國 立 交 通 大 學

電 機 與 控 制 工 程 研 究 所

碩 士 論 文

利用影像輪廓與臉形於人體方位偵測

Subject Direction Detection Using the Silhouette and Face

研 究 生：李 宗 翰

指 導 教 授：張 志 永

中 華 民 國 九 十 七 年 六 月

利用影像輪廓與臉形於人體方位偵測

Subject Direction Detection Using the Silhouette and Face

學　　生：李宗翰　　　　Student : Tsung-Han Li

指導教授：張志永　　　　Advisor : Jyh-Yeong Chang

國立交通大學

電機與控制工程學系

碩士論文

A Thesis

Submitted to Department of Electrical and Control Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master in

Electrical and Control Engineering

June 2008

Hsinchu, Taiwan, Republic of China

中 華 民 國 九 十 七 年 六 月

利用影像輪廓與臉形於人體方位偵測

學生: 李宗翰　　　　　指導教授: 張志永博士

國立交通大學電機與控制工程研究所

摘要

　　由於近年來智慧型機器人的發展迅速，機器人與人之間的互動也愈來愈頻繁，當機器人與人溝通時，必須知道人的位置、距離及面對的方向等資訊。在此篇論文中，我們結合二維影像輪廓比對與臉部方位偵測來完成三維人體方位偵測。首先，任一張影像的前景人物利用一個基於前後影像比值而建立之統計背景模型抽取出來，並將抽取出來影像轉換成二值化的影像格式，進而獲得前景人物的輪廓。透過人體輪廓樣板比對及線性內插法，可以初步得到前景人物所朝的方向。當人臉朝向前方 $\pm 30°$ 以內時，也可透過雙眼與臉的三角幾何關係來估算人體所朝的方向。經實驗證明，我們提出的方法對於人體方位偵測的準確度相當高，平均角度誤差低於 $4°$。

Subject Direction Detection Using the Silhouette and Face

STUDENT: Tsung-Han Li   ADVISOR: Dr. Jyh-Yeong Chang

Institute of Electrical and Control Engineering
National Chiao-Tung University

ABSTRACT

In recent years, the advancement of robot technology brings robot into human daily activities, and makes robot as an essential part of modern life. The interaction between human and robot has become more and more frequent. When robot interact with human, the information of the position, distance, and direction of a human becomes important.

In the thesis, we combine 2D image silhouette matching and face direction detection to detect a human direction. Firstly, a foreground subject is extracted and converted to a binary image by a statistical background model. Then we obtain the silhouette of the foreground subject. Linear interpolation on silhouette matching results can be utilized to detect the human direction. When one's face is within $\pm 30^{\circ}$, one's pupils and their geometric relationship can also be exploited to estimate one's direction. By numerical simulation, we have obtained a high accuracy, less than 4° on the average, in (or on) subject direction detection.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. Jyh-Yeong Chang for valuable suggestions, guidance, support and inspiration he provided. Without his advice, it is impossible to complete this research. Thanks are also given to all of my lab members for their suggestion and discussion.

Finally, I would like to express my deepest gratitude to my family for their concern, supports and encouragements.
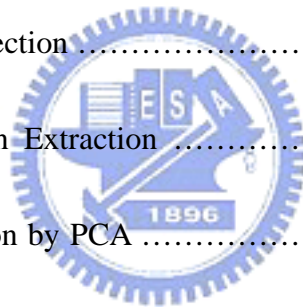
Contents

List of Figures

viii

List of Tables

Chapter 1　Introduction

1.1　Motivation

In recent years, the advancement of robot technology brings robot into human daily activities, and makes robot as an essential part of modern life. The interaction between human and robot has become more and more frequent. For many intelligent robots, computer vision is an important resource that enable robot autonomously to do several tasks to interact with human. For example, in home care, when human need the robot to bring a glass of water, the robot need to identify both the position and direction of human. Therefore, when robot interact with human, the direction of human becomes crucial.

The human visual system has an uncanny ability to recognize objects from single views, even when presented monocularly under a fixed viewing condition. However, for computer vision, it is a very challenging task to recognize objects in different directions from 2D image. Therefore, model construction and shape recognition of 3-Dobjects from 2-D silhouettes have been important areas in computer vision.

Ample literatures on 3D object recognition from 2D views have been proposed in the past few years. Hayar *et al*. [1] use principal component analysis (PCA) on a set of object views to generate a distance between an unknown view and prototypical views. Cootes and Taylor [2] proposed an active appearance model (AAM) which learns a statistical model by training on a series of 2D images. Koenderink and Doom [3, 4] proposed the aspect graph representation which identifies regions of the viewing sphere. In fact, they have attempted to identify unknown objects (from their 2-D silhouettes) by comparing them with the models in a library. As a consequence,

the storage requirement is very large, and can not detect object direction accurately.

The silhouette of a subject usually conveys insufficient 3-D information. When the silhouette of a subject is extended into 3-space along the corresponding viewing direction to from a cylinder, one may only infer that the subject is bounded by the cylinder. This problem is resolved by intersecting the bounding cylinders from different views. Chien and Aggarwal [5] have developed a scheme to generate the octree of an object from the silhouettes obtained from multiple viewpoints.

In this thesis, we present a technique to detect a 3-D subject direction from silhouettes of subject from multiple viewpoints, and combine the direction of subject's face, when the one faces forward within $\pm 30^{\circ}$, to estimate subject direction. We recovered the orientations known subjects from the silhouette of frontal views, which were taken from the PTZ camera. Then we use 2-D contour matching process and linear interpolation to estimate the direction of subject. In order to enhance the accuracy of direction detection, we also further fuse the face direction detection if face is available and suitable. As a consequence, we can identify the known subject's orientation. Our subject direction detection system flowchart is illustrated in Fig. 1.1 below. The proposed system can be separated into four components.
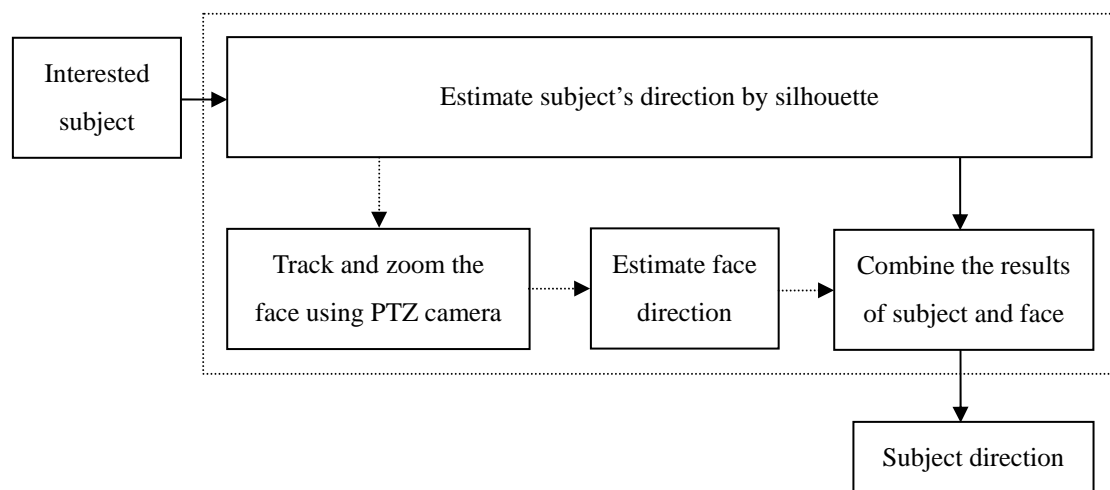
Fig 1.1   The flowchart of our subject direction detection system.

1.2 Background Modeling and Foreground Subject Extraction

Background subtraction is widely used for detecting moving objects from image frames of static cameras. The rationale of this approach is to detect the moving objects by the difference between the current frame and a reference frame, often called the "background image," or "background model." A review is given in [6] where many different approaches were proposed in recent years. Most of this work has been based on background subtraction using color or luminance component. In these approaches, difference between the coming frame and the background image is performed to detect foreground objects. W^4 [7] is a famous one to be noted. It records the maximum and minimum luminance and the maximum inter-frame difference in every position of a frame in a background video. Then every pixel of the image frame subtracts the maximum and minimum luminance at this position. If the pixel's absolute value of this difference is larger than the maximum inter-frame difference, the pixel is a foreground.

Background subtraction is extremely sensitive to dynamic scene changes due to illumination change. In order to solve the effect of varying luminance conditions, we develop a method which is robust to the illumination changes, which utilizes frame ratio in the luminance component.

After building a background model, we can extract foreground subject from video frames by subtracting each pixel value of background model from that of current image frame. The resulting image is converted to a binary one by setting a threshold. The binary image mainly contains foreground subject with only little noise. Therefore, we can set a threshold in the histogram of the binary image to extract a rectangle image, which is the most resemble shape of a person, of the target subject. The rectangle image is resized to become uniform in the measurements.

1.3　Subject Direction Detection

In most image processing and pattern recognition, the problem in picture processing is the classification of objects in a scene. Once the object is isolated in the scene, the goal is to describe or classify the object. For most cases, classification can be carried out from the boundary of the object, it is thus natural to retain the boundary for further analysis of the object. We will utilize the object's silhouettes to recognize its direction.

There are many techniques available to describe closed curves, the chain encoding used by Freeman [8] and the polygonal approximation proposed by Pavlidis [9]. The medial axis transform (MAT) uses the skeleton to describe a closed curve [10]-[12]. Fourier Descriptor (FD) is one of these techniques which is useful for describing the shape of a closed, planar figure. The main advantage of Fourier descriptor is the invariance to translation, rotation, scaling of the observed object and starting point. Thus will shape description become independent of the relative position and size of the object in the input image. Therefore, we utilize the Fourier descriptor to detect the subject's direction. In addition, we utilize the Linear Discriminant Analysis (LDA) to optimize the class separability and improve the classification performance. Our Deteciontion method combines Fourier descriptor and linear discriminant analysis which are described as follows.

The Fourier descriptor [13]-[15] is a well-known method of describing the shape of a closed figure which first suggested by Cosgriff [16]. It provides a means for representing the boundary of a two-dimensional shape. The idea behind FD is that a closed curve may be represented by a periodic function of a continuous parameter. Namely, the curve can be expressed in a Fourier series expansion, i.e., by a set of Fourier coefficients of this function. The coefficients in this collection are referred to

as "Fourier descriptors" (FD's). Higher order terms represent changes in direction of the curve over very small arc lengths and their elimination will probably reduce noise and serve to accentuate lower order terms which contain more macroscopic information on the shape. We can use these descriptors for pattern recognition applications.

Linear discriminant analysis [17] easily handles the case where the within-class frequencies are unequal and their performances has been examined on randomly generated test data. This method maximizes the ratio of between-class variance in the same time minimize within-class variance in any particular data set thereby guaranteeing maximal separability. Including LDA method in our subject direction detection increases the accuracy greatly.

1.4 Face Direction Detection

As described above, we introduce Fourier descriptors for human direction detection. In order to enhance the accuracy of human direction detection, it may be helpful to detect from one's face direction through locating one's pupils and their geometric relationship in the face to be utilized in estimating the face direction. Combining face direction estimation with human direction estimation escalates the accuracy greatly.

When we take an image from the front of a subject, facial image is available. If we search for the eye from the whole face image, the background would deteriorate the result frequently. In order to reduce the region for eye search and find the eye position more accurately, we carry out the face segmentation first. There are many methods proposed for face detection in recent years. According to the survey of to Hjelmas and Low [18], the major approaches are listed in Table I. In this thesis, we adopted the approach proposed by Garcia and Tziritas [19], in which color feature is used to identify a human face in an image. This is feasible because human faces have a special color distribution that differs significantly (although not entirely) from those of the background objects. Hence, this approach requires a color map that models the skin-color distribution characteristics.

After we separate the skin color region from a mug image, it is still difficult to search for the iris from their large skin color region. Because the search region is still too large. Obviously, before any of the components of the eye can be extracted and fitted, the eyes have first to be located in the face. Donato *et al.* [20] compared several techniques for recognizing upper face images and lower face images. These techniques include optical flow, principal component analysis, independent component analysis, local feature analysis, and Gabor wavelet representation. The

best performance was achieved by Principle Component Analysis (PCA). We therefore start with the PCA technique to estimate eye locations so that we can set up efficient search for the iris.

TABLE I

MAJOR FACE DETECTION APPROACHES

Authors	Year	Approach	Feature Used	Head Pose	Test Databases
Féraud et al. [21]	2001	Neural Networks	Motion; Color; Texture	Frontal and profile	Sussex; CMU; Web images
Maio et al. [22]	2000	Facial Templates; Hough Transform	Texture; Directional images	Frontal to near frontal	Static images
Garcia et al. [19]	1999	Statistical wavelet analysis	Color; wavelet coefficients	Frontal to profile	MPEG videos
Wu et al. [23]	1999	Fuzzy color models; Template matching	Color	Frontal	Still color images
Sung et al. [24]	1998	Learning	Texture	Frontal	Mug shots; CCD pictures; Newspaper scans
Yang et al. [25]	1998	Multiscale segmentation; color model	Skin Color; intensity	Frontal	Color pictures
Yow et al.	1997	Feature; Belief networks	Geometrical facial feature	Frontal to profile	CMU

1.5 Combining Subject Direction with Face Direction

In this thesis, we propose a novel method to estimate the direction of subject using subject's silhouette and face region. When we get an image including whole human body, the face region is too small to recognize its direction. For this reason, after subject direction detection by silhouette, we automate Pan-Tilt-Zoom (PTZ) camera to track and then zoom the face in to a level at least 80% vertically, which is ready for face direction detection.

When the subject direction angle is almost frontal to the camera, the face is more accurately than that from subject's silhouette. It is wise to fuse these two estimation angles for better direction detection.

1.6 Thesis Outlines

The contents of this thesis are organized as follows. In Chapter 2, the detection algorithm of subject direction will firstly be described. In this chapter, we discuss the process to estimate the direction of a subject. In Chapter 3, we propose a method to estimate the face direction to enhance the direction estimation accuracy of subject interested. Then we do some simulations and show our experimental results in Chapter 4. At last, we give some conclusions and discuss future work to be investigated further in Chapter 5.

Chapter 2   Subject Direction Detection

In this chapter, we propose an approach which combines Fourier Descriptor (FD) based on Fourier series analysis, with Linear Discriminate Analysis (LDA). This method can be used to extract subject features and to optimize the class separability of different subjects by their contours. Firstly, we utilize Fourier descriptors to discriminate different direction of target subject. Secondly, we use LDA to maximize between-class and in the same time minimize within-class variations to improve the classification performance. At last, we calculate the Euclidean distance between the subject of input image and out model which built from Fourier descriptors to estimate the direction angle of the subject.

2.1   Pre-processing for Subject Extraction

2.1.1   Background Modeling

We assume the image captured by a camera can be described as

$$I_i(x, y) = S_i(x, y) r_i(x, y), \tag{2.1}$$

where I_i is the intensity of the scene, S_i is the spatial distribution of source illumination, r_i is the distribution of scene reflectance, (x, y) is the location of a pixel in the image and i is the image sequence index. If the camera is fixed stationary and moving subjects are not permitted to show up in the scene, the reflectance of the background may remain the same at any time. That is

$$r_i(x, y) = r(x, y), \tag{2.2}$$

9

Although the reflectance is not changed, the effect of illumination is still going on. The frame ratio between two consecutive frames can respectively be written as

$$
\begin{aligned}
\log\!\left(\frac{I_i(x,\,y)}{I_{i-1}(x,\,y)}\right) &= \log\!\left(\frac{S_i(x,\,y)\,r(x,\,y)}{S_{i-1}(x,\,y)\,r(x,\,y)}\right) \\
&= \log\!\left(\frac{S_i(x,\,y)}{S_{i-1}(x,\,y)}\right) \\
&= \log(S_i(x,\,y)) - \log(S_{i-1}(x,\,y)),
\end{aligned}
\tag{2.3}
$$

where I is the intensity of captured images, S is the spatial distribution of source illumination.

We propose to utilize the frame ratio to build the background model. Each pixel of background scene is characterized by three statistics: minimum intensity value $n(x,\,y)$, maximum intensity value $m(x,\,y)$ and maximum inter-frame ratio $d(x,\,y)$ of a background video. Because these three values are statistical, we need a background video, without any moving objects, for background model training. Let I be an image frame sequence and contains N consecutive images. $I_i(x,\,y)$ be the intensity of a pixel which is located at $(x,\,y)$ in the i-th frame of I. The background model, $[\,m(x,\,y),\,n(x,\,y),\,d(x,\,y)\,]$, of a pixel is obtained by

$$
\begin{bmatrix} m(x,\,y) \\ n(x,\,y) \\ d(x,\,y) \end{bmatrix} =
\begin{cases}
\begin{bmatrix} \max\limits_{i}\{I_i(x,\,y)\} \\ \min\limits_{i}\{I_i(x,\,y)\} \\ \max\limits_{i}\{I_i(x,\,y)/I_{i-1}(x,\,y)\} \end{bmatrix} & \text{if } I_i(x,\,y)/I_{i-1}(x,\,y) \ge 1 \\[20pt]
\begin{bmatrix} \max\limits_{i}\{I_{i-1}(x,\,y)\} \\ \min\limits_{i}\{I_i(x,\,y)\} \\ \max\limits_{i}\{I_{i-1}(x,\,y)/I_i(x,\,y)\} \end{bmatrix} & \text{otherwise}
\end{cases}
\tag{2.4}
$$

$$
i = 1,\,2,\,\ldots,\,N.
$$

2.1.2 Foreground Subject Extraction

Foreground subjects can be segmented from every frame of the video stream. Each pixel of the video frame is classified to either a background or a foreground pixel by the difference between the background model and a captured image frame. We utilize the maximum intensity $m(x, y)$, minimum intensity $n(x, y)$ and maximum inter-frame ratio $d(x, y)$ of the training background model to segment a foreground by

$$B(x, y) = \begin{cases} 0, \text{ a background pixel} & \text{if } \begin{cases} I_i(x, y)/m(x, y) \ < \ kd(x, y) \\ \text{or} \\ I_i(x, y)/n(x, y) \ < \ kd(x, y) \end{cases} \\ \\ 255, \text{ a foreground pixel} & \text{otherwise} \end{cases} \tag{2.5}$$

where $I_i(x, y)$ be the intensity of a pixel which is located at (x, y), $B(x, y)$ is the gray level of a pixel in a binary image and k is a threshold. Threshold k is determined by experiments according to difference environments. The value of k affects the mount of information retained in binary image B.

According to binary image B, we extract the region of foreground subject to minimize the image size. Foreground region extraction can be accomplished by simply introducing a threshold on the histograms in X and Y direction. Fig. 2.1 shows an example of foreground region extraction. We utilize the binary image and project it to X and Y directions. The interested section has higher counts in the histogram. We obtain the boundary coordinates x_1, x_2 of X axis and y_1, y_2 of Y axis from the projection histogram. We can use these boundary coordinates as the corner of a rectangle to extract foreground region. Fig. 2.2 is the extracted foreground region.

11
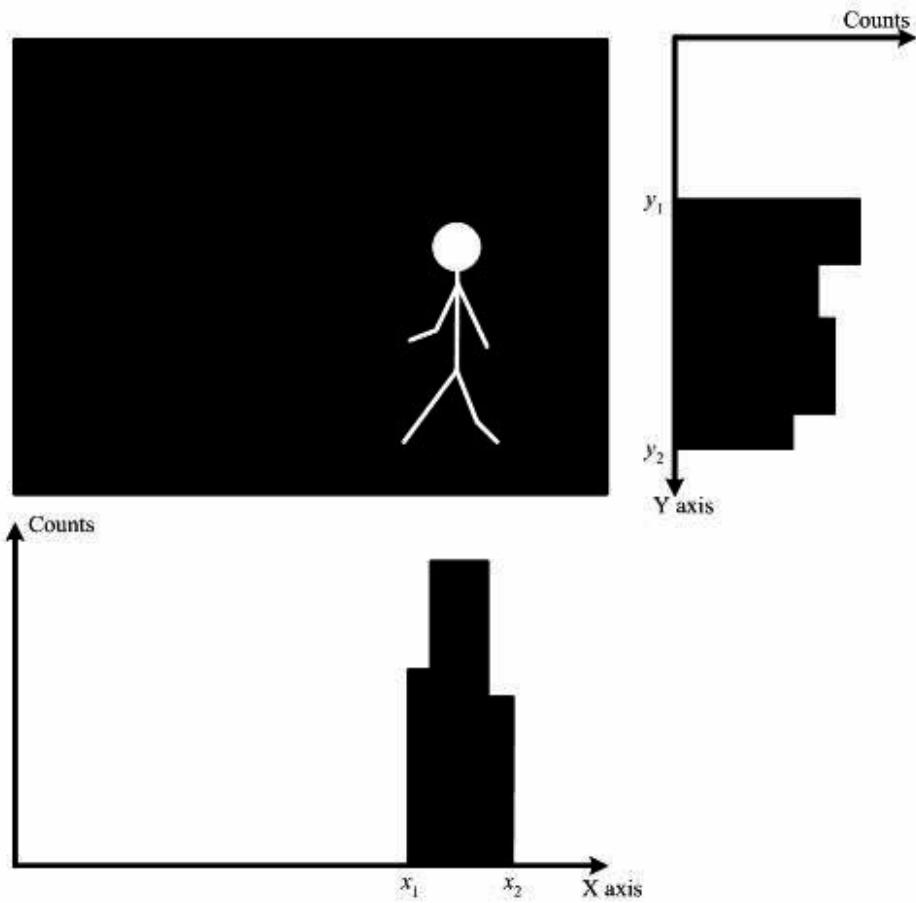
Fig. 2.1   Histogram of binary image projection in X and Y direction.

Fig. 2.2   The binary image of extracted foreground region.

2.2 Model Establish Using Fourier Descriptor

2.2.1 Fourier Descriptor Review

Fourier Descriptors is a useful implement for describe closed curve shape which obtained from subject contour. We can obtain a periodic function along the closed curve and this function can represent by a Fourier series. Fourier descriptors was first suggested by Cosgriff [16] who represented an image using Fourier series in 1960. The follow-up development work is established the foundation by much scholars.

Zahn and Roskies [13] introduced Fourier descriptors using normalized arc-length (assuming boundary is traced counter clock-wise). They expressed the close curve into the function of arc-length. With the accumulation change in this curve each point direction, we can obtain the Fourier series.

2.2.2 Pre-processing for Subject Contour Extraction

There are many operations based on morphology, such as dilation operator, erosion operator, opening, closing, segmentation and watersheds. We apply the morphological filtering by using the cascaded opening and closing operations, which consist of dilation and erosion operators defined in the following:

$$X \oplus B = \big\{ (x, y) + (u, v) \mid (x, y) \in A, \text{ and } (u, v) \in B \big\}, \tag{2.6}$$

$$X \ominus B = \big\{ w \mid B_w \subseteq A \big\}, \tag{2.7}$$

where $\oplus$ and $\ominus$ are erosion and dilation respectively. To combine $\oplus$ and $\ominus$, we have the opening and closing which are defined as:

13

$$X \circ B = (X \ominus A) \oplus B, \tag{2.8}$$

$$X \bullet B = (X \oplus A) \ominus B, \tag{2.9}$$

The examples of opening operation and closing operation are illustrated in Fig. 2.3.

(a)

(b)
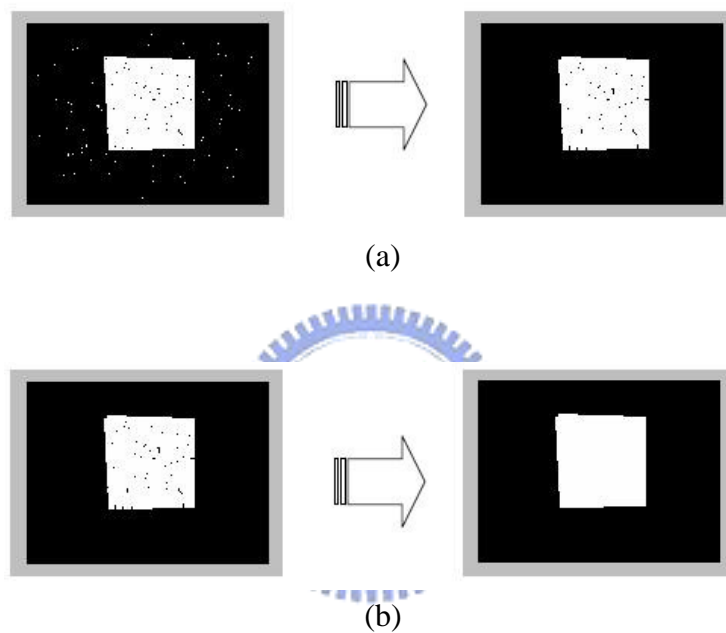
Fig. 2.3   Morphology operation. (a) opening operation. (b) closing operation.

Opening operation removes small noise. Closing operation repairs the hole inside the foreground. To repair the hole with a larger region, morphology closing must use a larger mask. But a larger mask will also dilate the boundary of human foreground. Therefore, we use region filling method to fill the larger region where inside the foreground and unable to dilate the boundary.

Assume *A* denote a set containing a subset whose elements are 8-connected boundary points of a region. Beginning with a point *p* inside the boundary, the objective is to fill the entire region with 1's. If we adopt the convention that all

nonboundary (background) points are labeled 0, then we assign a value of 1 to p to begin. The following procedure then fills the region with 1's:

$$\{p\} = X_0, X_1, X_2, \dots, X_k = X_{k+1}$$

$$\tag{2.10}$$

$$X_k = (X_{k-1} \oplus B) \bigcap \overline{A} \qquad k = 1, 2, 3, \dots, n$$
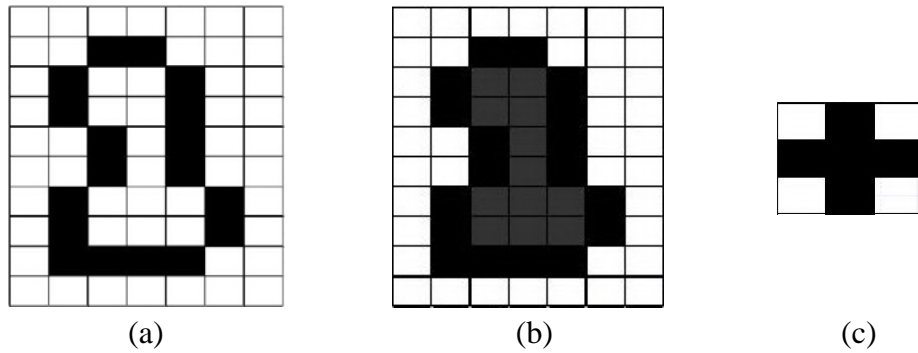
Fig. 2.4 (a) Original image A. (b) Result of filling all region. (c) Symmetric structuring element B.

where $X_0 = p$, and B is the symmetric structuring element. The algorithm terminates at iteration step k if $X_k = X_{k-1}$. The set union of X_k and A contains the filled set and its boundary. The dilation process would fill the entire area. Therefore, we can extract the boundary of the original image.

Then we have to go clockwise around the contour and recode the image boundary coordinate for Fourier descriptors analysis. Chain codes are used to represent a boundary by a connected sequence of straight-line segments of specified length and direction. Typically, this representation is based on 4- or 8-connectiviry of the segments (we use 8-connectivity in our research). The direction of each segment is coded by using a numbering scheme such as the ones shown in Fig. 2.5.
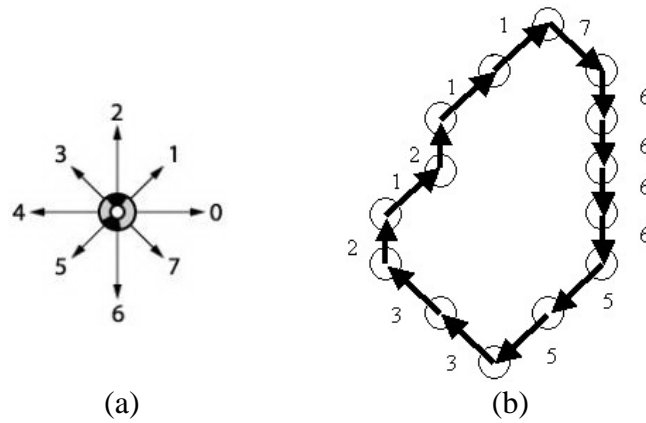
Fig. 2.5   (a) The direction of each segment. (b) 8-directional chain code.

2.2.3   Zahn and Roskies' Cumulative-Angle Approach

In this section, we introduce Zahn and Roskies' Cumulative-Angle Approach to represent Fourier descriptors. We assume γ is a clockwise-oriented simple closed curve with parametric representation $(x(l), y(l))$. Let $(x(0), y(0))$ be the starting point and we denote $Z(l) = x(l) + jy(l)$, where l is the arc length of the starting point to $Z(l)$ and $0 \le l \le L$. Denote the angular direction of γ at point l by the function $\theta(l)$ and let $\delta_0 = \theta(0)$ be the absolute angular direction at the starting point $Z(0)$. We now define the cumulative angular function $\phi(l)$ as the net amount of angular bend between starting point and point l (see Fig. 2.6).
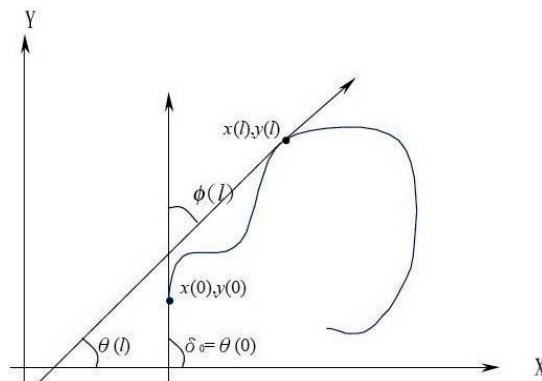
Fig. 2.6   Parameters of a curve representation. $\theta(l)$ is angular direction of point. $Z(l)$, $\phi(l)$ is the cumulative angular between starting point and point l.

16

With this definition $\phi(0)=0$ and $\phi(l)+\delta_0$, is identical to $\phi(l)$ except for a possible multiple of 2π. Besides, It is not hard to see that $\phi(L)=-2\pi$ because all smooth simple closed curves with clockwise orientation have a net angular bend of -2π. As a result, $\phi(L)$ does not convey any shape information. The domain of definition $[0, L]$ of $\theta(l)$ simply contains absolute size information and we would like to normalize to the interval $[0, 2\pi]$ which is standard for periodic functions. Hence we define a normalized variant $\phi^*(t)$ whose domain is $[0, 2\pi]$ and such that $\phi^*(0)=\phi^*(2\pi)=0$. The relation between t and l is $t=2\pi l/L$. The formal definition is

$$\phi^*(t) = \phi\left(\frac{Lt}{2\pi}\right) + t, \tag{2.11}$$

and $\phi^*(t)$ is a periodic function which is invariant under translations, rotations and changes of perimeter L.

We now expand ϕ^* as a Fourier series

$$\phi^*(t) = \mu_0 + \sum_{k=1}^{\infty} \left(a_k \cos kt + b_k \cos kt\right), \tag{2.12}$$

In polar form, the expansion is

$$\phi^*(t) = \mu_0 + \sum_{k=1}^{\infty} A_k \cos\left(kt - \alpha_k\right), \tag{2.13}$$

where (A_k, α_k) are polar coordinates of (a_k, b_k). These numbers A_k and α_k are the Fourier descriptors for curve γ and are known respectively as k-th harmonic amplitude and phase angle.

According to Euler formula, we can use the polar coordinates $x=r\cos\theta$ and $y=r\sin\theta$ to rewrite the complex parameter $z=x+jy$ as

$$z = r(\cos\theta + j\sin\theta) = re^{j\theta}, \tag{2,14}$$

Then we introduce Zahn and Roskies' derive formulas for the Fourier coefficients $\{a_k, b_k\}$ and μ_0 when γ is a polygonal curve. We assume the curve γ has m vertices $V_0, \cdots, V_{m-1}$ and that the edge (V_{i-1}, V_i) has length Δl_i. The change in angular direction at vertex V_i is $\Delta\phi_i$ and $L = \sum_{i=1}^{m} \Delta l_i$. With these definitions as shown in Fig. 2.7, it is not hard to verify that

$$\phi(l) = \sum_{i=1}^{k} \Delta\phi_i \qquad \text{for} \quad \sum_{i=1}^{k} \Delta l_i \le l < \sum_{i=1}^{k+1} \Delta l_i, \tag{2.15}$$

and

$$\phi(l) = 0 \qquad \text{for} \quad 0 \le l < l_1, \tag{2.16}$$
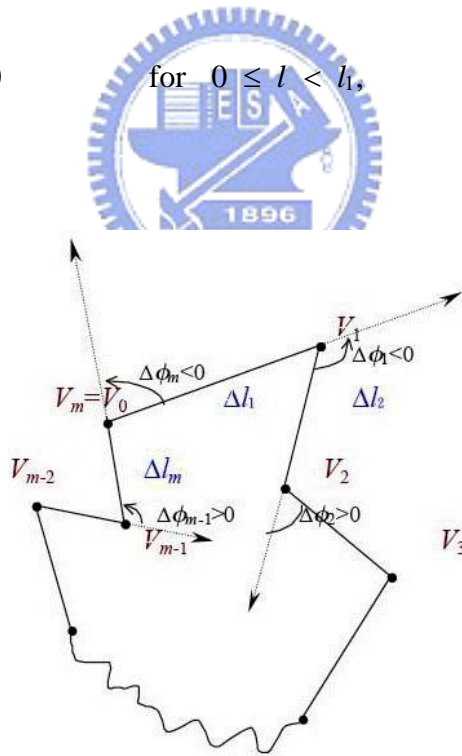
Fig. 2.7   Simple representation of a closed polygonal curve; Δl_i is edge length and $\Delta\phi_i$ is the angle between two edges.

Expanding ϕ^* we get $\phi^*(t) = \mu_0 + \sum\limits_{n=1}^{\infty} (a_n \cos nt + b_n \cos nt)$ where

$$\mu_0 = \frac{1}{2\pi} \int_0^{2\pi} \phi^*(t)\, dt, \qquad (2.17)$$

and

$$a_n = \frac{1}{\pi} \int_0^{2\pi} \phi^*(t) \cos nt\, dt, \qquad (2.18)$$

$$b_n = \frac{1}{\pi} \int_0^{2\pi} \phi^*(t) \sin nt\, dt, \qquad (2.19)$$

Note that $\phi^*(t) = \phi\!\left(\dfrac{Lt}{2\pi}\right) + t$ and change the variable $\lambda = \dfrac{Lt}{2\pi}$ we can obtain

$$\mu_0 = \frac{1}{L} \int_0^L \phi(\lambda)\, d\lambda + \pi, \qquad (2.20)$$

$$a_n = \frac{2}{L} \int_0^L \left(\phi(\lambda) + \frac{2\pi\lambda}{L}\right) \cos \frac{2\pi n\lambda}{L}\, d\lambda, \qquad (2.21)$$

$$b_n = \frac{2}{L} \int_0^L \left(\phi(\lambda) + \frac{2\pi\lambda}{L}\right) \sin \frac{2\pi n\lambda}{L}\, d\lambda, \qquad (2.22)$$

Since $\phi(l)$ is a step function, after some manipulations we can obtain

$$\mu_0 = -\pi + \frac{1}{L} \sum_{k=1}^{m} l_k \Delta\phi_k, \qquad (2.23)$$

$$a_n = \frac{-1}{n\pi} \sum_{k=1}^{m} \Delta\phi_k \sin\frac{2\pi nl_k}{L}, \qquad (2.24)$$

$$b_n = \frac{1}{n\pi} \sum_{k=1}^{m} \Delta\phi_k \cos\frac{2\pi nl_k}{L}, \qquad (2.25)$$

Where

$$l_k = \sum_{i=1}^{k} \Delta l_i.$$

19

The final forms of the expressions for a_n, b_n are especially appealing because of their similarity and also because $\Delta\phi_k$, represents the angular change (bend) in the curve's direction at the k-th polygonal vertex, and l_k is the arc length from the starting vertex to the k-th vertex.

It is clear from these expressions alone that the Fourier coefficients (a_n, b_n) contain no information relating to absolute position or rotational orientation of the curve. In the amplitude/phase angle form of the Fourier series

$$\phi^*(t) = \mu_0 + \sum_{n=1}^{\infty} A_n \cos(nt - \alpha_n), \qquad (2.26)$$

Coefficient pair (A_n, α_n) are the polar coordinates for the point (a_n, b_n). Coefficient A_n is called the n-th harmonic amplitude and α_n is the nth harmonic phase angle. Of course when $A_n = 0$ the n-th term vanishes and α_n is undefined.
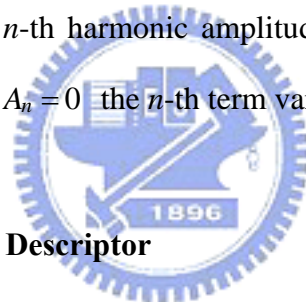
2.2.4   Properties of Fourier Descriptor

For a specific curve, the main advantage of Fourier descriptors is the invariance to translation, rotation, scaling of the observed object and starting point. Thus will shape description become independent of the relative position and size of the object in the input image. To be more specific, distance between object and camera and placement of the object relative to the optical axis of image acquisition system will not affect values of the Fourier descriptors. In fact, Fourier descriptors are not immediate sensitivity to this change. But this change are related to simple operations of the boundary's Fourier descriptors, as summarized in Table. 3.1.

● Translation

If we translate the object, we are really just adding some constant to all of the values of $x(l)$ and $y(l)$. Hence, we only change the zero-frequency component. Mean position only, nothing about the shape. So, except for the zero-frequency component, Fourier descriptors are translation invariant.

● Rotation

In complex analysis that rotation in the complex plane by angle ϕ is multiplication by $e^{j\phi}$. Thus, rotation about the origin of the coordinate system only multiplies the Fourier descriptors by $e^{j\phi}$.

● Scaling

Suppose that we resize the object. That's equivalent to simply multiplying $x(l)$ and $y(l)$ by some constant. Hence, that is just multiplication of the Fourier descriptors by the same constant.
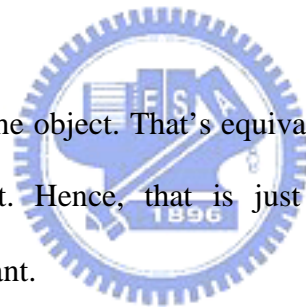
● Start Point

According to our discussion of the Fourier transform that translation in the spatial domain is a phase-shift in the transform. Thus, the harmonic amplitude is invariant to the start point, and the phase part shifts accordingly.

TABLE II

PROPERTIES OF FOURIER DESCRIPTORS WHEN BOUNDARY $u(t)$ CHANGED FROM

TRANSLATION, ROTATION, SCALING, AND STARTING POINT

Transformation	Boundary	Fourier Descriptor
Identity	$u(t)$	a_n
Translation	$\tilde{u}(t) = u(t) + Z$	$\tilde{a}_n = a_n + Z\delta_n$
Rotation	$\tilde{u}(t) = u(t)e^{j\phi}$	$\tilde{a}_n = a_n e^{j\phi}$
Scaling or Zooming	$\tilde{u}(t) = Ru(t)$	$\tilde{a}_n = Ra_n$
Starting point	$\tilde{u}(t) = u(t - t_0)$	$\tilde{a}_n = a_n e^{jnt_0}$

2.3 Linear Discriminant Analysis

In this section, we introduce LDA to optimize the class separability of different objects by their contours which has been described in Section 2.2. Linear Discriminant Analysis (LDA) is a classic method of classification which seeks to find a linear transformation by maximizing the between-class variance and minimizing the within-class variance, has proved to be a suitable technique for discriminating different pattern classes. The purpose of LDA is to find a vector $\mathbf{W}$. We can map the input data through $\mathbf{W}$ to a new coordinate $\mathbf{z}$ that has a larger separability.

Assume that there are c training classes to be learnt. Each class represents a direction of subject that obtain form Fourier descriptor analyze. $y_{i,j}$ is the j-th vector in class i, and N_i is the number of vectors in i-th class. The total number of training vectors is $N_T = N_1 + N_2 + \cdots + N_c$. This training set is represented by

$$
\begin{bmatrix} y_{1,1} & y_{2,1} & \dots & y_{c,1} \\ y_{1,2} & y_{2,2} & \dots & y_{c,2} \\ \vdots & \ddots & & \vdots \\ y_{1,N_1} & y_{2,N_2} & \dots & y_{c,N_c} \end{bmatrix} = [\Phi_1 \ \ \Phi_2 \ \ \dots \ \ \Phi_c],
\tag{2.27}
$$

The mean vector of the entire set is given by

$$
\mathbf{m}_y = \frac{1}{N_T} \sum_{i=1}^{c} \sum_{j=1}^{N_i} y_{i,j}, \qquad i = 1, 2, \dots, c; \ j = 1, 2, \dots, N_i
\tag{2.28}
$$

and the mean vector of the *i*-th class is represented by

$$
\mathbf{m}_i = \frac{1}{N_i} \sum_{y_{i,j} \in \Phi_i} y_{i,j},
\tag{2.29}
$$

Let $\mathbf{S}_w$ denote within-class matrix and $\mathbf{S}_b$ denote between-class matrix, then

$$
\mathbf{S}_w = \frac{1}{N_T} \sum_{i=1}^{c} \sum_{y_{i,j} \in \Phi} (y_{i,j} - \mathbf{m}_i)(y_{i,j} - \mathbf{m}_i)^{\mathrm{T}},
\tag{2.30}
$$

$$
\mathbf{S}_b = \frac{1}{N_T} \sum_{i=1}^{c} N_i (\mathbf{m}_i - \mathbf{m}_y)(\mathbf{m}_i - \mathbf{m}_y)^{\mathrm{T}},
\tag{2.31}
$$

where $\mathbf{S}_w$ represents the mean of within-class vectors distance and $\mathbf{S}_b$ represents the mean of between-class vectors distance. The objective is to minimize $\mathbf{S}_w$ and maximize $\mathbf{S}_b$ simultaneously, that is to minimize the criterion function known as the generalized Fisher linear discriminant function and given by

$$
\mathbf{J}(\mathbf{W}) = \frac{\mathbf{W}^{\mathrm{T}} \mathbf{S}_b \mathbf{W}}{\mathbf{W}^{\mathrm{T}} \mathbf{S}_w \mathbf{W}},
\tag{2.32}
$$

Because we want to get a maximal $\mathbf{J}(\mathbf{W})$, it is fine that the denominator is the smaller and the numerator is the bigger. It can utilize the Lagrange multiplier to solve $\mathbf{W}$ for

maximized $\mathbf{J(W)}$. We restrict the length of $\mathbf{W}$ to let denominator is 1, and to maximized the numerator.

$$
\begin{aligned}
c(\mathbf{W}) &= \mathbf{W}^{\mathrm{T}}\mathbf{S}_b\mathbf{W} - \lambda(\mathbf{W}^{\mathrm{T}}\mathbf{S}_w\mathbf{W} - 1) \\
&\Rightarrow \frac{dc}{d\mathbf{W}} = 2\mathbf{S}_b\mathbf{W} - 2\lambda\mathbf{S}_w\mathbf{W} = 0 \\
&\Rightarrow \mathbf{S}_b\mathbf{W} = \lambda\mathbf{S}_w\mathbf{W},
\end{aligned}
\tag{2.33}
$$

This is a generalized eigenvalue problem. When $\mathbf{S}_w$ have inverse, we can translate Eq. (2.33) as

$$
\mathbf{S}_w^{-1}\mathbf{S}_b\mathbf{W} = \lambda\mathbf{W},
\tag{2.34}
$$

Then we can solve the transformation matrix $\mathbf{W}$

$$
\mathbf{W} = \mathbf{S}_w^{-1}\mathbf{S}_b,
\tag{2.35}
$$

By using this basis, each point in FD can be further projected to another point in this new space by

$$
\mathbf{z}_{i,j} = \mathbf{W}y_{i,j},
\tag{2.36}
$$

where $\mathbf{z}_{i,j}$ represents the new point and $[z_{i,1}, \ldots, z_{i,N_i}]$ is the new trajectory in LDA space. We call this basis $[\mathbf{W}_1, \ldots, \mathbf{W}_c]$ is the LDA space transformation matrix. Following this analysis, different classes will be greatly separated which means that linear discriminant analysis is useful to separate different classes of samples.

2.4 Estimation of Subject Direction by Interpolation

Using Fourier descriptors and linear discriminant analysis, we can estimate the direction of interested subjects. After the FD and LDA processes on the input image. We can calculate the Euclidean distance between the subject of interests and built model of the subject direction. For example, we build seven directions of a subject including $0°, 30°, 60°, 90°, 120°, 150°,$ and $180°,$ as shown in Fig. 2.8. Then we calculate the Euclidean distances between the subject of interests and the seven direction models. The angels of the least two Euclidean distance models could represent the possible direction range of the subject. The direction of minimum distance model is more probable than the direction of the second minimum model, in which the leverage is depending on the ratio of these two minima. The subject's direction could be estimated by the linear interpolation on these two direction angles.
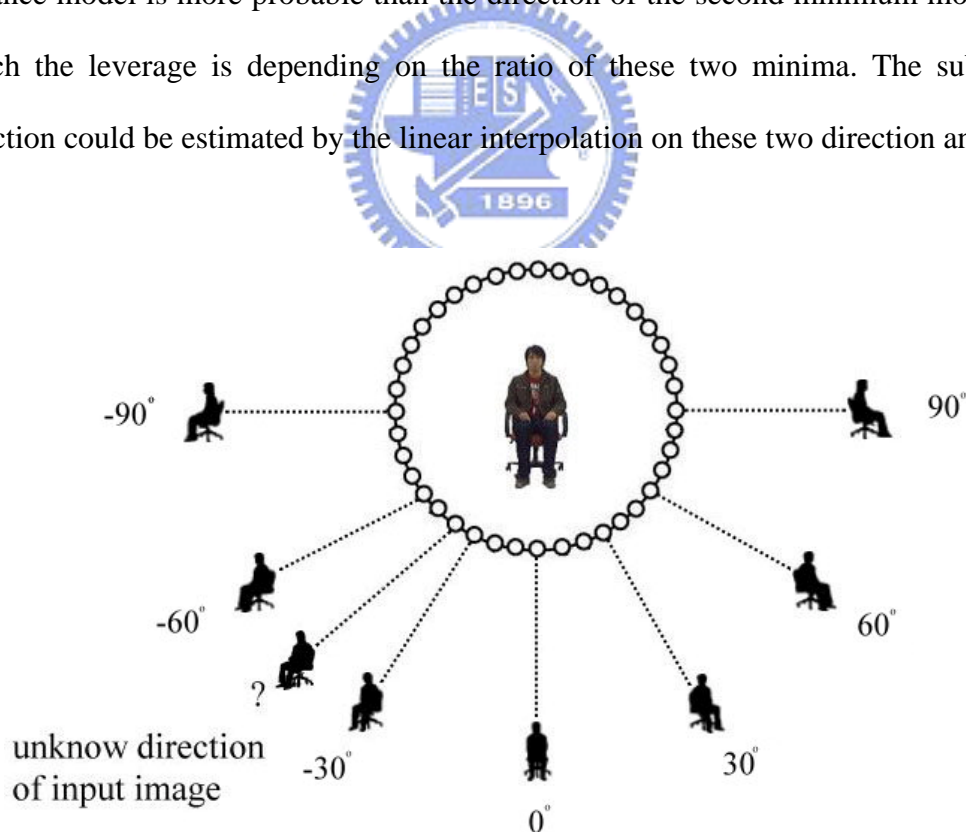
Fig. 2.8   The representation of a 3D subject by 2D views.

Chapter 3   Face Direction Detection

In this chapter, we propose a method to estimate the face direction of subject interests to enhance the accuracy of direction detection of target subject interests. Using image processing, the face angle will be estimated from the geometric relationship of pupils in the face. Firstly, we utilize skin detection formula in the YC_bC_r color space and edge detection to find head region. Secondly, we utilize PCA to find eyes position and then calculate the pupil centers for estimate the face direction. Consequently, we can combine object direction from FD and face direction to better detect the direction of subject/object of interests.

3.1   Head Region Extraction

3.1.1 YC_bC_r Color Space

Our purpose is to segment the head region for estimating face direction, the first stage in the head detection algorithm is using skin segmentation to reject as much "non-face" of the image as possible. There are two color models of segmenting the image based on skin color have been evaluated and used. The YC_bC_r model is naturally related to MPEG and JPEG coding. The HSV (Hue, Saturation, Value) model is used mainly in computer graphics. In [19], YC_bC_r and HSV color space for skin color segmentation have detailed description. The skin colors distribution in YC_bC_r and HSV color space is shown in Fig. 3.1.

From Fig. 3.1, we can conclude the skin color distribution in YC_bC_r color space is more centralize than HSV color space, and the advantage of converting the

image to the YC_bC_r color space is that the effect of luminosity can be decoupled with coloring components during our image processing. For this reason, we utilize YC_bC_r color space for skin color region detection.
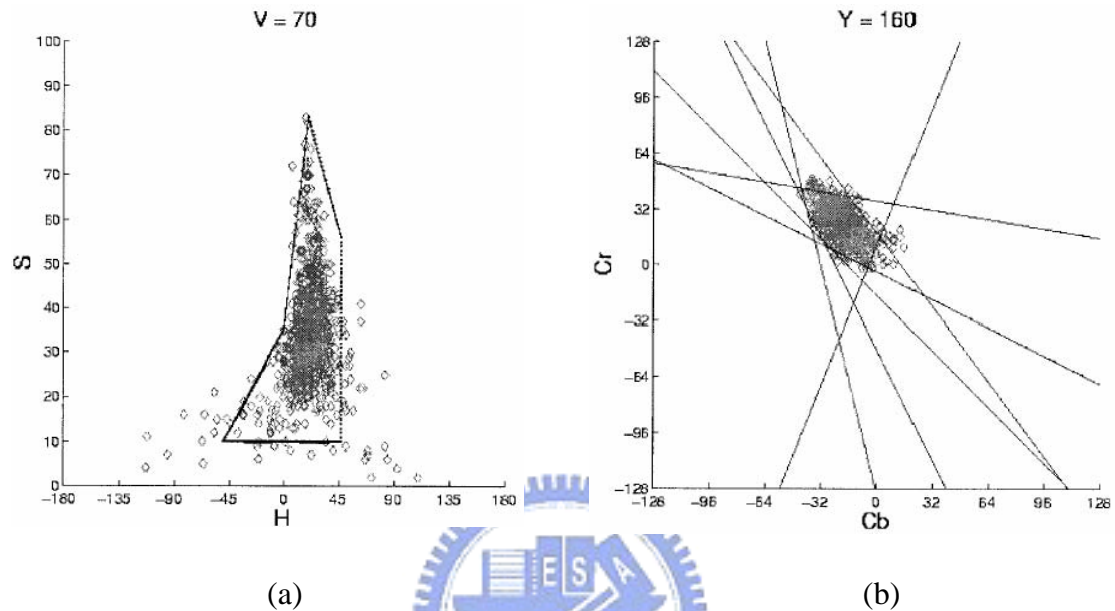
Fig. 3.1　(a) Sample skin colors distribution in HSV color space and (b) YC_bC_r color space.

In the RGB domain, each component, red, green and blue, of the picture element has a different strength value. However, in the YC_bC_r domain, the pixel's is given by the *Y*-component, the C_b (blue) and C_r (red) components. The color space of YC_bC_r, which revises the color space of YUV, can divided luminance component (*Y*), and two chromatic blueness component (C_b), redness component (C_r). The separation of the luminance is high. The following conversion matrix is used to convert the RGB image into *Y*, C_b and C_r components

$$\begin{cases} Y = 0.2989 \times R + 0.5866 \times G + 0.1145 \times B \\ C_b = 0.5674 \times (B - Y) \\ C_r = 0.7132 \times (R - Y) \end{cases} \tag{3.1}$$

3.1.2 Combine the Region of Skin Color Detection and Edge Detection

The first stage of the algorithm is to classify the pixels of the input image into skin region and non-skin region. To do this, we obtain a skin-color reference map in YC_bC_r color space. It has been proved that a skin-color region can be identified by the presence of a certain set of chrominance values (i.e., C_b and C_r) narrowly and consistently distributed in the YC_bC_r color space. In Fig. 3.1(b), the intersections of the adjusted bounding planes with the C_bC_r plane for $Y = 160$ are displayed. We report the respective ranges of C_b and C_r values that correspond to skin color defining the bounding planes, which subsequently define our skin-color reference map. The ranges to be most suitable for all the input images are

$$\begin{cases} \text{if } (Y > 128) \begin{cases} \theta_1 = -2 + \dfrac{256 - Y}{16}, \quad \theta_2 = 20 - \dfrac{256 - Y}{16} \\ \theta_3 = 6, \quad \theta_4 = -8 \end{cases} \\ \\ \text{if } (Y \leq 128) \begin{cases} \theta_1 = 6, \quad \theta_2 = 12 \\ \theta_3 = 2 + \dfrac{Y}{32}, \quad \theta_4 = -16 + \dfrac{Y}{16} \end{cases} \end{cases} \tag{3.2}$$

$$\begin{cases} C_r \geq -2 \times (C_b + 24) ; \quad C_r \geq -(C_b + 17) \\ C_r \geq -4 \times (C_b + 32) ; \quad C_r \geq 2.5 \times (C_b + \theta_1) \\ C_r \geq \theta_3 ; \quad C_r \geq 0.5 \times (\theta_4 - C_b) \\ C_r \leq \dfrac{220 - C_b}{6} ; \quad C_r \leq \dfrac{4}{3} \times (\theta_2 - C_b) \end{cases} \tag{3.3}$$

As one can notice, there are two sets of eight equations since $Y = 128$ depending on two areas of the color space, separated by the horizontal plane in order to

approximate the distribution borders in the light and dark extreme cases. An example of output image to illustrate the classification of the input image Fig. 3.2 is shown in Fig. 3.3.

Fig. 3.2   Input image.
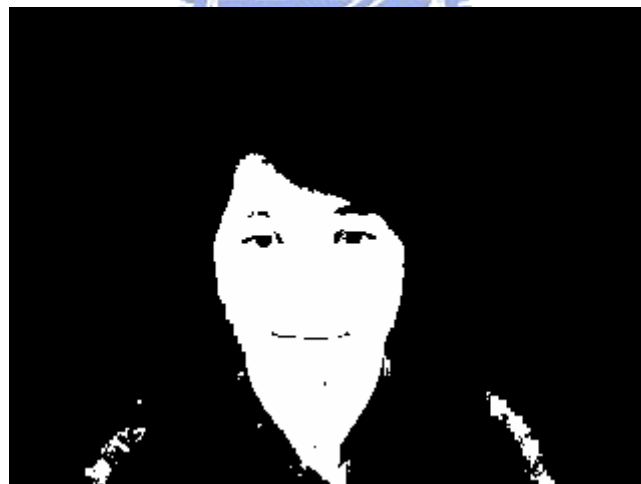
Fig. 3.3   Output image after segmented by skin-color map in the first stage.

The second stage of the algorithm is to combine the region of skin color detected and the result from edge detected. We can get the face complexion area after color segmentation described above, and we call this area face region. Because skin color segmentation is unable to get the whole region of head, thus we have to utilize the

edge detection method, then do the union with the face region. Thus, we can get a more complete head region. The resulting region might however contains the region of no interests (e.g. neck, shoulders, etc.). We only need head region to estimate the face direction, we can remove the region that we do not need. By the scheme above, Fig. 3.3 is the result of input image after edge extraction, and Fig. 3.5 is the head region.

Fig. 3.4   Output image after foreground extraction.

Fig. 3.5   Head region by combining the regions of skin and edge detection.

3.2 Eye Detection and Pupil Center Estimation Using PCA

3.2.1 PCA Review

Most approaches in computer recognition of faces and expressions have been focused on detecting individual features such as eyes, head outline, mouth, or defining a face model by position, size, and relationships among these features. Features extraction plays an essential role in the pre-processing stage. Principal component analysis (PCA) has been commonly used to face recognition problems [26]. Typical PCA algorithm is one of the main streams of research on face feature processing [27]. PCA has advantage over other face recognition schemes in its speed and simplicity. We utilized PCA in the pre-processing stage to extract features from input eye region image which has been extracted in Sections 3.1 and 3.2.

The basis of the input image space is composed of all single pixel vectors. However, the input image space is not an optimal space for face representation and categorization. The aim of applying PCA is to build an eye space which better describes the eye regions. The basis vectors of this eye space are called the principal components. These components will be uncorrelated and will maximize the variance accounted in the original basis. It can also reduce the dimension of the feature space. The computation complexity is thus reduced.

3.2.2 Computation of Eigeneyes

In the following, we summarize the steps to compute Eigeneyes.

● Step 1: obtain eye region images $I_1, I_2, \dots, I_M$ (training eyes)

important note: the eye region images must be the same *size*.

● Step 2: represent every input image as a vector Γ_i
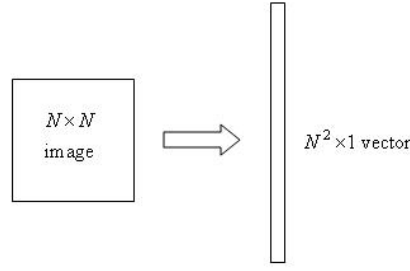
Fig. 3.6   Represent input eye region matrix to a vector.

● Step 3: compute the average eye vector Ψ

$$\Psi \; = \; \frac{1}{M} \sum_{i=1}^{M} \Gamma_i, \tag{3.4}$$

● Step 4: subtract the average eye vector Ψ

$$\Phi_i \; = \; \Gamma_i \; - \; \Psi, \qquad i \; = \; 1, 2, \ldots, M \tag{3.5}$$

● Step 5: compute the covariance matrix C:

$$C \; = \; \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T \; = \; AA^T, \qquad (N^2 \times N^2 \ \text{matrix}) \tag{3.6}$$

where $A \; = \; [\Phi_1 \ \Phi_2 \ \ldots \ \Phi_M]$ $\qquad (N^2 \times M \ \text{matrix})$

● Step 6: compute the eigenvectors u_i of AA^T

The matrix AA^T is very large $(N^2 \times N^2)$, thus it is not practical to compute the eigenvectors u_i of AA^T.

Step 6.1: consider the matrix A^TA $(M \times M$ matirx$)$

Step 6.2: compute the eigenvector v_i of A^TA

$$A^TAv_i = \lambda_i v_i, \tag{3.7}$$

Observe the relationship between u_i and v_i:

$$A^TAv_i = \lambda_i v_i \implies AA^TAv_i = \lambda_i Av_i \implies$$

$$CAv_i = \lambda_i Av_i \text{ or } Cu_i = \lambda_i u_i \qquad \text{where } u_i = Av_i$$

Thus, AA^T and A^TA have the same eigenvalues and their eigenvectors are

related as follows: $u_i = Av_i$

Note 1: AA^T can have at most N^2 eigenvalues and eigenvectors.

Note 2: A^TA can have at most M eigenvalues and eigenvectors.

Note 3: the M eigenvalues of A^TA correspond to the M largest eigenvalues of

AA^T and in the same way as eigenvectors.

Step 6.3: normalize u_i such that $\|u_i\| = 1$, compute M best eigenvectors of

AA^T for $u_i = Av_i$.

● Step 7: keep M eigenvectors which are corresponding to the M largest eigenvalues.

3.2.3 Representing Eyes onto PCA Basis

Each eye region image (minus the mean) Φ_i in the training set can be represent as a linear combination of the M eigenvectors:

$$\hat{\Phi}_i - mean = \sum_{j=1}^{M} w_j u_j, \qquad (w_j = u_j^T \Phi_j) \tag{3.8}$$

Each normalized training eye region image Φ_i is represent in this basis by a vector:

$$\Omega_i = \begin{bmatrix} w_1^i \\ w_2^i \\ \vdots \\ w_M^i \end{bmatrix}, \qquad i = 1, 2, \dots, M \tag{3.9}$$

3.2.4 Eye Region Recognition Using Eigeneyes

Input an unknown eye region image Γ with the same size as training image and follow these steps:

● Step 1: normalize Γ: $\Phi = \Gamma - \Psi$

● Step 2: project onto the eigenspace:

$$\hat{\Phi} = \sum_{i=1}^{K} w_i u_i, \qquad (w_i = u_i^T \Phi) \tag{3.10}$$

and representing by: $\Omega = \begin{bmatrix} w_1 & w_2 & \dots & w_k \end{bmatrix}^T$

● Step 3: compute the distance e_d (distance from eye space).

$$e_d = \left\| \Omega - \Omega^l \right\|, \qquad (\Omega^l \text{ is in the training set}) \tag{3.11}$$

34

3.2.5 Estimation the Coordinates of Pupil Center of the Eye

The next step is to find the pupil center of the eye region found by PCA. We use intensity to segment pupil region. An intensity field I_s are given by a threshold α owing to the low intensity of iris and highly contrast between iris and sclera as given by.

$$I_s(x, y) = \begin{cases} 1, & \text{if } I(x, y) \leq \alpha \\ 0, & \text{otherwise} \end{cases} \tag{3.12}$$

(a)　　　　　　(b)
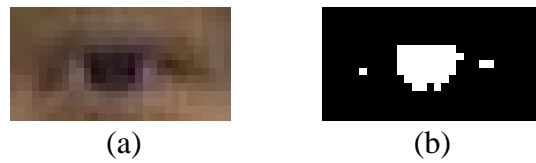
Fig. 3.7　An eye region image of the intensity field. (a) Input image (b) Intensity field.

Observe the region of iris in Fig. 3.7(b), the segmentation through pixel intensity thresholding have roughly found out the area of the pupil. But there exist a few low intensity pixels having segmented between iris and sclera, we utilize a rectangular mask to search the iris position. Moving the mask from top-left to down-right to record the number of pixel value is 1 in the mask which shown in Fig. 3.8(a). The position of the mask with largest number of 1 in the rectangular would be the best iris location as shown in Fig. 3.8(b). Therefore, the center of the mask is nearly and hence identified to be the center of the iris.

(a)　　　　　　(b)
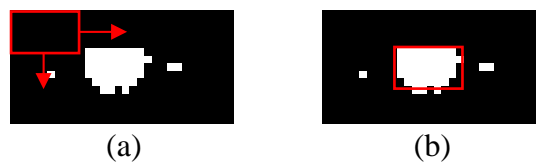
Fig. 3.8　(a) Simple representation of mask moving. (b) The position of the pixel value is 1 in the mask which is the most.

3.3 Estimation of Face Direction

After we calculate out the pupil center, face direction angle can estimated by using face and head model as shown in Fig. 3.9. Since the nose position is located near by the center of the eyes. We select the center of two pupils as the nose position on the horizontal axis. Therefore, the face direction angle ψ can be obtained by the following equation.

$$\begin{cases} a = (R + L) / 2 \\ b = (R + L) / 2 - R \end{cases} \Rightarrow \sin \psi = \frac{b}{a} = \frac{L - R}{L + R},$$

$$\psi = \sin^{-1} \frac{L - R}{L + R}, \tag{3.13}$$
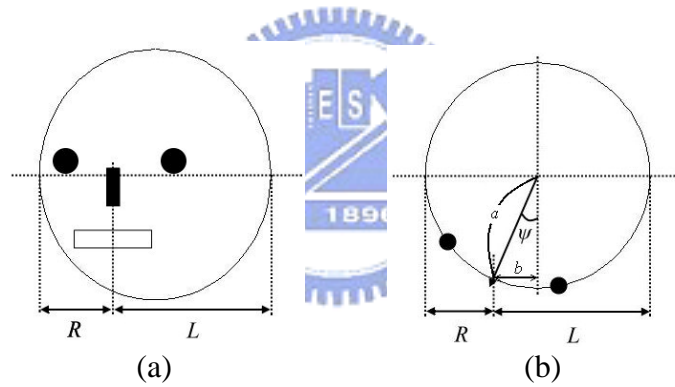
Fig. 3.9   Face and head model for estimating face direction angle. (a) Front view.

(b) Top view.

Chapter 4   Experimental Results

In our experiment, we test our system on images containing a static subject that have different direction. There are five persons with nineteen directions which a picture is taken from the front ranging from 90° to -90°. The image is taken in a lab at the 5th Engineering Building in NCTU. The light source is fluorescent lamps and is stable. The background is not complex, and the Sony EVI-D100 PTZ camera is set up at a location frontal to the person. The camera has a frame rate of thirty frames per second and the image resolution is 320×240 pixels. The environment of the lab is shown in Fig. 4.1.

Fig. 4.1   The scene environment of our system.

Each member sat on a chair on a location and oriented to one of nineteen directions. Besides, a video of background with no subject in the scene is adopted in our experiment and this is used as a background model. One model randomly chosen from the five models is used for testing, and the rest four are used for training. The image of each subject model is used for testing in turn. Namely, we made use of five fold cross validation strategy.

4.1 Background Model and Foreground Subject Extraction

A background model is used for segmenting the foreground subject or object. If the background model is affected by the illumination change, there will be some noise or wrong segmented region left in the extracted image. From our experience, the frame ratio method can eliminate the influence of illumination variations.

A threshold k is applied in frame ratio approach to obtain binary image $B(x, y)$ by Eq. (2.5) of Section 2.1.2. The threshold value of k is chosen by experiment and varies with different environment. Hence, we ran a series of experiments to determine the optimal threshold k and the corresponded binary images are shown in Fig. 4.2. The threshold value $k = 1.3$ was adopted in our experiment.

Foreground subject region is extracted from binary image $B(x, y)$ in order to minimize the size of images. Foreground region extraction is accomplished by simply taking a threshold along X and Y directions. Fig. 4.3 shows an example of foreground region extraction. Fig. 4.3(a) is an image frame of the video stream. Fig. 4.3(b) is the binary image after performing background model analysis. Figs. 4.3(c) and 4.3(d) show the projection of Fig. 4.3(b) onto the X and Y directions, respectively. We can find the boundary coordinates of X and Y directions by observing the projection histogram. We used these boundary coordinates to define a rectangle to extract foreground region from Fig. 4.3(b). Fig. 4.3(e) is the extracted foreground region.
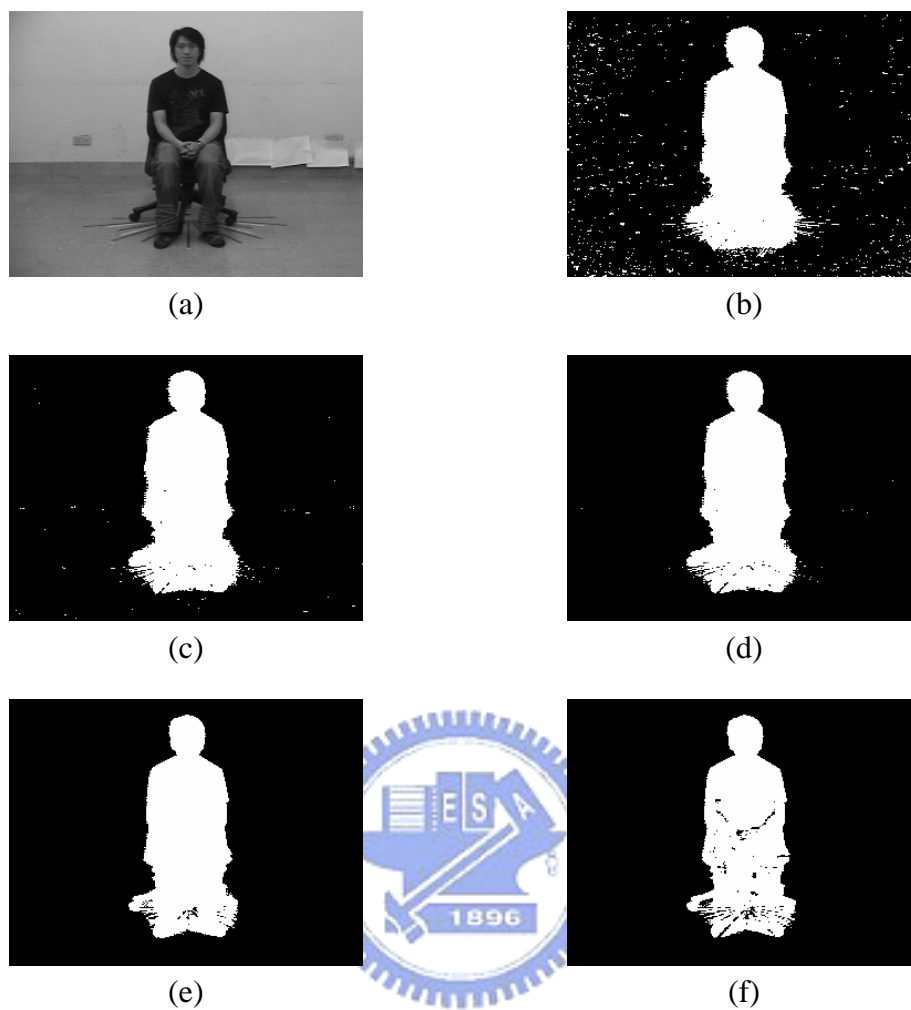
Fig. 4.2　An example of foreground region extraction at different threshold, *k*, values. (a) An image frame, (b) k = 1.0, (c) k = 1.1, (d) k = 1.2, (e) k = 1.3, and (f) k = 1.4.
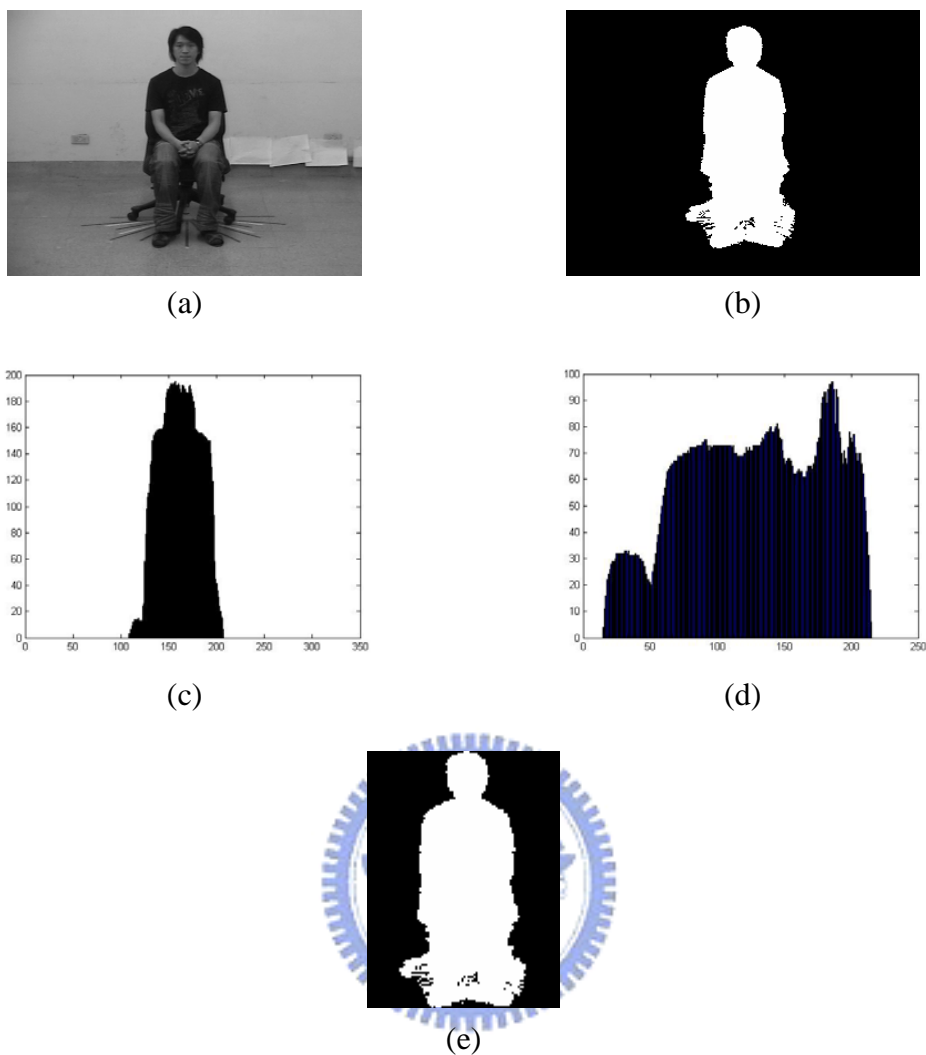
Fig. 4.3 An example of foreground region extraction. (a) An image frame. (b) Binary image after background analysis. (c) Projection of (b) onto X direction. (d) Projection of (b) onto Y direction. (e) Foreground region extracted.

4.2   Subject Direction Detection

4.2.1   Subject Contour Extraction

For subject direction detection, we use the Fourier descriptor analysis to detect subject direction. We choose 19 essential templates which a picture is taken from the front ranging from $90°$ to $-90°$. There are totally 19 essential templates, and comprising 19 classes. In order to do Fourier descriptor analysis, we have to extract the contour of the subject from foreground image. Fig. 4.4 shows the foreground images which include different directions. Fig. 4.5 shows the contour of the subjects extracted by edge detection.

Fig. 4.4   Representation a human sitting on a chair at the same location with different direction, binary image which include $-90°$, $-60°$, $-30°$, $-10°$, $10°$, $30°$, $60°$, and $90°$.
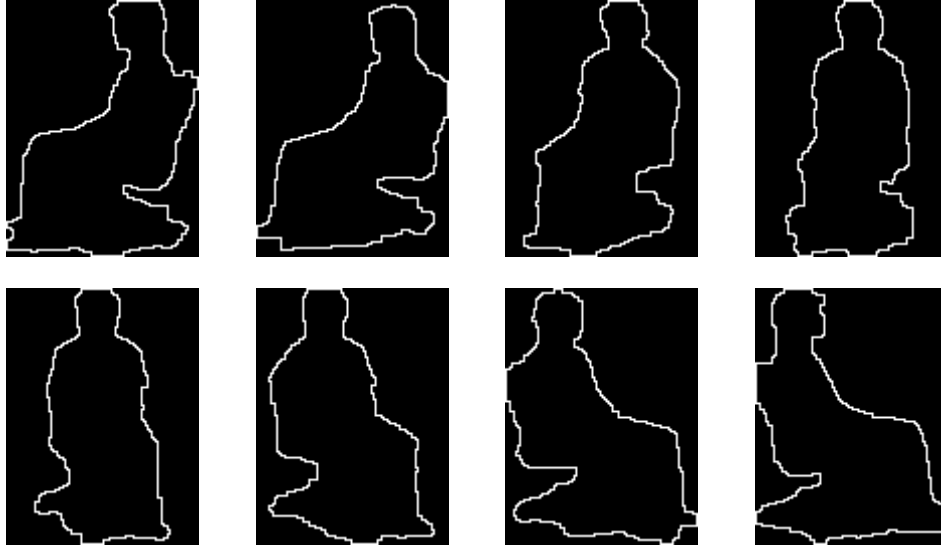
Fig. 4.5   Representation a human silhouette extracted from Fig. 4.4.

4.2.2   Detection Result Using FD and LDA

In Chapter 2, we discussed the Fourier coefficients $\{a_n, b_n\}$ of a polygonal curve. We can get the coefficients from the following formula.

$$\mu_0 = -\pi + \frac{1}{L} \sum_{k=1}^{m} l_k \Delta\phi_k, \tag{4.1}$$

$$a_n = \frac{-1}{n\pi} \sum_{k=1}^{m} \Delta\phi_k \sin\frac{2\pi n l_k}{L}, \tag{4.2}$$

$$b_n = \frac{1}{n\pi} \sum_{k=1}^{m} \Delta\phi_k \cos\frac{2\pi n l_k}{L}, \tag{4.3}$$

where

$$l_k = \sum_{i=1}^{k} \Delta l_i \quad , \qquad L = \sum_{i=1}^{m} \Delta l_i ,$$

42

The Fourier descriptor $\{A_n, \alpha_n\}$ can be obtained from the equations above, and f is the preceding term number of Fourier descriptors. We can obtain the Fourier descriptor as following.

$$A_n = \sqrt{a_n^2 + b_n^2},\qquad(4.4)$$

$$\alpha_n = \pi + a\tan 2(b_n, a_n) = \begin{cases} \pi + a\tan\dfrac{b_n}{a_n}, & \text{if } a_n > 0 \\[2mm] (2 + \text{sign } b_n) \cdot \dfrac{\pi}{2}, & \text{if } a_n = 0 \qquad(4.5) \\[2mm] (1 + \text{sign } b_n) \cdot \pi + a\tan\dfrac{b_n}{a_n}, & \text{if } a_n < 0 \end{cases}$$

$$n = 1, 2, 3, \cdots, f.$$

In the following, we summarize the steps to compute Fourier descriptors.

● Step 1: Firstly, define the starting point of the curve, and clockwise around the contour and recode the image boundary coordinates using chain code.

● Step 2: Base on the coordinate sequence, calculate the δ_0 of the starting point.

$$\delta_0 = \begin{cases} \dfrac{\pi}{4}(j_0 - s + 2) \cdot \text{sign}(i_0 - r), & \text{if } r \neq i_0 \\[2mm] \dfrac{\pi}{4}(j_0 - s + 2) + \text{sign}(i_0 - s), & \text{if } r \neq i_0 \end{cases}\qquad(4.6)$$

where coefficients (i_0, j_0) is the coordinates of starting point, and coefficients (r, s) is the coordinates of next point. The coefficient δ_0 represent the orientation of the starting point of the curve.

● Step 3: Starting from the starting point, calculate the a_p^* and Δl_p.

43

$$a_p^* = \begin{cases} 2 + [2(r + 1 - i) + s - j] \cdot \text{sign}(r - i), & \text{if} \quad \text{sign}(r - i) \neq 0 \\ 2 + [1 + \text{sign}(j - s)], & \text{if} \quad \text{sign}(r - i) = 0 \end{cases} \quad (4.7)$$

$$\Delta l_p = 1 + \frac{\sqrt{2} - 1}{2} [1 - (-1)^{a_p^*}], \qquad p = 1, 2, \cdots, v \qquad (4.8)$$

where Δl_p is the distance between two points which are neighbors.

● Step 4: Calculate the angle $\Delta\phi_{p-1}$ between two edges, and calculate the $\Delta\phi_v$ at last.

$$\Delta\phi_p = \begin{cases} \dfrac{\pi}{4} (a_p^* - a_{p-1}^*) & \text{if} \quad |a_p^* - a_{p-1}^*| < 4 \\ -\pi & \text{if} \quad |a_p^* - a_{p-1}^*| = 4 \\ \dfrac{\pi}{4} [a_p^* - a_{p-1}^* - 8\text{sign}(a_p^* - a_{p-1}^*)] & \text{if} \quad |a_p^* - a_{p-1}^*| > 4 \end{cases} \quad (4.9)$$

$$p = 2, 3, \cdots, v$$

$$\Delta\phi_v = \begin{cases} \dfrac{\pi}{4} (a_1^* - a_v^*) & \text{if} \quad |a_1^* - a_v^*| < 4 \\ -\pi & \text{if} \quad |a_1^* - a_v^*| = 4 \\ \dfrac{\pi}{4} [a_1^* - a_v^* - 8\text{sign}(a_1^* - a_v^*)] & \text{if} \quad |a_1^* - a_v^*| > 4 \end{cases} \quad (4.10)$$
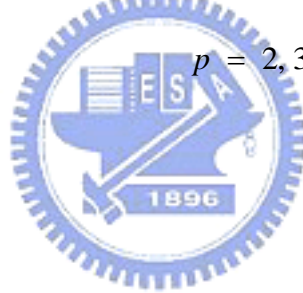
● Step 5: Calculate the i_0 and l_k, $k = 1, 2, \cdots, v$.

● Step 6: Decide the number of terms of Fourier descriptors f.

● Step 7: Calculate a_n and b_n, $n = 1, 2, \cdots, f$.

● Step 8: Compute A_n and α_n from a_n and b_n.

After these manipulations, we can obtain Fourier descriptors for detecting the subject direction. In the training phase, there were 5 models, each model provides 19 directions which a picture is taken from the front ranging from $90°$ to $-90°$. Using five-fold cross-validation, there were four training subject models to be trained in turn.

The descriptors are calculated by using Eqs. (2.23-25). Each direction of subject will be represented by a group of descriptors which describe the silhouette of the subject. We extracted the 20 preceding descriptors which contain more macroscopic information, and calculate the average of the descriptors of the same term from equivalent direction of the four training models. Therefore, we gathered nineteen groups of descriptors from different direction of subject for constructing data base.

In the testing phase, there were five test subject models to be tested in turn. Also, the image of the testing model is converted to the coefficients using Fourier descriptors. Then, we calculate the Euclidean distances between the testing image and the nineteen direction data base. The angels of the least two Euclidean distance models could represent the possible direction range of the subject. Consequently, the subject's direction could be estimated by the linear interpolation on these two direction angles. Table III shows that the relationship between desired angles and estimated results of subject direction detection using only Fourier descriptor method.
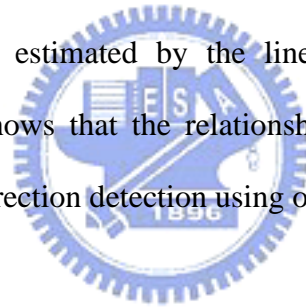
TABLE III

THE ESTIMATED SUBJECT DIRECTION BY FD ONLY

angel[°]	Model 1	Model 2	Model 3	Model 4	Model 5
	α [°]	α [°]	α [°]	α [°]	α [°]
90	75.91	84.58	75.39	75.1	85.01
80	75.47	84.22	65.01	75.43	84.55
70	73.75	75.1	64.59	64.88	75.03
60	64.85	55.27	54.85	55.45	74.83
50	54.24	45.94	44.56	54.45	65.01
40	44.06	43.54	34.54	44.52	34.59
30	37.36	34.52	25.49	34.82	15.15
20	23.55	16.58	23.85	15.95	5.36
10	14.07	6.58	14.27	13.99	4.49
0	2.95	-4.12	4.62	-5.22	-5.08
-10	-5.14	-14.38	-4.88	-24.8	-6.613
-20	-14.61	-24.35	-14.85	-33.21	-14.54
-30	-25.25	-34.41	-15.91	-35	-16.06
-40	-64.31	-45.05	-44.63	-64.9	-44.53
-50	-64.45	-64.9	-45.26	-65.33	-44.35
-60	-65.34	-64.8	-54.21	-65.28	-65.54
-70	-66.56	-65.3	-64.97	-65.41	-74.28
-80	-75.02	-65.57	-85.54	-74.5	-74.96
-90	-84.55	-65.74	-86.63	-74.34	-85.26

In order to improve the classification performance, after we constructed the data base of the subject directions, we do LDA to maximize between-class and in the same time minimize within-class variations. Combine the FD and the LDA methods, we can obtain better performance. Table IV shows the result of subject direction detection using combine FD and LDA.

TABLE IV

THE ESTIMATED SUBJECT DIRECTION BY FD AND LDA

angel[°]	Model 1	Model 2	Model 3	Model 4	Model 5
	α [°]	α [°]	α [°]	α [°]	α [°]
90	85	83.56	85.30	83.61	86
80	74.65	76.33	84.53	82.98	75.9
70	65.73	74.4	65.67	75.61	74.12
60	55.13	63.62	64.24	65.10	56.48
50	53.43	46.24	54.35	54.80	54.07
40	34.18	37.24	43.86	43.61	43.47
30	26.05	26.3	26.31	34.45	26.26
20	15.16	16.2	23.70	23.84	15.97
10	5.74	5.98	6.41	13.57	5.97
0	-3.99	-3.02	-4.25	-3.33	-4.43
-10	-13.05	-5.97	-5.80	-16.78	-6.55
-20	-16.87	-15.87	-15.2	-26.43	-16.67
-30	-24.79	-26.68	-33.51	-35.88	-34.20
-40	-35.61	-43.59	-43.76	-44.37	-44.60
-50	-44.01	-54.02	-55.08	-55.17	-53.44
-60	-65.12	-63.9	-63.6	-63.89	-63.96
-70	-74.82	-65.76	-65.67	-85.81	-65.91
-80	-84.69	-85.92	-73.13	-83.99	-71.88
-90	-84.55	-86.78	-84.30	-85.20	-76.28

To evaluate the estimation performance, we calculated the mean absolute error (MAE) of estimated angle of each model. Furthermore, we can estimate the average error of these five models. When subject direction detection using FD only, the average error is $7.08°$. However, we combine the FD and the LDA methods, it can reduce 35% of error to $4.58°$. Hence, combine the FD and the LDA have better performance then FD only. Table V shows the comparison of the MAE and average error of different method.

TABLE V

COMPARISON OF MAE OF DIFFERENT METHODS

Method	Model 1	Model 2	Model 3	Model 4	Model 5	Average
FD only	$6.50°$	$6.52°$	$6.42°$	$8.45°$	$7.50°$	$7.08°$
FD+LDA	$4.61°$	$4.37°$	$4.37°$	$5.31°$	$4.65°$	$4.58°$

4.3 Face Tracking by PTZ Camera

In our research, we using Pan-Tilt-Zoom (PTZ) camera to track human face before face direction detection. In order to estimate face direction correctly, we have to zoom the face of target person in to occupy the image between 70% to 80% vertically. We utilize YC_bC_r skin color segmentation method to detect the face region initially. There will be some noise or other skin color region (e.g. hand, leg, etc.) segmented left in the extracted image. We have to eliminate noise and non-face region. Then calculate the coordinates of the center of the face region, which are used to compute the distance to the center of image. We recode the pixels of x-direction and y-direction from the distance which is proportionality to the camera pan and tilt time, and compute the occupying ratio of the human face in the image. According to these parameters, PTZ camera will track human face automatically. The flow chart of PTZ camera control is shown in Fig.4.6, with two examples of facial tracking shown in Figs. 4.7 and 4.8.
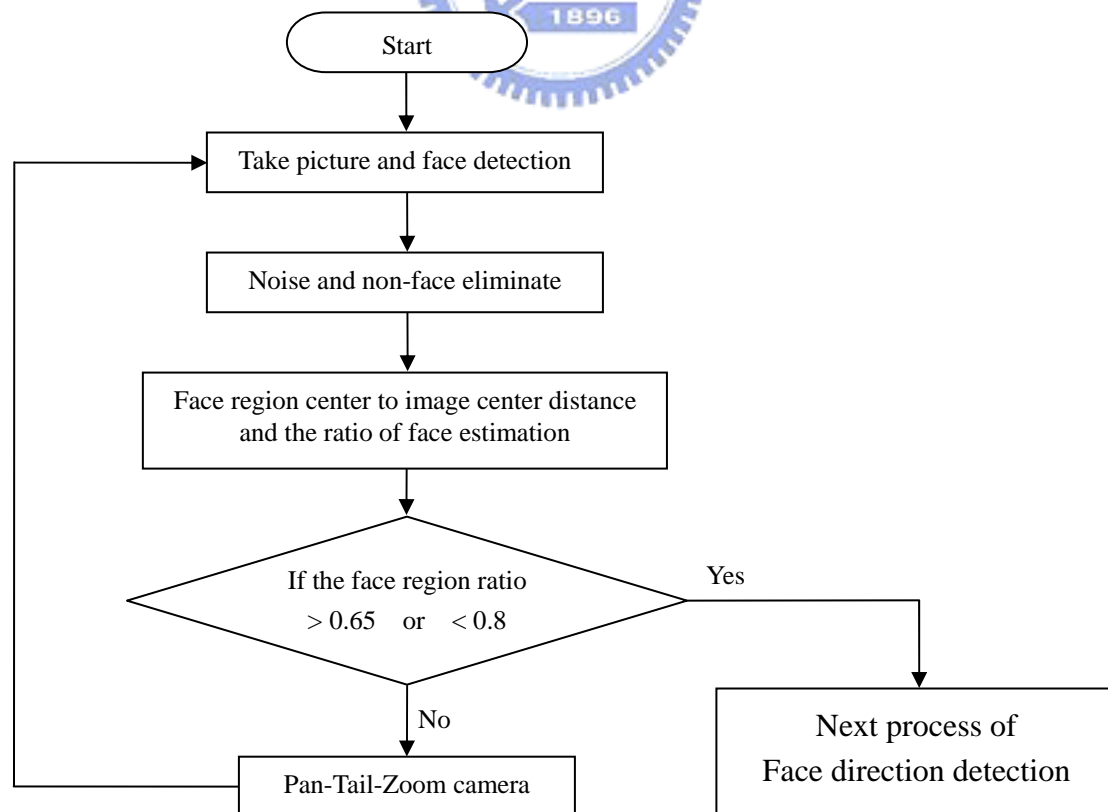
Fig. 4.6   The PTZ camera control for face tracking.

Fig. 4.7   The first example of face tracking process by PTZ camera. (a)–(d) Input

image. (a1)–(d1) Face detection by YC_bC_r skin color segmentation

method. (a2)–(d2) The result of face tracking process.

(a)   (a1)   (a2)

(b)   (b1)   (b2)

(c)   (c1)   (c2)

(d)   (d1)   (d2)
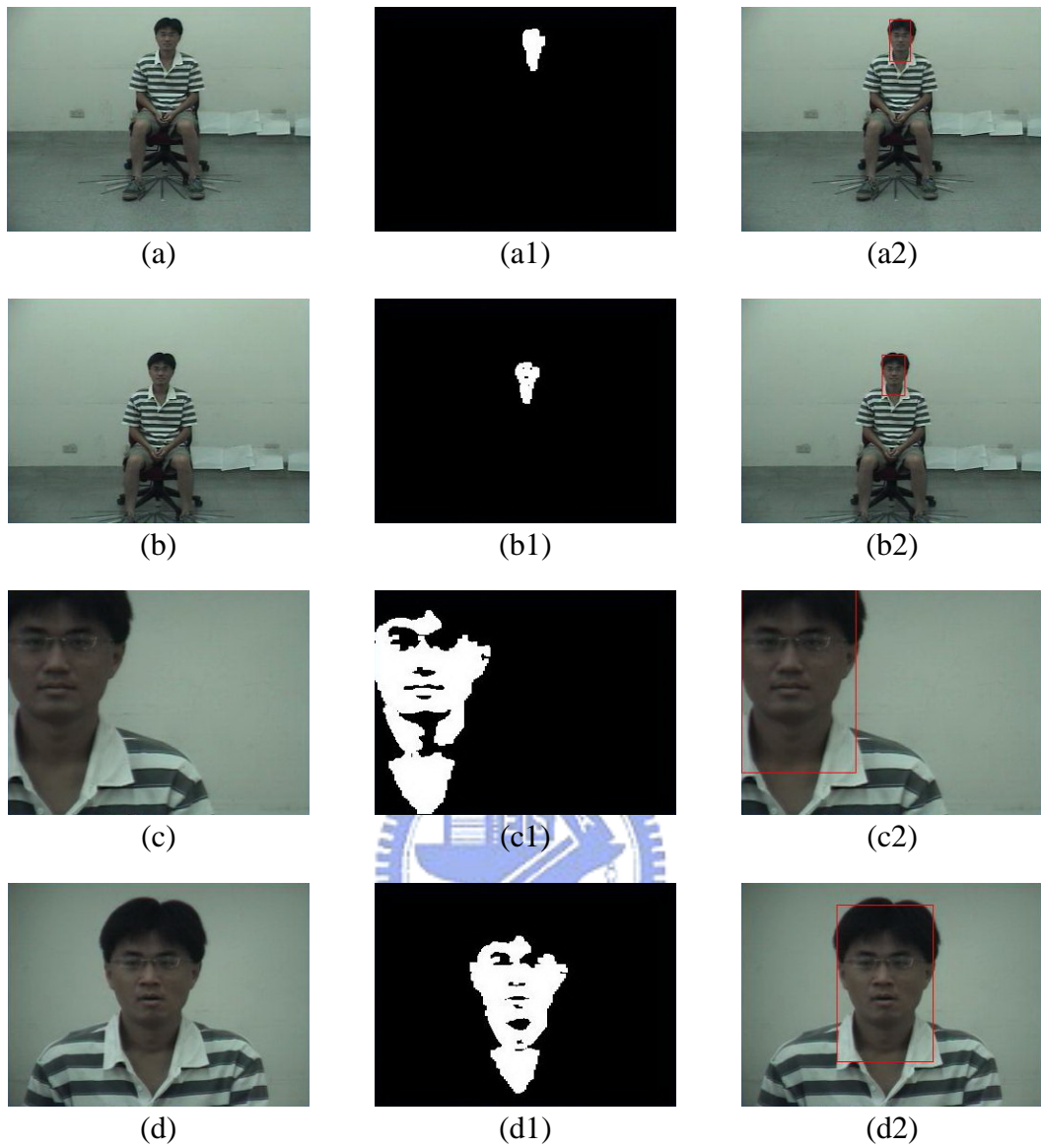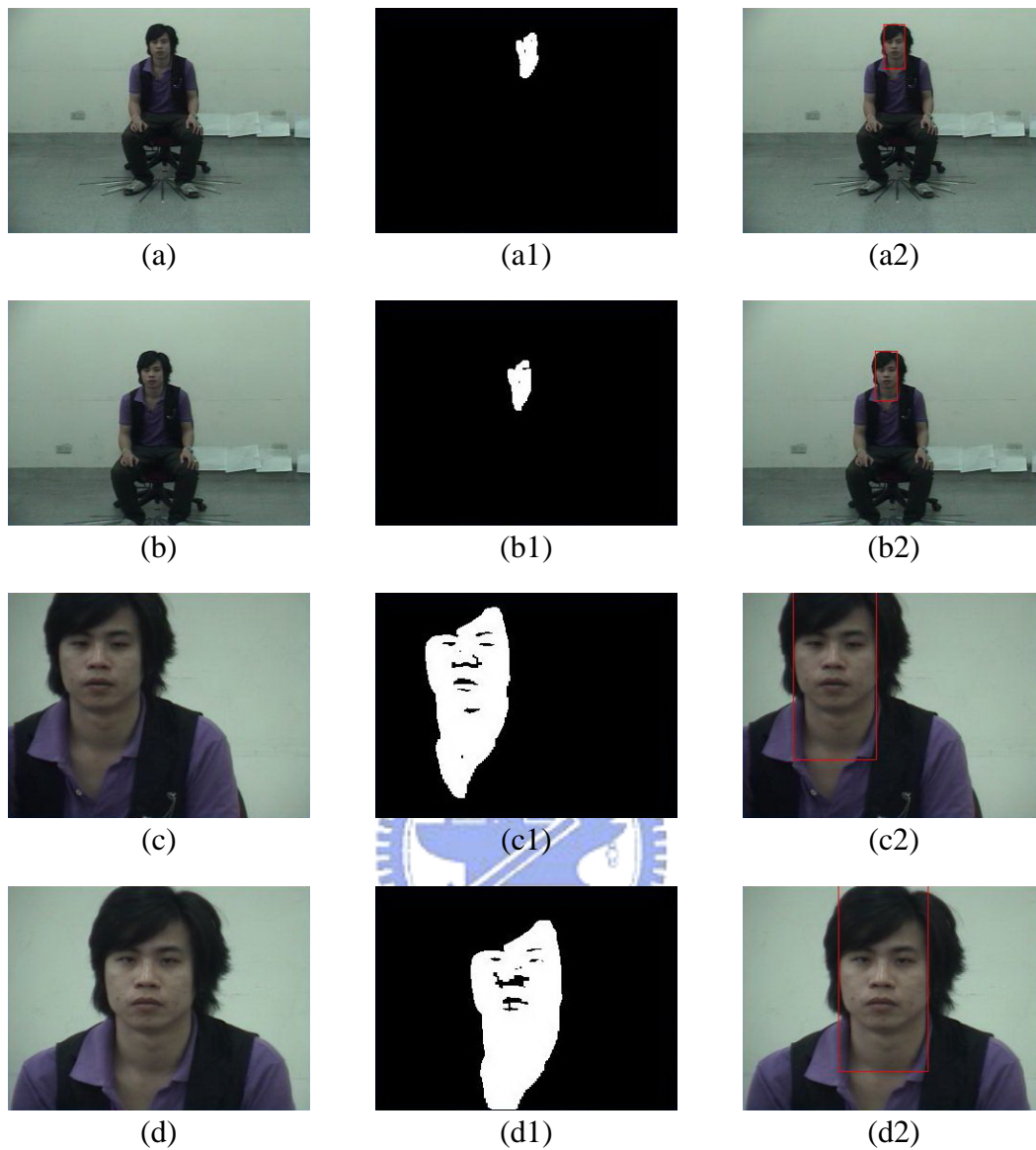
Fig. 4.8   The second example of face tracking process by PTZ camera. (a)–(d) Input image. (a1)–(d1) Face detection by YC_bC_r skin color segmentation method. (a2)–(d2) The result of face tracking process.

4.4   Face Direction Detection

4.4.1   Head Region Extraction

We utilize both YC_bC_r skin color detection and sobel edge detection to extracted head region. Firstly, we do the skin color segmentation by Eq. (3.1-3) of Sections 3.1 and 3.2 to extract the face region. Then, we utilize the edge detection method to extract the edge of head. At last, we can combine the above two results to locate the face region. This facial region however contains some regions of no interests. Thus, we can remove the lower region that contains neck, shoulders, etc. Consequently, we can extract the head region for face direction detection. In the following, we show two examples of head region extraction on the images of bare faces in Figs. 4.9 and 4.10.

Fig. 4.9   The example 1 of head region extraction. (a)–(e) The input image. (a1)–(e1)
Union of the results of YC_bC_r skin color detection and sobel edge
detection. (a2)–(e2) The result of head region extraction.

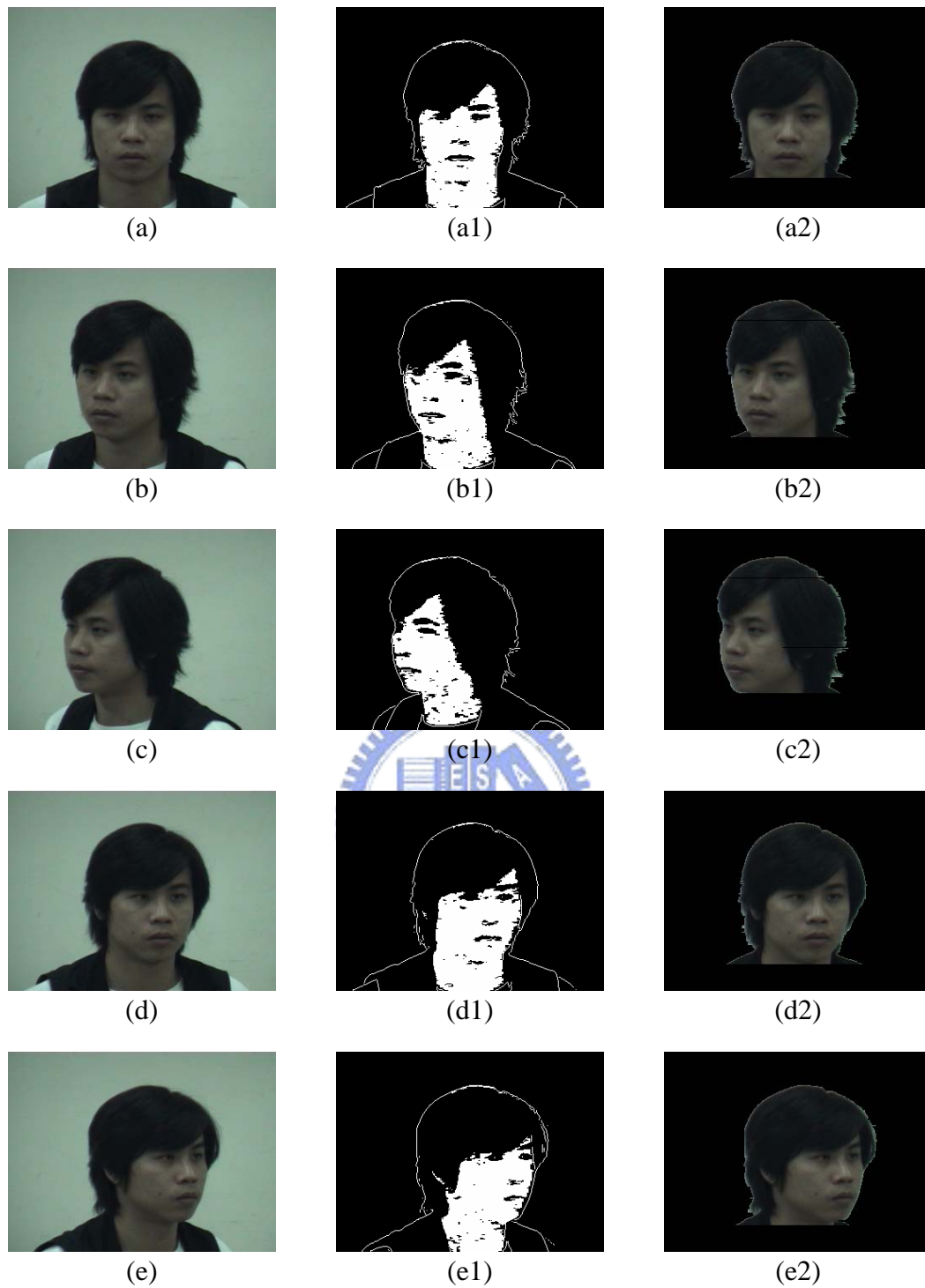Fig. 4.10 The example 2 of head region extraction. (a)–(e) The input image. (a1)–(e1) Union of the results of YC_bC_r skin color detection and sobel edge detection. (a2)–(e2) The result of head region extraction.

4.4.2 Eye Detection by PCA

To train a robust eye detector, we have collected training data from various sources. Ten images of eye are collected randomly from our members as the training data. In application, both right and left eyes were trained due to detect various eyes, and the training eyes were randomly rotated with small angles. The eigeneyes computation was described in Section 3.2.2. The training images from Fig. 4.11 were taken from research members, and the image resolution is 30×15 pixels.
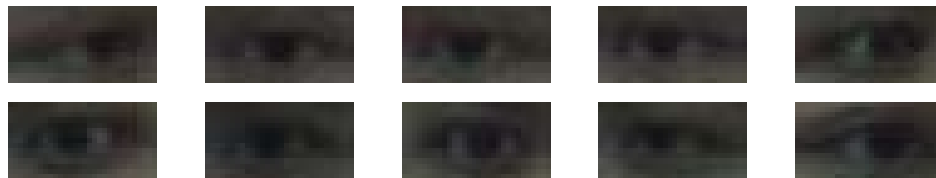
Fig. 4.11   Some eye images used in training.

Since the eye position probably locates within the region of 2/5 to 3/5 of the height of the head, we extract only this region for further processing to improve the eye detection speed. The extraction region is shown in Fig. 4.12. In the detecting phase, we choose the preceding five eigenvalues and their corresponding eigenvectors as the eigenspace transformation matrix. There were five test subject models to be detected in turn. The 30×15 image patch demonstrates small Euclidean distances between the scan region and the eye models when we scan the path to be near the eye. If the distance is smaller than the threshold, Th=500, we will thus identify it to be the potential position of the eye region. However, these patches would contain two clusters which represent the right and the left eyes. Therefore, we utilize threshold clustering algorithm to classify two clusters. Then, we find the minimum distance each of these two clusters, which constitute the left and right eyes. The eye detection result is shown in Fig. 4.13.
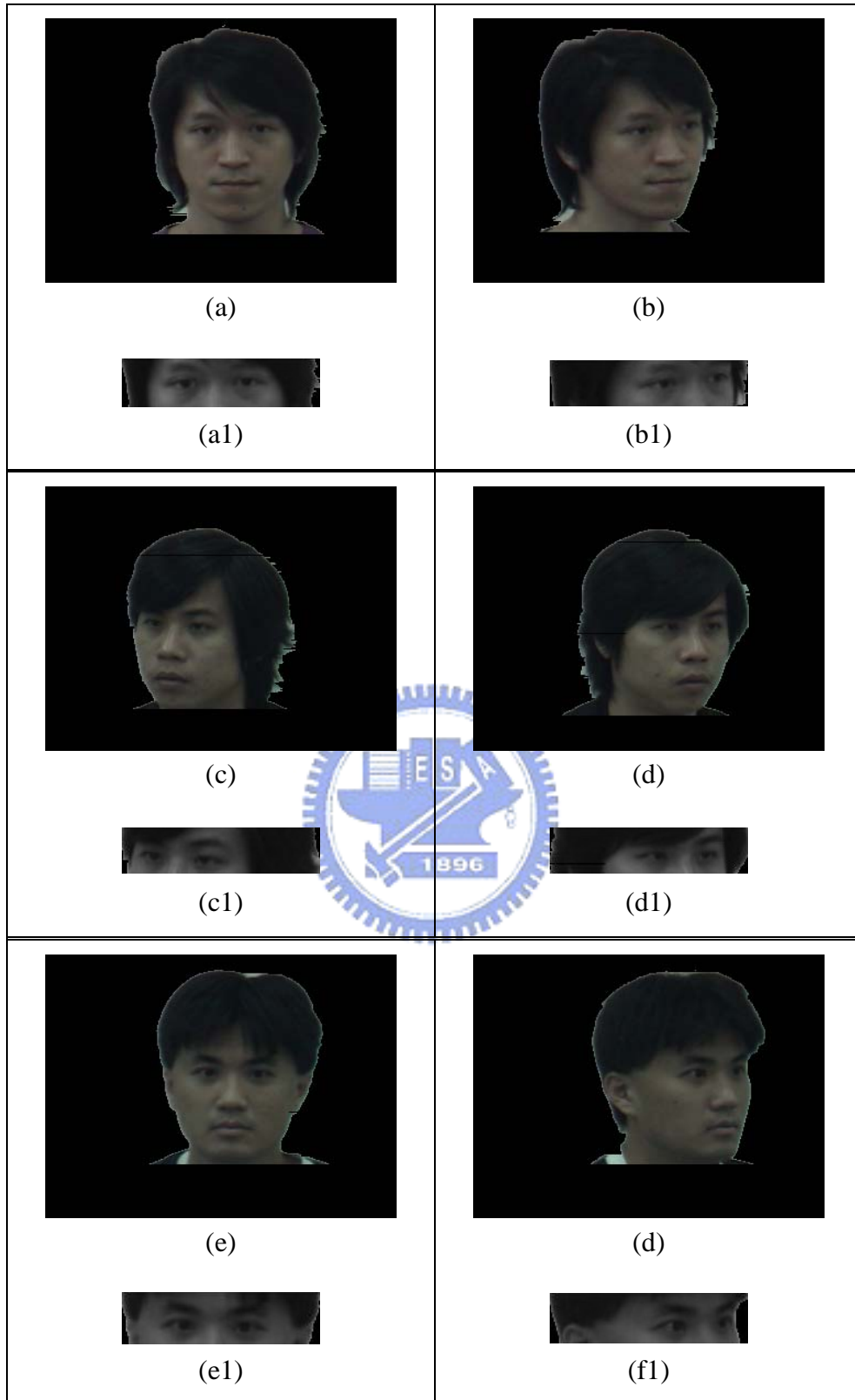
Fig. 4.12 Some examples of the extraction region for eye detection. (a)–(e) The head region. (a1)–(e1) The extraction region from head region (a)–(e).
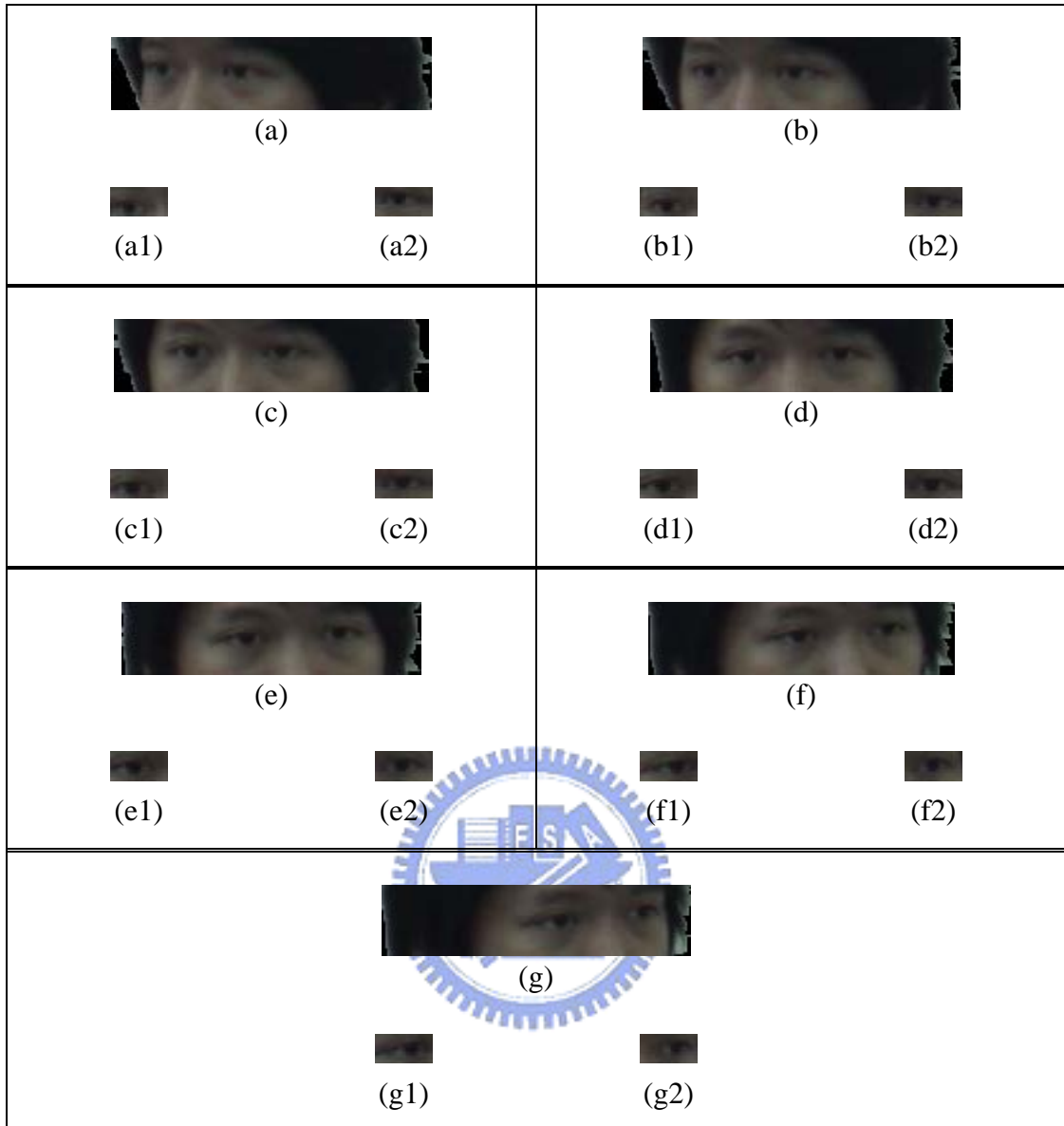
Fig. 4.13 The eye region extraction using eye detection by PCA which the model facing from -30° to 30°, 10° an interval. (a)–(g) The extraction region for eye detection. (a1)–(g1) The right eye region. (a2)–(g2) The left eye region.

4.4.3 Face Direction Detection Result

To detect face direction, we need the horizontal coordinate of the nose position which is located close to the middle of the two eyes. Firstly, we have to detect the pupil form the eye region detected by PCA. Our pupil detection method is based on gray intensity field in Eq. (3.12) described in Sections 3.2.5. Fig. 4.14 shows some examples of using a mask to locate the iris centers. Moreover, as shown in Fig. 4.15, the horizontal line segment passing through the middle of the two pupils and starting from the left side and ending on the right side specifies the R, L parameters of Eq. (3.13) for face direction detection.

(a)　　　　　　(b)　　　　　　(c)

Fig. 4.14　Different face direction of iris circling. (a) Input image of eye region. (b) The intensity field. (c) Iris center located.

After we computed the coordinates of the center of the two pupils, we can draw a horizontal line through two pupils. Therefore, we can compute the coordinates on the right and the left side of head. According to these parameters of coordinates, we can utilize Eq. (3.13) to estimate the face direction. Some examples of face direction estimation are shown in Fig. 4.15.
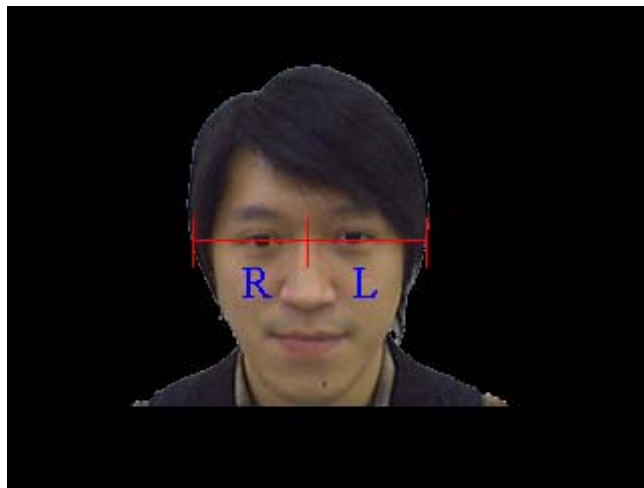
Left eye position：126.7
Right eye position：172.8
Nose position：149.75
Head left position：92.7
Head right position：208.8
Head width：116.1
R=57.05
L=59.05

$\psi = 0.987^{\circ}$

(a)
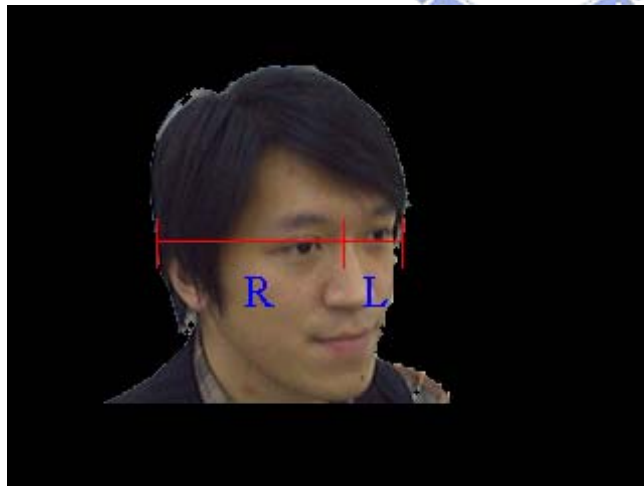
Left eye position：147.6
Right eye position：184.7
Nose position：166.15
Head left position：74.7
Head right position：197.6
Head width：122.9
R=91.45
L=31.45

$\psi = 29.22^{\circ}$

(b)

Fig. 4.15   The example of face direction estimation using proposed method. (a) The first example of facing 0°. (b) The second example facing 30°.

Table VI shows the result of face direction detection by the pupil locations.

TABLE VI

THE FACE DIRECTION BY THE PUPIL LOCATIONS

angel[°]	Model 1	Model 2	Model 3	Model 4	Model 5
	ψ [°]	ψ [°]	ψ [°]	ψ [°]	ψ [°]
60	12.33	54.23	36.57	27.54	-6.04
50	12.32	12.85	26-87	33.65	39.45
40	33.98	37.56	36-75	37.15	33.12
30	25.43	27.95	30.54	28.68	26.88
20	15.75	18.38	19.64	20.38	19.03
10	8	12.17	9.44	13.22	8.35
0	-2.86	-2.45	3.71	4.55	0.66
-10	-12.88	-15.97	-10.24	-15.65	-12.73
-20	-18.39	-22.84	-20.92	-25.14	-19.79
-30	-26.72	-30	-31.61	-33.54	-28.09
-40	-39.84	-36.78	-41.81	-29.78	-36.57
-50	-34.25	-31.69	-32.23	-23.17	-30.17
-60	-35.3	-24.86	-15.68	-24.7	-45.2

To evaluate the estimation performance, we calculated the MAE of estimated angle from equivalent direction of each model. We compared the direction estimation results of the methods which use subject direction detection and face direction detection. Consequently, when the human direction angle (or face direction angle) is within $\pm 30°$, the face is more accurately than that from the subject's silhouette. On the other hand, the face accuracy will decrease quickly when the angle is larger than $\pm 40°$. Therefore, we can further fuse the face direction detection if the face is

available and within $\pm 30^{\circ}$. Table VII shows the comparison of the MAE of different methods which estimated from face and silhouette, ranging from 50° to -50°.

TABLE VII

COMPARISON OF MAE OF DIFFERENT METHODS

Method	0°	$\pm 10^{\circ}$	$\pm 20^{\circ}$	$\pm 30^{\circ}$	$\pm 40^{\circ}$	$\pm 50^{\circ}$
Face	2.85°	2.71°	1.83°	2.20°	4.03°	22.33°
Silhouette	3.80°	4.10°	4.20°	4.16°	4.02°	4.41°

4.5 Combine the results of Subject and Face Direction Detection

From Tables VI and VII, we can see the results of face direction detection. This result shows that the face direction detection is not suitable for large face angle because the eyes will be occluded for face angle lager than $\pm 40°$. Therefore, we concluded the face direction detection is meaningful in facial angle below $\pm 30°$. Moreover, comparing Tables IV and VI, we can see the face and silhouette direction detection is performs almost equal when the face angle between $30°$ and $40°$. Hence, we use the weights, w_1 and w_2, are constrained, Fig. 4.16, by the following relationship.
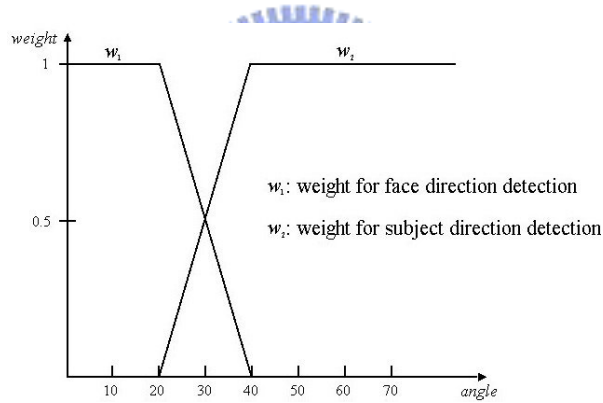

Fig. 4.16   Weight graph of detection reliability.

The subject direction θ is estimated by using the weights and a linear interpolation method as follows.

$$\theta = w_1\psi + w_2\alpha, \tag{4.11}$$

where ψ and α are the estimated face and subject direction angles, respectively.

62

Table VIII is the estimated results by the proposed method which combine the results of subject and face direction detection. We can see that the whole subject direction angle is well estimated from the proposed linear combination scheme.

TABLE VIII

THE ESTIMATED SUBJECT DIRECTION BY THE PROPOSED METHOD

angel[°]	Model 1 θ[°]	Model 2 θ[°]	Model 3 θ[°]	Model 4 θ[°]	Model 5 θ[°]
90	85	83.56	85.30	83.61	86
80	74.65	76.33	84.53	82.98	75.9
70	65.73	74.4	65.67	75.61	74.12
60	55.13	63.62	64.24	65.10	56.48
50	53.43	46.24	54.35	54.80	54.07
40	34.18	37.24	43.86	43.61	43.47
30	25.74	27.13	28.42	31.56	26.57
20	15.76	18.38	19.64	20.38	19.03
10	8	12.17	9.44	13.22	8.35
0	-2.86	-2.45	3.71	4.55	0.66
-10	-12.88	-15.97	-10.24	-15.65	-12.73
-20	-18.39	-22.84	-20.92	-25.14	-19.79
-30	-25.75	-28.34	-32.56	-34.7	-31.14
-40	-35.61	-43.59	-43.76	-44.37	-44.60
-50	-44.01	-54.02	-55.08	-55.17	-53.44
-60	-65.12	-63.9	-63.6	-63.89	-63.96
-70	-74.82	-65.76	-65.67	-85.81	-65.91
-80	-84.69	-85.92	-73.13	-83.99	-71.88
-90	-84.55	-86.78	-84.30	-85.20	-76.28

To evaluate the estimation performance, we calculated the mean absolute error (MAE) of estimated angle of each model. Furthermore, we can estimate the average error of these five models. When subject direction detection using silhouette only, the average error is $4.58°$. However, we fuse the face and subject direction detection, it can reduce 13% of error to $3.99°$. As shown in Table IX, linear combination approach produces better performance than that from subject's silhouette only. Table IX shows the comparison of the MAE and average error of method 1: using only subject direction detection, and method 2: combine subject and face direction detection.

TABLE IX

MAE COMPARISON OF DIFFERENT METHODS

Method	Model 1	Model 2	Model 3	Model 4	Model 5	Average
Silhouette only	$4.61°$	$4.37°$	$4.37°$	$5.31°$	$4.65°$	$4.58°$
Combining face & silhouette	$4.28°$	$3.64°$	$3.43°$	$4.87°$	$3.79°$	$3.99°$

Chapter 5 Conclusion and Future Work

In this thesis, we have proposed an approach which combine subject's silhouette matching and face direction detection for the human direction detection. In our direction detection system, we first utilize Fourier descriptor to discriminate different directions of target subject ranging from $90°$ to $-90°$. In addition, we utilize the linear discriminant analysis to optimize the class separability and improve the classification performance. Moreover, we also estimate the face direction from one's pupils and pupils' geometric relationship as well. When the human direction angle (or face direction angle) is within $\pm 30°$, direction detection from the face is more accurately than that from subject's silhouette, and decrease quickly when the angle exceeds $\pm 40°$. Therefore, we fuse to detect direction if face detection reliable. The experiment results have shown that our approach can obtain a high accuracy on subject direction detection, with mean absolute error less than $4°$.

For future work, we shall increase the detection from the back subject's side ranging from $90°$ to $-90°$. Therefore, we can complete the subject direction detection for all views. On the other hand, the distance of the subject is also important to be estimated, which is the subject of future research.

References

[1] S. Nayar, S. Rene, and H. Murase, "Realtime 100 object recognition system," in *Proceedings 1996 IEEE International Conference on Robotics and Automation,* pages 2321–2325, 1996.

[2] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models: Their training and application," *CVIU*, 61(1):38–59, January 1995.

[3] J. J. Koenderink and A. J. van Doom, "The singularilarities of the visual mapping," *Biol. Cyber.,* 24:51–59, 1976.

[4] J. J. Koenderink and A. J. van Doom, "The internal representation of solid shape with respect to vision," *Biol. Cyber.,* 32:211–216, 1979.

[5] C. H. Chien and J. K. Agganval, "Volume/surface octrees for the representation of 3-D objects," *Comput. Vision, Graphics, Image Processing,* vol. 36, pp. 100–113, 1986.

[6] M. Piccardi, "Background subtraction techniques: a review," in *Proc. IEEE Int. Conf. SMC.*, vol. 4, pp. 3099–3104, 2004.

[7] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 809–830, August 2000.

[8] H. Freeman, "Boundary encoding and processing," in *Picture Processing and Psychopictoric*s, Lipkin and Rosenfeld, Eds. New York: Academic Press, 1970, pp. 241–306.

[9] T. Pavlidis and F. Ali, "Computer recognition of handwritten numerals by polygonal approximations," *IEEE Trans. Systems, Man, Cybern.,* vol. SMC-6, pp. 610–614, Nov. 1975.

[10] H. Blum, "A transformation for extracting new descriptors of shape," in *Models for the Perception of Speech and Visual Form,* W. Wathen-Dum, Ed. Cambridge, Mass.: MIT Press, 1967.

[11] O. Philbrick, "Shape description with the medial axis transformation," in *Pictorial Pattern Recognition,* G. C. Cheng, Ed. Washington, D.C.: Thompson, 1968, pp. 395–407.

[12] J. C. Moth-Smith, "Medial axis transformations," in *Picture Processing and Psychopictorics,* Lipkin and Rosenfeld, Eds. New York: Academic Press, 1970, pp. 267–283.

[13] C. T. Zahn and R. Z. Roskies, "Fourier descriptors for plane closed curves," *IEEE Trans. Computers,* vol. C-21, pp. 269–281, 1972.

[14] G. H. Granlund, "Fourier preprocessing for hand print character recognition," *IEEE Trans. Computers,* vol. C-21, pp. 195–201, 1972.

[15] E. L. Brill, "Character recognition via Fourier descriptors," presented at WESCON, Session 25, Qualitative Pattern Recognition Through Image Shaping, Los Angeles, Calif., Aug. 1968.

[16] R. L. Cosgriff, "Identification of shape," Ohio State University Research Foundation, Columbus, Report No. 820-11, ASTIA AD 254 792, 1960.

[17] K. Etemad and R. Chellappa, "Discriminant Analysis for Recognition of Human Face Images," in *First Int. Conf.,* AVBPA'97, Crans-Montana, Switzerland, March 1997, pp. 127–142.

[18] E. Hjelmas and B. K. Low, "Face detection: A survey," *Computer Vision and Image Understanding*, vol. 83, pp. 236–274, 2001.

[19] C. Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis," *IEEE Trans. Multimedia,* vol. 1, no. 3, pp. 264–277, September 1999.

[20] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, "Classifying facial actions," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 21, no. 10, pp. 974–989, October 1999.

[21] R. Feraud, O. J. Bernier, J. Viallet, and M. Collobert, "A fast and accurate face detector. based on neural networks," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 23, no. 1, January 2001.

[22] D. Maio and D. Maltoni, "Real-time face location on gray-scale static images," *Pattern Recognition*, vol. 33, pp. 1525–1539, 2000.

[23] H. Wu, Q. Chen, and M. Yachida, "Face detection from color images using a fuzzy pattern matching method," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 21, pp. 557–563, 1999.

[24] K. Sung and T. Poggio, "Example-based learning for view based human face detection. " *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 20, no. 1, 39–51, 1998.

[25] M. Yang and N. Ahuja, "Detecting human faces in color images." in *Proc. IEEE Conference on Image Processing*, Chicago, pp. 127–139, 1998.

[26] T. Kurozumi, Y. Shinza, Y. Kenmochi, and K. Kotani, "Facial individuality and expression analysis by eigenspace method based on class features or multiple discriminant analysis," in *Proc. International Conference on Image Processing*, vol. 1, pp. 648–652, 1999.

[27] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 19, pp. 743–756, 1997.