指導教授林進燈老師的盡心指

導並提供完善的研究環境，以及口試委員陳永平教授、張志勇教授、陶金旺教授給予的精闢見解，此外還要感謝台大應力所的恩師李世光教授的推薦及鼓勵使我能再次重拾書本學習新知。在論文研究期間，感謝琳達學姊、東霖學長、肇廷學長在論文上的指導，同學良成、庭伊、廷維在準備口試期間的相互砥礪，此外也要感謝洸本、宗濬在口試當天的鼎力相助，實驗室夥伴 Mukesh、柏義、峪鋒、威良、建維在平日的幫忙與建議。

最後要感謝內人佳佩在這段期間的支持及體諒，以及母親和岳父母大人幫忙照顧寶貝兒子昱升使我能無後顧之憂地完成學業。謹以此篇論文紀念父親大人，因為他的諄諄教誨使我在學問上秉持著尚可稱做”追求卓越”的精神，也希望以此精神做為寶貝兒子的表率。

Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1. Motivation

Biometrics is an emerging technology for identifying people by their physical and behavioral characteristics [1] [2], and plays an important role in recent years. The physical characteristics of an individual that can be used in biometric identification contain fingerprint, palmprint, iris, and face. Besides the behavioral characteristics including signature, speech, gesture, and gait are also suitable for biometric identification. Among all biometric identification methods, face recognition has attracted much attention in pass few decades. Since face recognition is belong to a passive and non-invasive system while verifying personal identity.

There are two predominant approaches to the face recognition problem: geometric (feature based) and photometric (view based). Early face recognition algorithms used simple geometric features, like feature color, shape, and position etc., but face recognition technology has now matured into a sophisticated mathematical representations and matching processes. Advancement in the past few decades has propelled face recognition technology into the spotlight.

Recently, face recognition are used for both security and biometrics applications. Many different algorithms were developed, in which Principal Components Analysis (PCA) related algorithms are the most popular method in face recognition field. However PCA basically is very sensitive to pose, expression change. Thus developing a reliable face recognition system that is immune from pose and expression influence is an important issue in present day.

1.2. Related works

Automated face recognition is a relatively new concept. Developed in the 1960s, the first semi-automated system for face recognition required the administrator to locate features (such as eyes, ears, nose, and mouth) on the photographs before it calculated distances and ratios related to a reference point, and then compared with the reference data. In the 1970s, Goldstein, Harmon, and Lesk [3] used 21 specific features such as hair color and lip thickness to recognize faces. The problem with both of these early solutions was that the measurements and locations were manually computed. In 1988, Kirby and Sirovich [4] applied principle component analysis, a standard linear algebra technique, to the face recognition problem. This was considered a milestone as it showed that just a small number of parameters were required so that a face image can be accurately described. In 1991, Turk and Pentland [5] used the eigenfaces techniques to achieve a fast automated face recognition system. However these approaches are typically constrained by pose and expression variations. Hence, some methods based on using a number of multi-view samples are proposed. For example, Du [6] used several different views of a face to deal with face variation problems. Beymer [7] proposed another multi-view method, which models a face with templates from 15 views, and sampling different poses on the viewing sphere.

1.3. System overview

This thesis proposes a novel feature base face recognition system that is more robust to pose and expression influence. Our face recognition system can be separated into two steps: the first one is database construction and the second is face recognition.

In the database construction step (as figure 1.1), images are input and facial orientation is executed to localize feature points of facial outline. Then all these feature points are coded as parameters of elastic bunch graph. Furthermore, all parameters are analyzed to design the best representative feature points for the database. Eventually, the database is constructed and will be used for face recognition step.

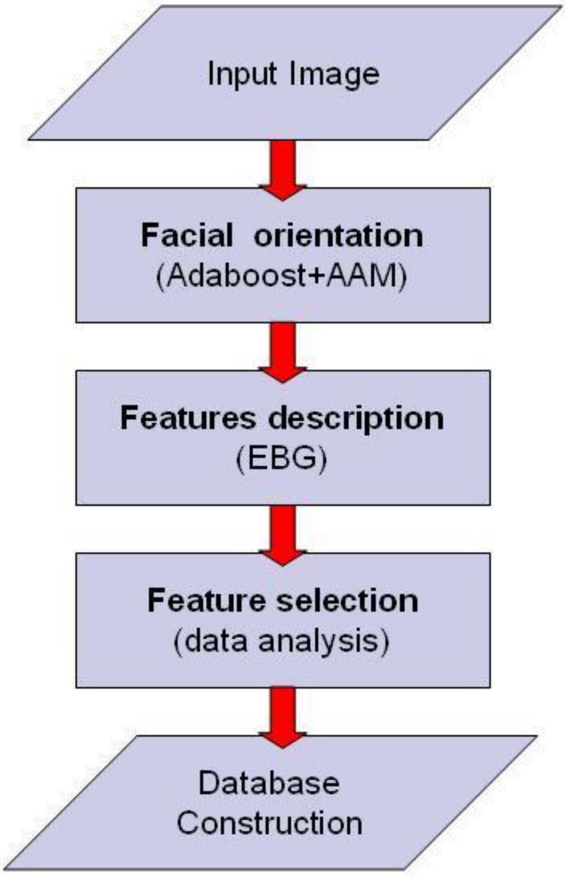

Figure 1.1: The flowchart of database construction step

The face recognition step is shown as figure 1.2. In the same way, the registered image is input and facial orientation is executed to localize feature points of facial outline. Then the face is described by parameters of elastic bunch graph according to the best representative feature points of the database. These parameters are compared with that in the database to find out the best similar one. Finally, the best solution is output and face recognition step is finished.

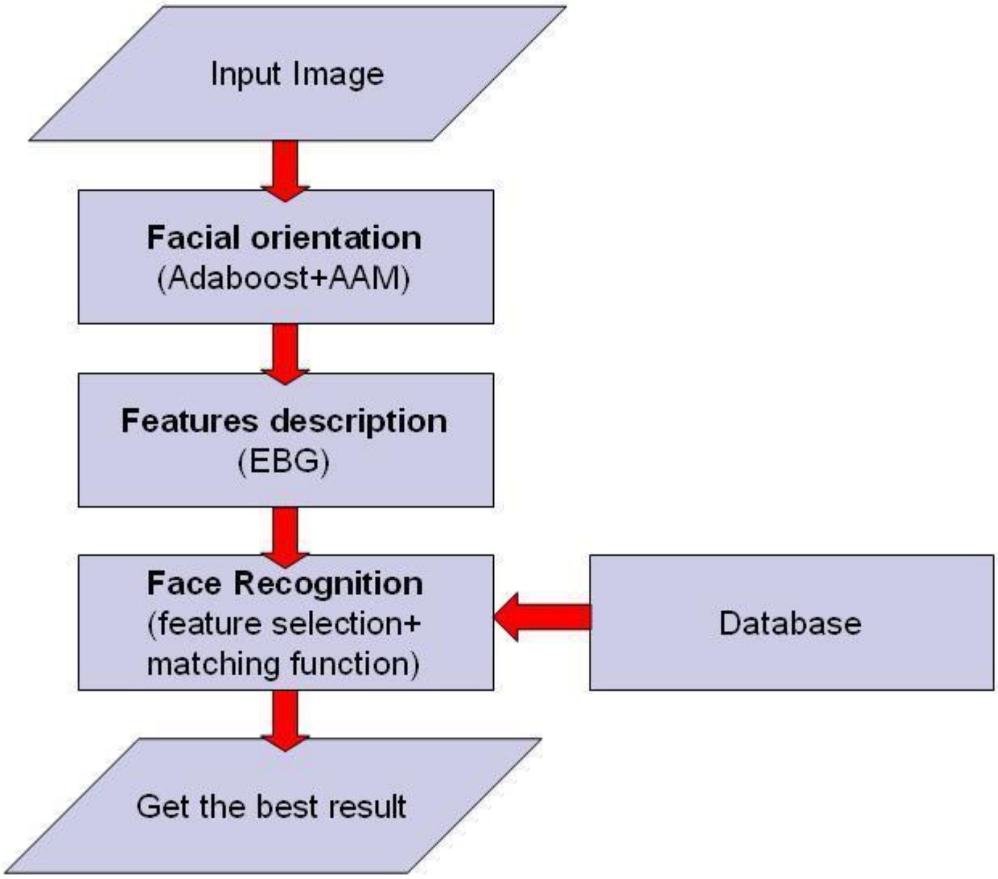

Figure 1.2: The flowchart of face recognition step

1.4. Thesis architecture

In this thesis, facial orientation technology including face detection and Adaptive Appearance Model will be introduced in chapter 2 firstly. In the chapter 3, methods of feature description containing Elastic Bunch Graph and

Gabor filter will be presented. In the chapter 4, we will illustrate some face recognition algorithms related to feature selection and matching process. Finally, we will show various experiment results and then make some conclusions.

Chapter 2: Feature Orientation

Above all, the first thing to do is face detection no matter what method the face recognition uses. However, only the face position in image does not provide sufficient information to extract significant features for the face recognition. So it is necessary to extract the representative features from facial orientation for face recognition. In this chapter, we will describe how to detect face and what method we use in facial orientation in this subsystem.

2.1. Face detection

The face detection technique employed in this system is based on AdaBoost algorithm that Viola and Jones proposed in 2004 [8]. The algorithm incorporates several innovative features: (1) It uses Haar-like input features: a threshold applied to sums and differences of rectangular image regions. (2) Its integral image technique enables rapid computation of the value of rectangular regions or such regions rotated 45 degree. This data structure is used to accelerate computation of the Haar-like input features. (3) It uses statistical boosting to create binary (face-not face) classification nodes characterized by high detection and weak rejection. (4) It organizes the weak classifier nodes of a rejection cascade. In other words, the first group of classifiers is selected, which best detects image regions containing an object while allowing many mistaken detections. The next classifier group is the second-best for detection with weak rejection, and so forth. In test mode, an object is detected only if it passes through the entire cascades.

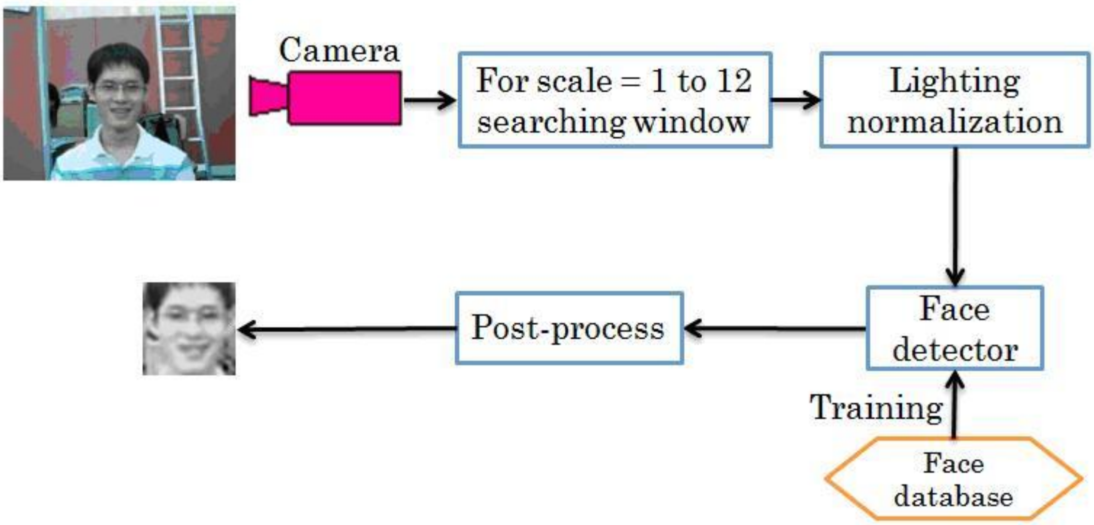

Figure 2.1: The process of face detection

Figure 2.1 shows the implementation of the face detection process. Firstly, searching windows with various sizes would be used to find out different face candidates in multi-scale from the images captured by a camera. The size of the searching window starts from resolution of 24 X 24 (pixels). Images are scanned by searching windows of 12 scales with a scaling factor of 1.25. All face candidates are normalized to the same size and luminance. Then, each normalized candidates will be classified as a face or not by the face detector which was trained beforehand. Finally, the face regions are located precisely from the image in the end of the subsystem.

2.2. Active Appearance Model

Facial orientation is an important technique for face recognition and expression recognition. The correctness of extractive features including shape and texture will greatly affect the recognition result. Many matching deformable models of shape and texture to real-time images have been proposed. Lanitis et al. [9] and Edwards et al. [10] used a boundary finding algorithm (Active Shape Model) to find out the best shape and used that for texture matching. Direct Appearance Model (DAM) algorithm [11], used texture information directly and predicted the new face position and appearance based on principal components of texture difference vectors. Song G, etc. [12] integrated Lucas-Kanade optical flow tracking algorithm, face alignment statistical model, and DAM, together in a Bayesian framework. Template Selection Tracker (TST) algorithm [13], used

Nelder-Meade simplex algorithm to drive the parameters of shape model and performed a nearest neighbor search between shape model and training set. Active appearance model (AAM) algorithm [14] used a combined statistical model of shape and texture, which adopted texture differences between the model and target image to predict translation parameters. And then an improved model was obtained from the best possible match.

Among algorithms mentioned above, AAM is a reliable and efficient method for features tracking and mapping. In this thesis, AAM will be adopted to localize feature points used for following recognition algorithm. AAM is mainly through building a combined statistical model of shape and texture and then executing an efficient iterative matching algorithm to obtain the best fitting result of the human face.

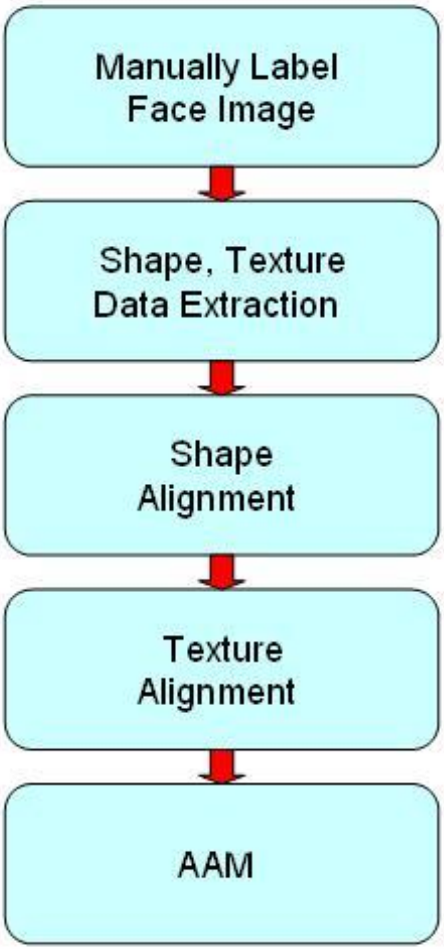

Figure 2.2: The flow chart of AAM modeling procedure

Figure 2.2 shows the flow chart of AAM modeling procedure. The statistical model is generated by using facial shape and texture information, in witch "texture" means the intensity across facial patch. To build a combined shape and texture model, training images with manually labeled features are needed and then shape alignment of location, scale and rotation is performed. After triangulation of manually labeled points, meshed face image is done as showed in figure 2.3.

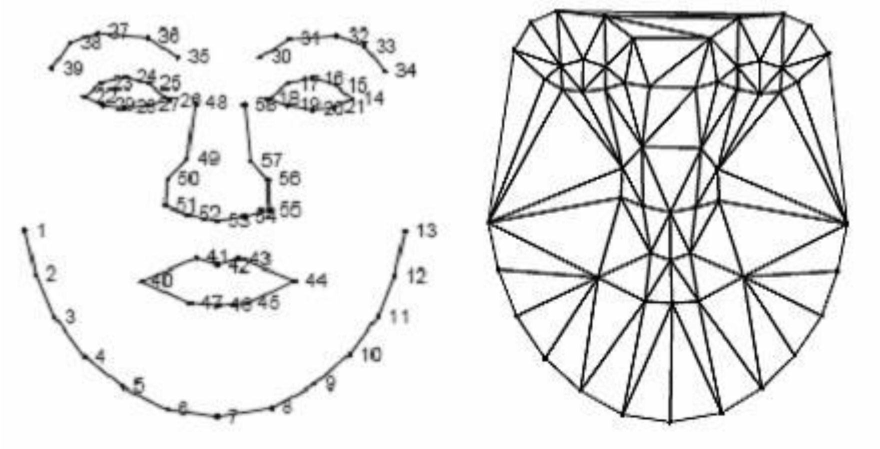

Figure 2.3: Feature points and meshed facial image

According to meshed face image, a table of texture model containing points' number and positions in each triangle is built up. The extracted shape and texture data are as figure 2.4.

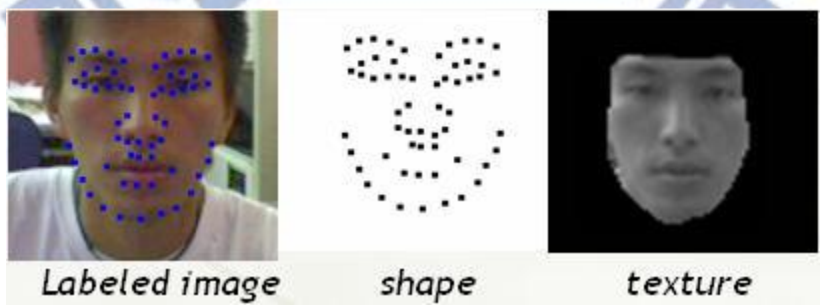

Figure 2.4: Extracted shape and texture data

Furthermore, Principal Components Analysis (PCA) is used to form an appearance model that has parameters $C = \{c_i\}$ controlling the shape and texture.

The shape and texture models can be described as equation (2.1).

$$
\begin{aligned}
x &= \bar{x} + Q_s C \\
g &= \bar{g} + Q_g C
\end{aligned}
\tag{2.1}
$$

Where $\bar{x}$ represents the mean shape, $\bar{g}$ stands for the mean texture, Q_s and Q_g are the matrices of selected eigenvectors obtained from PCA.

This thesis applied AAM to build a combined model of shape and texture and used a training set of 180 face images, and each was labeled with 58 points around the main profile. The shape model was generated with 116 parameters that indicate 58 points' coordinates. The texture model had 1944 parameters that represent gray values of sampled pixels. Thus the total number of used parameters is 2040 in this thesis.

Chapter 3: Feature Descriptor

We could localize facial points by combining face detection and AAM, but the position of these points are not enough to describe the characteristics of each individual face. We have to extract the texture information around each point for face recognition. Moreover, to choose a reasonable feature descriptor is very important. So we would describe how to define the feature vector of each feature point in this chapter.

3.1. Elastic Bunch Graph

Elastic Bunch Graph (EBG) was firstly proposed by Wiskott etc [15]. The main idea of EBG is to select specific locations on the face as feature points such as eyes, nose, and mouth. Furthermore, coefficients of Gabor wavelet transform can be extracted by using various Gabor kernels to describe each feature point. These coefficients are then compared with that stored in the database to identify whether the input face is a group member or not. Comparing with other global view base face recognition algorithms, EBG has characteristics including better recognition accuracy and invariance to pose and expression changes.

In EBG, a feature point on the face is called a jet. A jet is defined as a data set $\{J_j\}$ containing 40 parameters obtained from Gabor wavelet transform by using 40 different Gabor kernels. By combining several specific jets, an "elastic bunch graph" is constructed (as shown in figure 3.1) and then can be used to recognize faces between individuals. Each set of parameters J_j can be written as $J_j = a_j \exp(i\phi_j)$ with magnitude a_j and phase ϕ_j. Magnitude a_j varies slowly as position shifting, however phase ϕ_j is very sensitive to position variation. In this thesis, only magnitude a_j was used to recognize faces in the matching process.

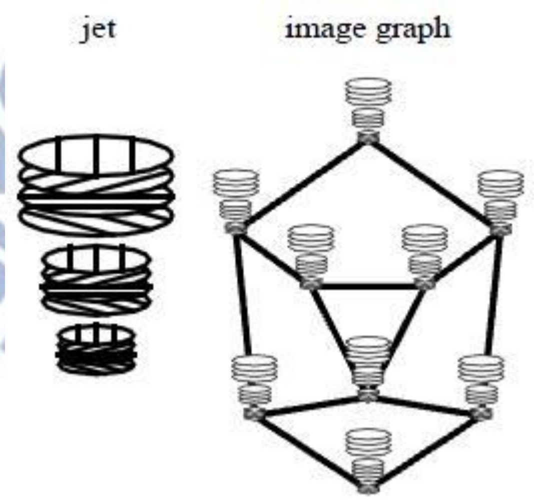

Figure 3.1: Elastic bunch graph

3.2. Gabor filter

In image processing, a Gabor filter is a linear filter used for edge detection. Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation and discrimination. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave. The Gabor filters are self-similar: all filters can be generated from one mother wavelet by dilation and rotation.

Gabor filter plays an important role in the EBG algorithm because the parameter used in EBG is exactly the convolution result of Gabor filter and pixels near the feature point. Thus selection of Gabor filters has a crucial effect on EBG algorithm.

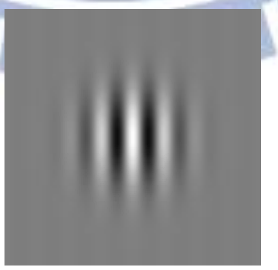

Figure 3.2: 2D Gabor filter

In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave as shown in the figure 3.2. Definitions of the Gabor filter in the 2D space are as follows:

Complex form:

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma)=\exp(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2})\exp(i(2\pi\frac{x'}{\lambda}+\psi)) \tag{3.1}$$

Real part:

$$g_{re}(x,y;\lambda,\theta,\psi,\sigma,\gamma)=\exp(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2})\cos(2\pi\frac{x'}{\lambda}+\psi) \tag{3.2}$$

Imaginary part:

$$g_{im}(x,y;\lambda,\theta,\psi,\sigma,\gamma)=\exp(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2})\sin(2\pi\frac{x'}{\lambda}+\psi) \tag{3.3}$$

Where:

$$\begin{aligned} x' &= x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta \end{aligned} \tag{3.4}$$

As equations above, θ represents orientation of the Gabor filter. The first part of equation (3.1) is the Gaussian envelope term, in which γ defines spatial aspect ratio and σ specifies radius of the Gaussian envelope. The last part of equation (3.1) is the sinusoidal term and can be divided into real part (equation (3.2)) and imaginary part (equation (3.3)), in which ψ and λ stand for phase offset and wavelength of the Gabor filter respectively.

In order to fully describing feature points on the face, several Gabor filters containing various orientations and wavelengths are used to extract coefficients of Gabor wavelet transform. In general, EBG uses eight orientations and five wavelengths to form 40 different Gabor filters as figure 3.3. The commonly used θ and λ are listed below:

$$\begin{aligned} &\theta \in \{0, \quad \pi/8, \quad 2\pi/8, \quad 3\pi/8, \quad 4\pi/8, \quad 5\pi/8, \quad 6\pi/8, \quad 7\pi/8\} \\ &\lambda \in \{4, \quad 4\sqrt{2}, \quad 8, \quad 8\sqrt{2}, \quad 16\} \end{aligned} \tag{3.5}$$

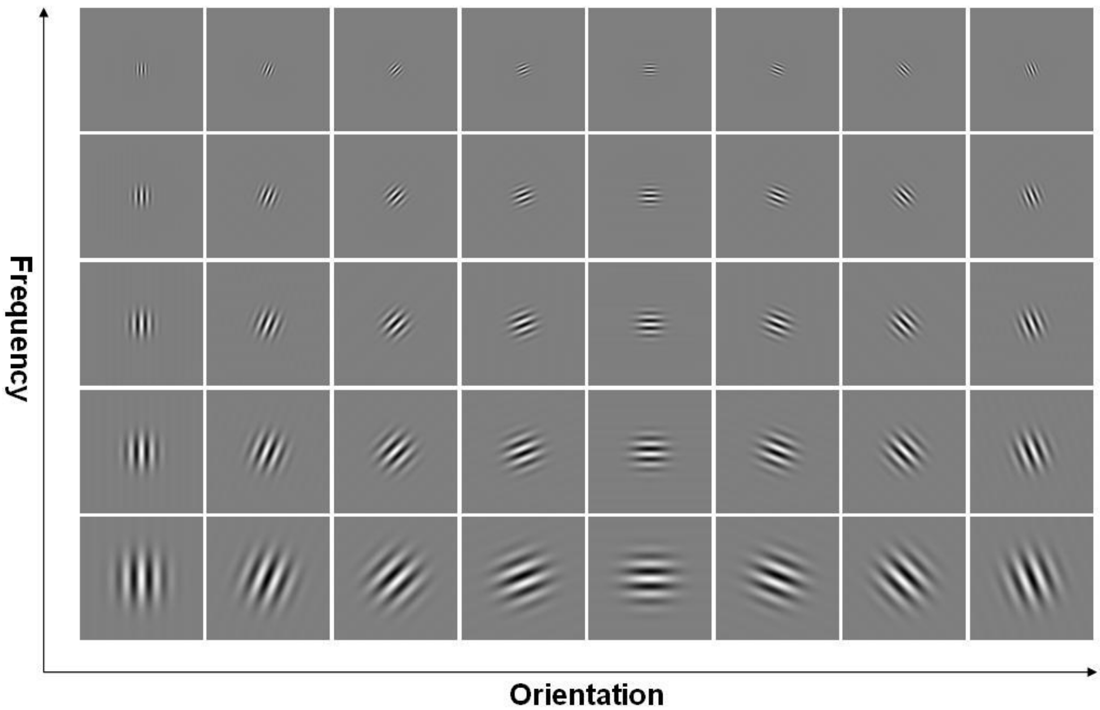

Figure 3.3: 40 different Gabor kernels

Chapter 4: Face Recognition

4.1. Feature Selection

The feature selection from points extracted by AAM is significant for face recognition. Some feature points have lager variation among different faces and some do not. These variations may be caused by illumination, pose, and expression changes. Not all feature points are well discriminable in recognition process. In fact, the discriminability of these points does not only depend on the variation among different faces but also the variation in each individual. For example, feature points with smaller variation are more suited to identification for single person. Meanwhile, feature points with lager variation are more beneficial to classification for different persons. As illustration of figure 4.1, to find out a set of feature points with smaller deviation within each class (each person) and lager deviation among different classes (different persons) is our goal. Besides, the selection of best feature points is not unique for every face database. Hence, how to select the optimized feature points automatically for each database is another important topic in this section.

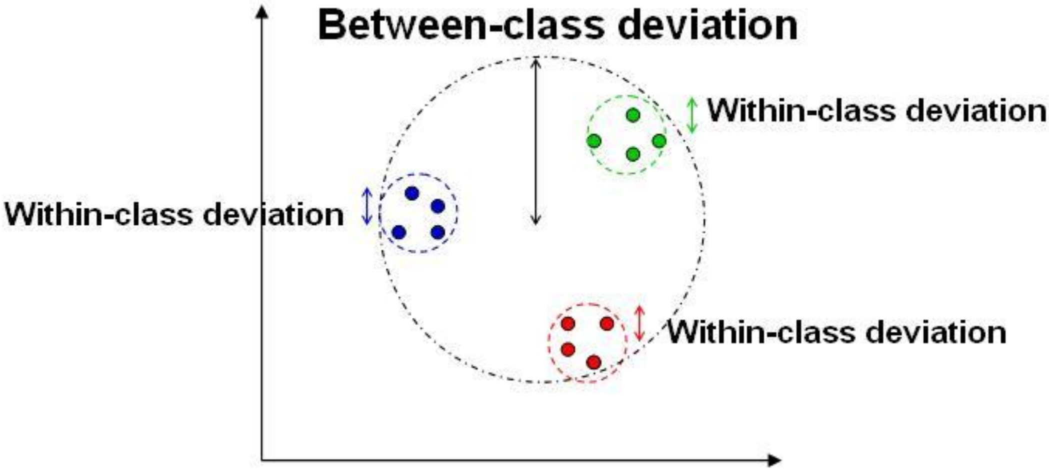

Figure 4.1: The illustration of within-class and between-class deviation

In our system, the optimized feature points will be selected according to the concept mentioned above for each face database. After AAM calculation, 58 feature points that describe facial outline can be obtained as figure 4.2. We just choose 14 feature points rather than all the 58 feature points in recognition process, because using all the 58 feature points will take too much time on convolution calculation.

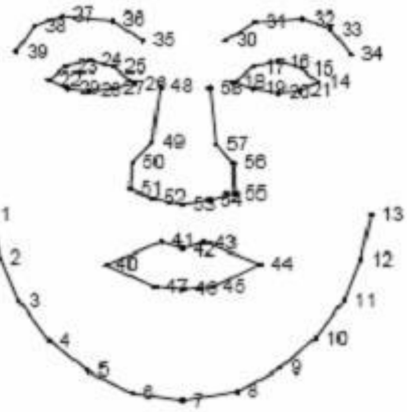

Figure 4.2: The 58 feature points obtained from AAM

First, we calculate 40 parameters of EBG for each of 58 feature points, that is, there are totally 2320 parameters for each face. The complete data structure for whole database is shown as figure 4.3, in which there are N classes in the database and M images in each class.

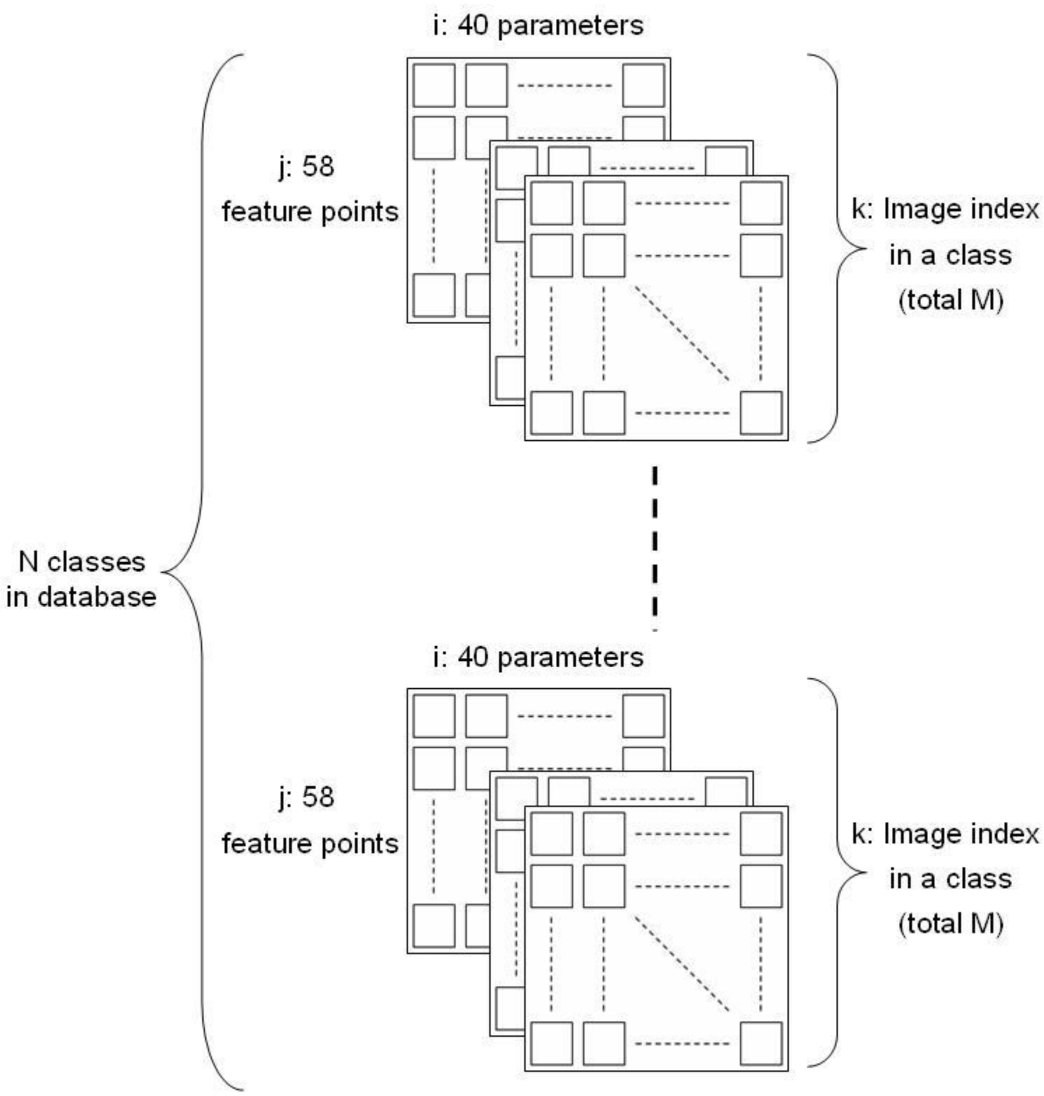

Figure 4.3: The complete data structure for the database

Afterward feature analysis can be separated into two steps. The first one is to compute standard deviation within each class and the second is to calculate standard deviation between classes. In the analysis of within-class deviation, deviation of each feature point is defined as the summation of standard deviations of 40 EBG parameters within a class. The deviation of each class is shown as equation (4.1). In equation (4.1), σ_j represents deviation of each feature point indexed by j. And μ_{ji} is the average of EBG matrix, in which the symbol i stands for parameter's index and symbol j represents index of feature point.

$$\sigma_j = \sqrt{\frac{1}{40}\sum_{i=1}^{40}(J_{ji} - \mu_{ji})^2}$$
$$\mu_{ji} = \frac{1}{M}\sum_{k=1}^{M} J_{kji} \tag{4.1}$$

After acquiring a vector of 58 deviation values, deviation vectors of all classes have to be averaged to obtain a more representative within-class deviation for the whole database.

For between-class deviation as equation (4.2), the average EBG parameters of each class and the whole database have to be determined firstly ($\bar{J}_{ji}$ and μ_{ji}, respectively). Then these average EBG parameters are used to calculate standard deviations between classes. And σ_j is the so called between-class deviation for each feature point indexed by j.

$$\sigma_j = \sqrt{\frac{1}{40}\sum_{i=1}^{40}(\bar{J}_{ji} - \mu_{ji})^2}$$
$$\bar{J}_{ji} = \frac{1}{M}\sum_{k=1}^{M} J_{kji} \tag{4.2}$$
$$\mu_{ji} = \frac{1}{M \times N}\sum_{l=1}^{M \times N} J_{lji}$$

To get the feature points having smaller within-class deviation as well as larger between-class deviation, we introduced two methods to combine the results of within-class deviation and between-class deviation. The first one is to sort feature points by deviation values and give them a score from 0 to 57. For within-class deviation, smaller deviation value gets higher score. For between-class deviation, lager deviation value gets higher score. Finally, the scores of within-class deviation and between-class deviation are summed so that we can choose first 14 feature points with higher total scores used for recognition process.

However, giving score by the order of deviation values without considering the difference between deviation values will cause a little distortion. Therefore, in the second method, between-class deviation and within-class deviation are divided by each other directly. The between-class deviation is put at the denominator because it is direct proportion to resolving power. And the within-class deviation is put at the numerator because it is an inverse proportion related to resolving power. Eventually, we refer the order of these ratios to select the best 14 feature points as representatives for face recognition.

In the next chapter, feature points from both methods will be used to examine system performance.

4.2. Matching process

In the matching process, there are several methods used to discriminate different individuals. The simplest method is the sum of absolute errors (SAE), which sums up each absolute difference between a_j and a'_j as shown in equation (4.3). Finally, the one with minimum SAE is chose for the best solution.

$$SAE = \sum_j \left| a_j - a'_j \right| \tag{4.3}$$

To examine recognition performance, there are 180 pictures in five different classes (individuals) as a testing set. After calculating with equation (4.3), the testing result is 72.2% matching accuracy.

However, if the error function only takes the sum of each parameter difference without considering the effect of kernel size, the parameter obtained from convolution with lager kernel size will have higher weight related to that with smaller kernel size. Therefore equation (4.3) can be modified and divided by kernel area $W_j^{\ 2}$ to normalize the affection of each EBG parameter, and the illustration is shown as figure 4.4. The modified error function is shown as equation (4.4), and we call it "normalized error function". By using normalized error function as equation (4.4), an improved testing result of 80.6% matching accuracy can be gained.

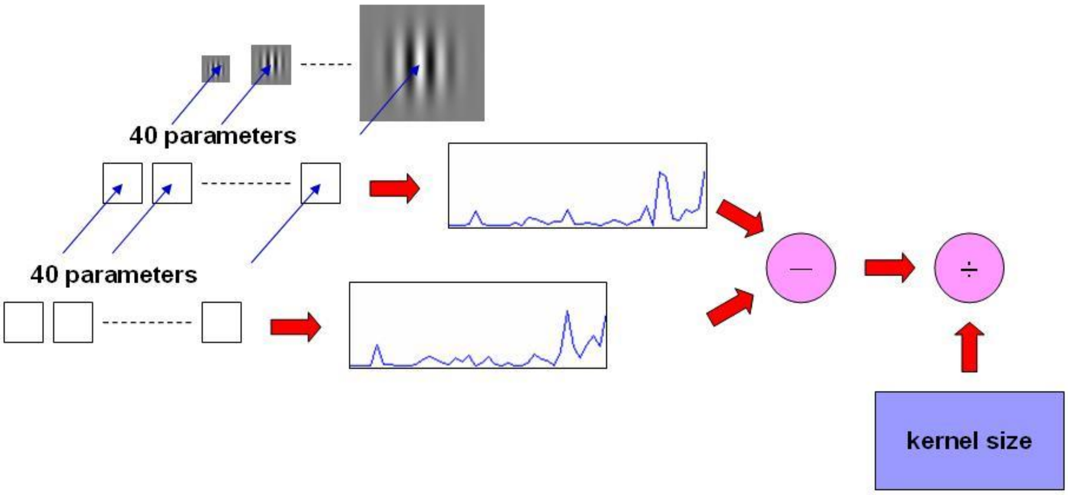

Figure 4.4: The illustration of normalized error function

$$Norm.Error = \sum_j \frac{\left|a_j - a'_j\right|}{W_j^{\,2}} \tag{4.4}$$

Another error function that we proposed as equation (4.5) can also eliminate the influence caused by kernel size difference. In the equation (4.5), magnitudes a_j and a'_j are directly divided each other. Due to a_j and a'_j are from the same kernel size for each index j, the affection of kernel size can be cancelled by division operation. In addition, log and modulus operations are performed to transfer the ratio of a_j and a'_j to absolute logarithm relation for physical rationality. As illustration of figure 4.5, two ratios of 0.01 and 100 have physically equivalent similarity except arrangement of a_j and a'_j. By taking log and modulus operations, two error functions are the same and equal to 2. According to definition of equation (4.5), a more improved result of 85% matching accuracy can be achieved.

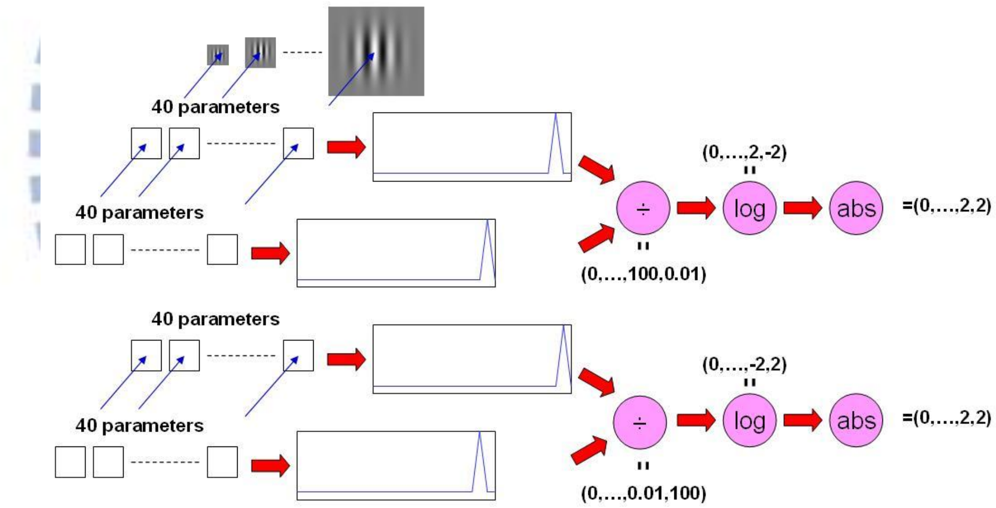

Figure 4.5: The illustration of log error function

$$Log.Error = \sum_j \left| \log(\frac{a_j}{a'_j}) \right| \quad (4.5)$$

Besides, a similarity function used by Buhmann and Lades [16] as equation

(4.6) is also examined here. In equation (4.6), numerator represents dot product of vectors a_j and a'_j as well as denominator stands for multiplication between norms of vectors a_j and a'_j. As illustration of figure 4.6, if a_j and a'_j are perpendicular to each other, the similarity function will be equal to one. On the contrary, if a_j and a'_j are orthogonal to each other, the similarity function will be zero. Differing from error function searching for minimum value, similarity function seeks out the maximum as the best solution in matching process. The matching accuracy computed with equation (4.6) is 86.7%, which is slightly higher than that with equation (4.5).

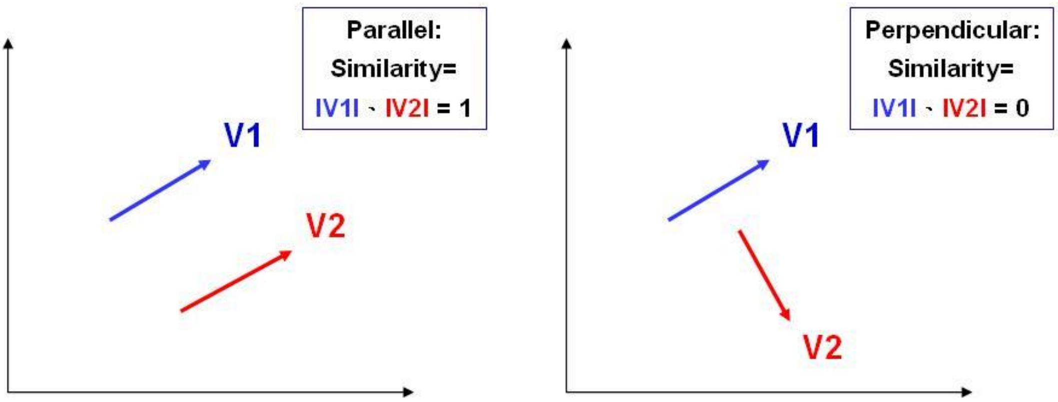

Figure 4.6: The illustration of similarity function

$$S_a(J,J') = \frac{\sum_j a_j a'_j}{\sqrt{\sum_j {a_j}^2 \sum_j {a'_j}^2}} \tag{4.6}$$

Chapter 5: Experiment Results

5.1. Database

For examining the performance of our system, we use two databases in the following experiments. One is the self-made database made by our laboratory, which contains 180 face images divided into 5 classes and each class has 36 images. This database is mainly used to examine system's resistance to expression change. For this reason, the database contains fewer classes but more images with extremely different expressions in each class. Figure 5.1 shows some example images of one person that represents many different expressions.

Figure 5.1: Examples of self-made database

The other is AR face database that was created by Aleix Martinez and Robert Benavente [17]. AR database contains frontal view faces with different facial expressions, illumination conditions, and occlusions (sun glasses and scarf). These pictures were taken under strictly controlled conditions but no restrictions on wear, make-up, hair style, etc. imposed to participants. In our experiments, only images without occlusions were used. Because EBGM belongs to feature-based algorithm and occlusions like scarf, glasses, will totally cover local feature information. Differing form the self-made database, here AR database is primarily used to examine system's capability for the database with more classes but fewer images in a class. We used 99 individuals of AR database in the experiments. In the first step, only 4 images without illumination variation in each class were used. (as images on the first row of figure 5.2) In the second step, we used 7 images in each class, including images with illumination variation. (as images on the second row of figure 5.2) And the experiment result was compared with others which were made by PCA and LDA.

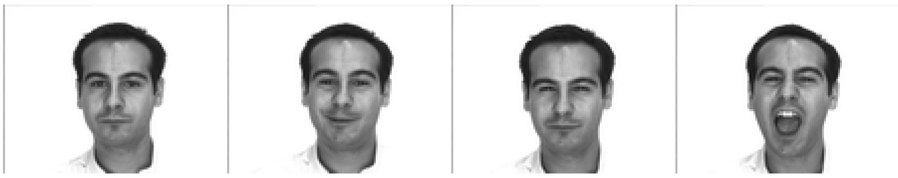

Figure 5.2: Examples of AR database

5.2. Feature selection versus matching method

As mentioned in last chapter, some feature points on the face have lager variation and some have smaller variation. These variations may be caused by illumination, pose, and expression changes. Therefore feature points with smaller variation are more suited to identification for single person. On the other hand, feature points with lager variation are more beneficial to classification for different persons. To get the feature points having smaller within-class deviation as well as larger between-class deviation, we proposed two methods to combine the results of within-class deviation and between-class deviation.

The first one is to sort feature points by deviation values and give them a score from 0 to 57. Then the scores of within-class deviation and between-class deviation are summed to get optimized 14 feature points with higher total scores. In the second optimization method, between-class deviation and within-class deviation are divided by each other directly. The between-class deviation is put at the denominator and the within-class deviation is put at the numerator.

Eventually, we refer the order of these ratios to select optimized 14 feature points.

Above all, we selected feature points through four different ways including within-class deviation, between-class deviation, combined deviation by score, and combined deviation by division, respectively. As well as, we adopted three different matching methods containing similarity function, normalized error function, and log error function, to cross-refer the experiment results.

Firstly, the self-made database was used. Figure 5.3 shows different sets of selective points from four manners mentioned above. Table 5.1 is the experiment results that used various conditions in the self-made database. The results indicate that the feature points from division method have better recognition ability and the recognition accuracy can achieve 90%. Besides, feature points from within-class deviation are superior to that from between-class deviation. The reason may be that the self-made database contains fewer classes but more images in a class so that optimization within classes is more important than optimization between classes.

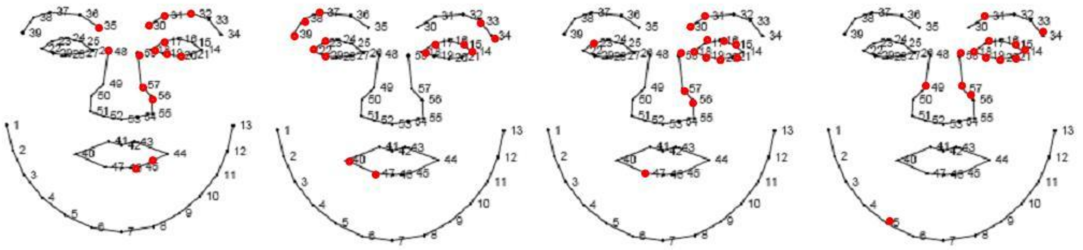

Figure 5.3: Selective points from four manners in self-made database

Table 5.1: Experiment results of various conditions in self-made database

Accuracy / Points / Method	Points from wi-std	Points from bt-std	Points from score method	Points from division method
Similarity function	77.8%	75%	72.2%	78.3%
Normalized error function	86.1%	72.2%	84.4%	88.3%
Log error function	90%	83.9%	85%	90%

Secondly, we performed experiments with the same conditions in AR database, in which there are 99 classes and each class includes 4 images. The sets of selective points from four different ways are shown in the Figure 5.4. And the experiment results that utilized AR database are listed in table 5.2. Similarly, the feature points from division method have better recognition capability and the recognition accuracy can be over 80%. In addition, feature points from between-class deviation are superior to that from within-class deviation. That may be because AR database contains more classes but fewer images in a class so that optimization between classes is more important than optimization within classes.

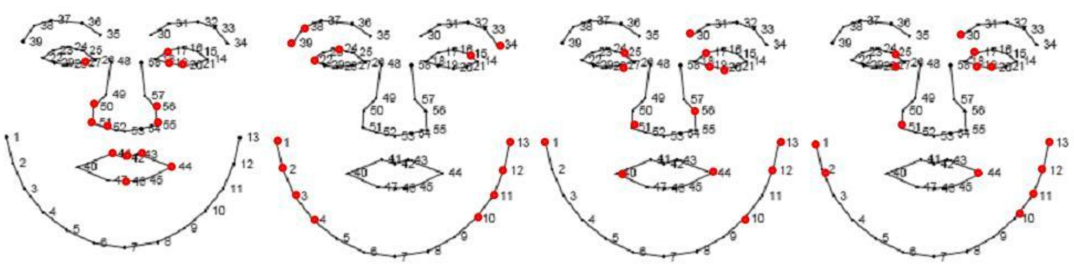

Figure 5.4: Selective points from four manners in AR database

Table 5.2: Experiment results of various conditions in AR database

Accuracy / Points / Method	Points from wi-std	Points from bt-std	Points from score method	Points from division method
Similarity function	74.4%	80.6%	78.5%	79.5%
Normalized error function	71.7%	80.3%	79.3%	80.1%
Log error function	69.4%	79.8%	77%	80.1%

5.3. Optimization for different database size

For analyzing optimization effect on different database size, different amount of classes in AR database was chose to perform experiments. The numbers of classes including 5, 10, 20, 30, 40, 50, 60, 70, 80, 99, was selected and each class contains 4 images. According to the second optimization method (combined deviation by division), 14 optimized feature points were obtained for each database size and then used these feature points in the face recognition process. As figure 5.5, the experiment results for different matching methods show that the recognition accuracy decreases as database size increasing, when database size is smaller than 30 classes. As well as the accuracy is convergent to approximate 80% when the size of database is grater than 40.

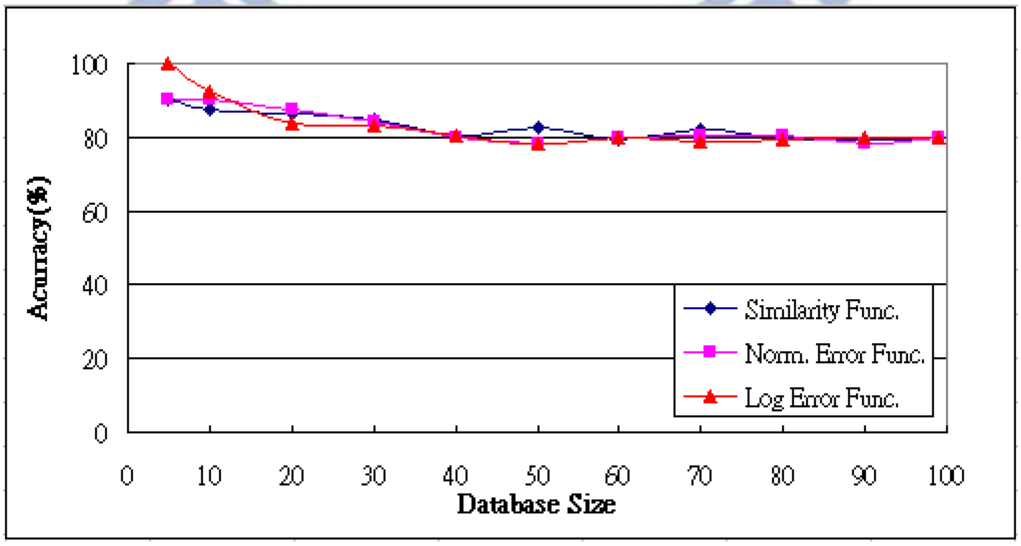

Figure 5.5: Experiment result of the first optimization method

5.4. FRR and FAR

To examine system's performance, two important properties have to be checked, which are FRR (Fault Rejection Rate) and FAR (Fault Acceptance Rate). FRR represents the rate of rejected persons who should be in the database. On the contrary, FAR means the rate of acceptance persons who should not be part of the database. A threshold value defines the acceptance boundary of error function, which controls both values of FRR and FAR. Among which, FRR is decreasing from 100% to 0% as threshold increasing. Oppositely, FAR is rising from 0% to 100% as threshold increasing. Here, AR database and normalized error function are used in the following tests. In the test of FRR, first 40 classes of AR database are used. In each class, three images are as target images and one image is selected for the test image. In the test of FAR, three images in each of first 40 classes are as target images and one image of following 40 classes (41~80) is chosen for the test image. In addition, an important property that describes the system's performance is EER (Equal Error Rate). Figure 5.6 shows the experiment results of FRR and FAR, in which the value of EER is approximate 20% when the threshold is about 50.

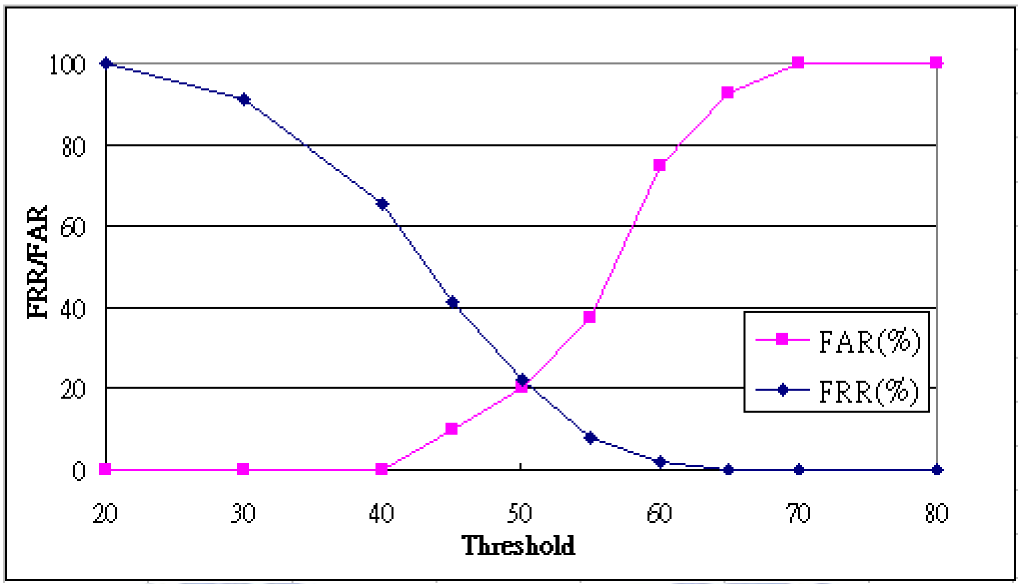

Figure 5.6: Experiment result of FRR and FAR

5.5. Comparison with PCA and LDA

Algorithm of EBGM belongs to feature-base face recognition method. Comparing with view-base face recognition methods such as PCA and LDA, EBGM has characteristic that is more immune from pose and expression variation. For comparing EBGM with PCA and LDA, we performed a comparison with the experiment result made by Martinez [18]. Here, both experiments used 50 individuals of AR database, in which there were 7 images for each person including images with illumination variation. Due to the difference of basic principle between feature-base and view-base algorithms, the controlling variable of EBGM was the number of feature points, and that of PCA and LDA was the number of eigenvectors.

In our experiment, the numbers of feature points were sampled from 7 to 56 with the interval of 7. Furthermore, similarity function, normalized error function, and log error function were used as well. The experiment result is shown as figure 5.7. In the reference experiment, the numbers of eigenvectors were sampled from 20 to 80 with the interval of 10. Besides, three kinds of methods were used including PCA, PCA without first 3 eigenvectors, and LDA, respectively. The experiment result is shown as figure 5.8.

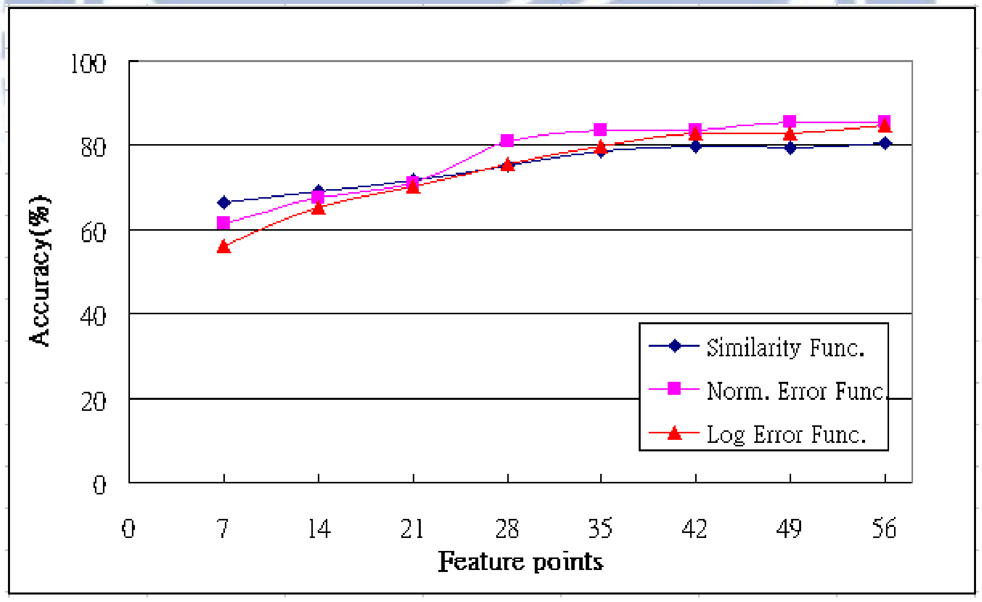

Figure 5.7: The experiment result of EBGM

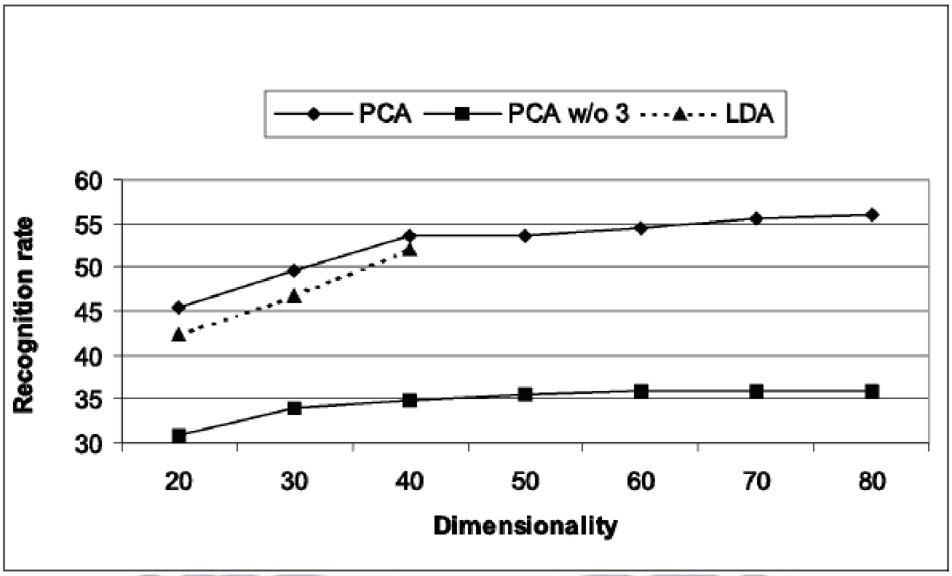

Figure 5.8: The experiment result of PCA and LDA

As figure 5.7 and figure 5.8, the recognition accuracy is rising as the number of used vectors (jet vector of EGB, eigenvector of PCA and LDA). In the experiment of EBGM, the accuracies of all methods are over 75% while the number of feature points goes beyond 21, and the average accuracy can surpass 80% when the among of feature points is lager than 35. In the reference experiment, all accuracies are basically below to 60%, in which PCA has better performance than LDA. Finally, comparing figure 5.7 and figure 5.8, the order of performance from better to worse is EBGM, PCA, and LDA, respectively. Owning to inherent difference of basic principle, the experiment is not a precise comparison between EBGM, PCA, and LDA but the results still present a rough performance trend of them.

Chapter 6: Conclusions

In this thesis, we proposed a feature-base face recognition system which is more robust to pose and expression influence. This face recognition system used Active Appearance Model to detect feature points on the face and adopted Elastic Bunch Graph Matching to recognize individuals in the database. First, we introduced 40 Gabor filters in 5 different sizes and 8 different directions to describe feature points obtained from AAM. Besides, we also analyzed within-class and between-class deviation of 58 feature points and found out the optimized 14 feature points for face recognition. Furthermore, we used different sizes of AR database to show that the recognition accuracy can be over 90% in the smaller database, and convergent to 80% in the larger database. Finally, our experiment result was compared with that of PCA and LDA, and the comparison showed that our system has a better performance than PCA and LDA.

Reference

[1] A.k. Jain, R. Bolle, and S. Pankanti, "Biometrics: Personal Identification in Networked Society," eds. Kluwer Academic, 1999.

[2] D. Zhang, "Automated Biometrics: Technologies Systems," eds. Kluwer Academic, 2000.

[3] A.J. Goldstein, L.D. Harmon, and A.B. Lesk, "Identification of Human Faces," Proc. IEEE, May 1971, Vol. 59, No. 5, 748-760.

[4] L. Sirovich and M. Kirby, "A Low-Dimensional Procedure for the Characterization of Human Faces," J. Optical Soc. Am. A, 1987, Vol. 4, No.3, 519-524.

[5] M.A. Turk and A.P. Pentland, "Face Recognition Using Eigenfaces," Proc. IEEE, 1991, 586-591.

[6] S. Du, and R.K. Ward, "Face recognition under pose variations," J. Franklin Inst. 343 (2006) 596-613.

[7] D.J. Beymer, "Face recognition under varying pose," Technical Report 1461, MIT AI Lab, Massachusetts Institute of Technology, Cambridge, MA, 1993.

[8] P. Viola and M.J. Jones, "Robust Real-Time Face Detection," International Journal

of Computer Vision 57(2), 137-154, 2004.

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

[10] G. Edwards, A. Lanitis, C. Taylor, and T. Cootes, “Statistical Models of Face Images－Improving Specificity,” Image and Vision Computing, vol. 16, pp. 203-211, 1998.

[11] X.W. Hou, S.Z. Li, H.J. Zhang, Q.S. Cheng, “Direct appearance models,” Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 828~833, 2001.

[12] G. Song, H.Z. Ai, G.Y. Xu, “Texture Constrained Facial Feature Point Tracking,” Conf. of Journal of Software, vol. 15, no. 11, 2004.

[13] D. Cristinacce and T. Cootes, “Facial Feature Detection and Tracking with Automatic Template Selection,” Proc. of 7th IEEE Conf. on Automatic Face and Gesture Recognition, pp. 429-434, April, 2006.

[14] T.F. Cootes, G.J. Edwards, and C.J. Taylor, “Active Appearance Models,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, June 2001.

[15] Laurenz Wiskott, J.M. Fellous, N. Kruger, and C. Malsburg, “Face recog.ni tion by elastic graph matching,’’ IEEE Transactions on Pattern Analysis and Machine

Inrelligence, vol. 19, pp. 775-779, 1997.

[16] M. Lades, J.C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R.P. Würtz, and W. Konen, “Distortion Invariant Object Recognition in the Dynamic Link Architecture,” IEEE Trans. Computers, vol. 42, no. 3, pp. 300–311, 1993.

[17] http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html.

[18] Aleix M. Martinez and Avinash C. Kak, “PCA versus LDA,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 2, pp. 228-233, Feb. 2001.