國立交通大學

資訊工程系

碩士論文

用於視覺監控系統的人臉可辨識度測量

Measurement of Face Recognizability for Visual Surveillance

研 究 生：曹育誠

指導教授：李錫堅　教授

中華民國九十三年七月

用於視覺監控系統的人臉可辨識度測量

Measurement of Face Recognizability for Visual Surveillance

研 究 生：曹育誠　　Student：Yu-Cheng Tsao

指導教授：李錫堅　　Advisor：His-Jian Lee

國 立 交 通 大 學
資 訊 工 程 系
碩 士 論 文

A Thesis

Submitted to Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer and Information Science

July 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年七月

用於視覺監控系統的人臉可辨識度測量

學生：曹育誠　　　　　　　　　　指導教授：李錫堅博士

國立交通大學資訊工程研究所碩士班

摘　　　要

本論文之研究目的在於提出一個評估人臉可辨識程度的機制。在一個監控環境內，監控系統可能需要辨識所偵測到的人臉，對於一張較難以辨識的人臉，例如側臉、或非正臉，通常會形成錯誤的辨識結果，導致辨識系統效能下降。如果能在人臉被辨識之前，先評估一個可正確被辨識的程度。對於辨識系統而言，可以藉由避免辨識低辨識度的人臉，減少不避要的工作量，並且能夠提升監控系統整體的效能。我們所提出的系統，分成為四個部份：前景偵測、頭部區域擷取、人臉特徵偵測、可辨識度計算。

第一個部份，先以一個統計式的方法來建立背景模組。這個背景模組在輸入一連串背景樣本後，分析背景影像內每個點的灰階值分佈，再計算其機率分佈。對輸入的影像，假設某個點的灰階值出現機率低於一定的門檻值，則代表這個點應該不屬於前景。最後找出所有前景點的連通單元，形成前景。

第二個部份，我們提出一個從前景頭部區域尋找橢圓的方法。因為頭部形狀比較接近於橢圓形，以橢圓為樣板擷取頭部影像，能更準確地擷取頭部影像。

第三個部份，尋找人臉特徵的位置，也就是眼睛和嘴巴的位置。我們希望能夠利用眼睛和嘴巴在人臉具有較強烈的方向特徵，依這個特徵來找尋眼睛和嘴巴的位置。首先，我們先從頭部區域依膚色灰階值來擷取人臉的區塊。再對人臉區塊計算方向統計圖，方向統計圖可以分析出區域的方向分佈。然後再由幾個較強的方向分佈來尋找人臉上往這些方向分佈的線段，這些線段有可能為眼睛、嘴巴所形成的線段。過濾其它的線段後，找出線段中心分佈密集的區域。再依據眼睛、

嘴巴的相對位置、顏色特徵來找驗證所找到的線段是否為眼睛和嘴巴所形成的。最後從找出的線段來推算出眼睛和嘴巴的位置。

第四個部份，以兩眼和嘴巴形成的三角形、方向的對稱度、區域灰階值的對稱程度，來區分人臉的轉向為三大類：正臉、有轉動的臉和非正臉。人臉的可辨識度以人臉大小，和兩眼和嘴巴的三角形和等腰三角形的相似和度為計算依據。

實驗的部份，測試可辨識度和實際辨識率之間的關係。實驗結果發現辨識率在可辨識度下降的情況，也會跟著下降。這代表我們所提出的可辨識度能夠有效提供辨識系統一個辨識度的參考。

Measurement of Face Recognizability for Visual Surveillance

Student: Yu-Cheng Tsao Advisors: Dr. Hsi-Jian Lee

Department of Computer Science and Information Engineering

National Chiao Tung University

Hsinchu, Taiwan 300

ABSTRACT

In this paper, measurement of face recognizability is proposed to evaluate the possible recognition degree of a face before face recognition. If we could measure the face recognizable degree after face detection, we can increase the efficiency by avoiding recognizing the face with power recognizabilities. The system consists of four stages: foreground segmentation, head extraction, facial component identification and recognizability measurement. In the first stage, based on a statistical background model, we apply the background subtraction approach to segment the foreground. After computing the probabilities for all pixels in the image, the connected pixels in the foreground would be collected as the foreground regions. In the second stage, we adopt an ellipse finding algorithm to extract the head region from the foreground regions. In the third stage, the system finds the facial components, eyes and mouth. Based on the features of the orientation distribution on the face regions, we collect the lines of the face with strong orientations. After line filtering, we could get the lines formed by the facial components. Lastly, the lines would be identified from the facial components based on their properties. The locations of the facial components could be evaluated from the locations of lines. In the forth stage, we use three features, that is, the triangle formed by eyes and mouth, the degree of the direction symmetry and the degree of the region symmetry to classify the facial orientations as three classes: frontal, oriented and non-frontal face. The recognizability of the face is then defined as the function of the face size and the relative locations of eyes and mouth. In our experiments, we tested the relation between the recognizability and the recognition rate. The experimental results show that the recognizability could be used as a measurement determines whether we will perform face recognition or not.

Acknowledgements

I heartily express my sincere gratitude to my advisor, Pro. His-Jain Lee, for his proper guidance and patient discussion throughout course of this study.

Many thanks are given to all colleagues in the Document Processing and Character Recognition Laboratory at National Chiao Tung University for their assistance on this thesis.

Finally, I would like to express my deep gratitude to my family and my friends for their help and encouragement in those working days. I would like to dedicate this thesis to them.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1 Motivation

In the modernized society, the security problem is more and more important with the rising of crime rate. Surveillance systems become essential equipments in many enterprises and organizations. Surveillance systems identify the person who enters the monitoring range all the time from recorded images. Traditionally, an administrator has to edit the film which contains the people and extract the faces without safe guards, to be recognized. Film editing would cost the administrators much time to look the monitor of surveillance systems. The system also needs much storage for recording possible. Vision-based surveillance systems are brought to avoid these disadvantages.

A vision-based surveillance system generally contains useful modules, like foreground segmentation, face extraction, and so on. Foreground segmentation module can segment the person from the recorded images. Face extraction module can locate the face regions from the segmented persons. These modules are helpful to reduce the workload of safe guards. However, the work of recognition is still ineffective. Not all detected faces are suitable to be recognized, because recognition systems could only recognize well the frontal view of faces. Faces with orientations or covers could get incorrect results. Therefore, the method that determines whether a face is appropriate for recognition systems is very useful.

1.2 Problem Definition

The problems that we want to solve in this study are listed as follows.

1.2.1 The Segmentation of foreground from a complex image

Foreground segmentation is a very important step. However, the surveillance environment could contain a complex background. To segment the foreground completely would be helpful to analyze.

1.2.2 Extract head component from a foreground

A head has variational orientations and human have many kinds of activity. It is difficult to extract the head from the foreground.

1.2.3. Estimate the recognizable degree under variational orientations

There are many factors that could affect the recognition result, for instance, orientation, light effect, changed facial expressions, and so on. For computing the correct recognizable probability, we must generalize all factors to define a measurement.

1.3 Survey of Related Research

1.3.1 Moving object detection

Background subtraction is a popular method for foreground segmentation, especially under those situations with a relatively stationary background. It attempts to detect moving regions in an image by differencing between the current image and a reference background image in a pixel-by-pixel manner. However, it is extremely sensitive to changes of dynamic scenes due to lighting and extraneous events. Yang and Levine [1] proposed an algorithm to construct the background primal sketch by taking the median value of the pixel color over a series of images based on the observation that the median value was more robust than the mean value. The median value, as well as a threshold value determined using a histogram-based procedure based on the least median squares method, was used to create the difference image. This algorithm proposed by Yang and Levine could handle some of the inconsistencies due to lighting changes, noise, and so on.

Some statistical methods to extract change regions from the background are inspired by the basic background subtraction methods described above. The statistical approaches use the characteristics of individual pixels or groups of pixels to construct more advanced background models. The statistics of the backgrounds can be updated dynamically during processing. Each pixel in the current image can be classified into foreground or background by comparing the statistics of the current background model. Stauffer and Grimson [2] presented an adaptive background mixture model for real-time tracking. In their work, they modeled each pixel as a mixture of Gaussians and used an online approximation to update it. The Gaussian distributions of the adaptive mixture models were evaluated to determine the pixels most likely from a background process, which resulted in a reliable, real-time outdoor tracker to deal with lighting changes and clutter.

Elgammal, et al. [3] present a non-parametric background model and a background subtraction approach. The background model can handle situations where the background of the scene is cluttered and not completely stationary but contains small motions such as tree branches and bushes. The model estimates the probability of observing pixel intensity values based on a sample of intensity values for each pixel. It could adapt quickly to changes in the scene which enables very sensitive detection of moving targets. The implementation of the model runs in real-time for both gray level and color imagery. Evaluation shows that this approach achieves very sensitive detection with very low false alarm rates.

1.3.2 Face detection in gray level images

Human face detection has a lot of applications such as video conferencing, multimedia, tracking, and personal identification. Many research studies had been proposed for detecting faces in these applications. These traditional methods almost concentrate on gray images and can be roughly classified into four categories [4]:

1. Knowledge-based top-down approach

In the first category, most proposed methods encode human knowledge of what constitutes a typical face. Usually, the rules capture the relationships between facial features. These methods are designed mainly for face location. Yang and Huang [5] proposed a hierarchical knowledge-based method to detect faces. Three levels of rules are employed in their system. At the first level, they locate all possible face candidates by globally searching the input image with a scanning window and applying a set of rules at each location. At the second level, these rules are detailed descriptions of what facial features look like. At the third level, another set of rules that correspond to eyes and mouth are applied to examine the remaining face candidates. This hierarchical mechanic is a coarse-to-fine strategy that is used to reduce the required computation. But this approach suffers from being difficult to implement the rules and it does not result in a high detection rate.

2. Feature-based bottom-up approach

In the second category, the proposed method tries to find invariant features of face for detection. The underlying assumption is based on the observation that must contain some features which are invariant over various variabilities such that human vision can detect faces effectively in different poses and lighting conditions. The commonly used facial features include eyebrows, eyes, nose, mouth, etc. One problem of these approaches is that the image features can be severely corrupted due to noise, occlusion, and illumination. Numerous methods have been proposed to detect facial features and identify face location based on detected features.

Zhou et al. [6] presents a framework of orientation analysis for rotated human face detection. An orientation histogram is constructed for statistical analysis of face images. The orientation's distribution of a region would be computed, and then a frontal face detector based on orientation analysis and other knowledge is designed to verify the given region.

Sirohey [7] proposed a localization method which used an edge map and heuristic to remove and group edges so that only the ones on the face contour are preserved.

Yow and Cippola [8-9] proposed a feature-based method which utilize a large amount of image evidence. Their algorithms are less sensitive to noise interference because of large amount of image evidence.

Augusteijn and Skufca [10] developed a method that infers the presence of a face through the identification of face-like textures. Dai and Nakano [11] also applied SGLD model to face detection. Color information is also incorporated with the face-texture model. Using the face texture model, they design a scanning scheme for face detection in color scenes in which the orange-like parts including the face areas are enhanced. One advantage of this approach is that it can detect faces which are not upright or have features such as beards and glasses. The reported detection rate is perfect for a test set of 30 images with 60 faces.

Jun [12] proposes a novel faster search scheme of gravity-center template matching compared with the traditional search method in an image for human face detection, which significantly saves the time consumed in rough detection of human faces in a mosaic image. Besides, the system is able to detect rather slanted faces (-25 ~ 25) and faces with much horizontal rotating angles (-45 ~ 45) and vertical rotating angles (-30 ~ 30), which are in various sizes and at unknown locations in an unconstrained background.

Hsiun and Fan [13] propose a robust and efficient human face detection system that can detect multiple faces in complex backgrounds. The proposed system consists of two primary parts. The first part is to search for the potential face regions by the feature of triangle formed by eyes-mouth. The second part is to perform face verification. Experimental results demonstrate that an approximately 98% success rate is achieved. The experimental results reveal that the proposed method is better than traditional methods in terms of efficiency and accuracy.

3. Template matching approach

Several standard patterns of a face are stored to describe the face as a whole or the facial features separately. The correlations between an input image and the stored patterns are computed for detection. These methods have been used for both face localization and detection.

Bruneli and Poggio [14] utilized template matching method to identify eye location. Given an input image, they use five scales of scanning window to globally search the image and to identify possible eye location. But this method still suffers from being sensitive to noises, distortion and variation in scale.

4. Appearance-based approach

The models are learned from a set of training images which should capture the representative variability of facial appearance. These learned models are then used for detection. These methods are designed mainly for face detection.

In 1991, Turk and Pentland [15] proposed an eigenfaces approaches for face detection and recognition. Their reasons are that the images of faces can be linearly encoded using a modest number of basis images. These basis images correspond to the eigenvectors obtained through principal component analysis. In general, only a few eigenvectors with the larger eigenvalues are kept to reduce computation complexity. In face detection step, a window is used to scan the whole image and the local image pattern is then projected onto the parameter subspace. The error between local image pattern and face template is measured by a specific distance measure.

Among all the face detection methods that used neural networks, the most significant work is arguably done by Rowley et al. [16-18]. A multilayer neural network is used to learn the face and nonface patterns from face/nonface images (i.e., the intensities and spatial relationships of pixels) whereas Sung and Poggio [19] used a neural network to find a discriminant function to classify face and nonface patterns using distance measures.

Support Vector Machines (SVMs) were first applied to face dtection by Osuna et al. [22]. SVMs can be thought of a new criterion to train polynomial function, neural networks, or radial basis function (RBF) classifiers. Most methods for training a classifier are based on of minimizing the training error. SVMs operate on another induction principle, structural risk minimization, which proposes to minimize an upper bound on the expected generalization error. A SVM classifier is a linear classifier where the separating hyperplane is chosen to minimize the expected classification error of the unseen test patterns. This optimal hyperplane is defined by a weighted combination of a small subset of the training vectors, called support vectors. Estimating the optimal hyperplane is equivalent to solving a inearly constrained quadratic programming problem. However, the computation is both time and memory intensive.

Jeffrey and Gong propose an application of SVM for detecting the face in mutliple views [23]. SVMs are extended to model the appearance of human faces which undergo non-linear change across multiple views. This approach uses inherent factors in the nature of the input images and the SVM classification algorithm to perform both multi-view face detection and pose estimation. The experimental result shows that pose estimation can be automatically performed by SVMs.

1.4 Assumptions

In this thesis, to concentrate on the methods for solving our proposed problems, we make the following assumptions.

1. Surveillance is an indoor environment.
2. Light condition is stable.
3. The digital camera is fixed.
4. Input images are gray-scale.
5. Only a single person is in the processing image.

1.5 System Description

We define the recognizable degree for the face, called recognizability. In general, a front face usually has the best recognizability. The recognizability of the face is low when the face is tilt, swing and rotation. The face with covers would affect the recognizability.

In this thesis, we develop a system to measure the recognizability of the detected face. The system flow diagram is shown in Fig.1.5.1. The main modules of the system are described as follows:

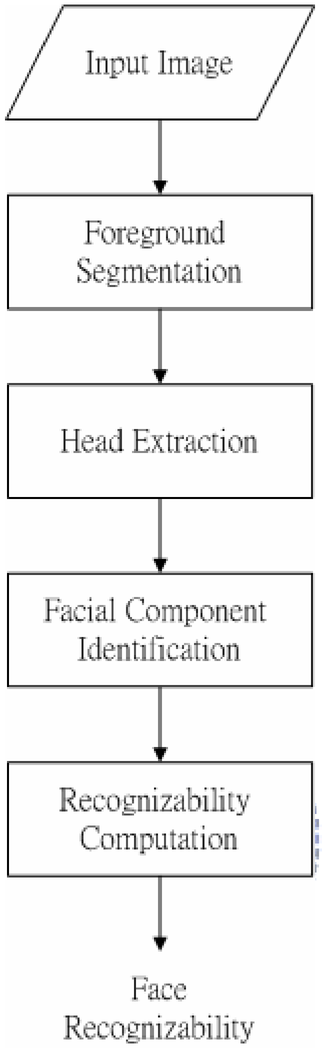


Fig.1.5.1 The system flow diagram.

1.5.1 Foreground Segmentation

We construct a statistical background model that could segment the foreground region from the input image. The foreground region would be collected to form the connected components.

1.5.2 Head Extraction

The system would extract the head region from the segmented foreground. In this phase, we propose an ellipse finding algorithm. Because the shape of a head is similar to an ellipse, it is suitable to use an ellipse to extract the head region for the foreground.

1.5.3 Facial Component Identification

In order to measure the recognizability, the locations of the facial components should be found. The intensity of a face should be in a range. It could segment the face from the head from intensity. Then we use an orientation histogram to compute the orientation's distribution of a face, and detect the lines formed by facial components with the major orientations. The detected lines would be filtered based on some properties. Finally, the locations of the facial components could be found.

1.5.4 Recognizability Estimation

We compute the recognizability for the extracted face by the facial information that includes the ratio of face area and head region, the region symmetry and the eyes-mouth triangle.

1.6 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 describes the foreground segmentation, the head extraction and the facial component identification. Chapter 3 describes the recognizability measurement. Chapter 4 describes experimental results and their analyses. Finally, chapter 5 presents some conclusions and suggestions for future works.

CHAPTER 2. FOREGROUND SEGMENTATION, HEAD EXTRACTION AND FACIAL COMPONENT IDENTIFICATION

The input images contain complex background. The foreground segmentation is an important pre-process. If the foreground is incomplete, it would make analysis incorrectly. For segmenting the foreground well, an effective background model is described in section 2.1. An ellipse finding algorithm would be used to find the suitable template to extract the head from the foreground. The algorithm is described in section 2.2. The method to identify the facial components of the face would be described in section 2.3.

2.1 Background Model

In this section, we describe the probability background model and the background subtraction approach to segment the foreground. Background subtraction segments moving regions in image sequences taken from a static camera by comparing each new frame to a model of the scene background. The range of intensity should be fixed for every pixel of the background, because the background is usually invariable. If the intensity of a pixel is rare to appear, the pixel should not belong to the background. Therefore, it is reasonable to use a statistical method to construct a background model.

For computing the probability, we have to get a sample of background. Assume the number of image in the sample is N. Let x_1, x_2, …, x_N be a sample of intensity values for a pixel.

The probability density function that this pixel will have intensity value x_t at time t can be non-parametrically estimated using the kernel estimator K as

$$Pr(x_t) = \frac{1}{N}\sum_{i=1}^{N} K(x_t - x_i)$$

If we choose our kernel estimator function, K, to be a Normal function $N(0, \Sigma)$, where Σ represents the kernel function bandwidth, then the density can be estimated as

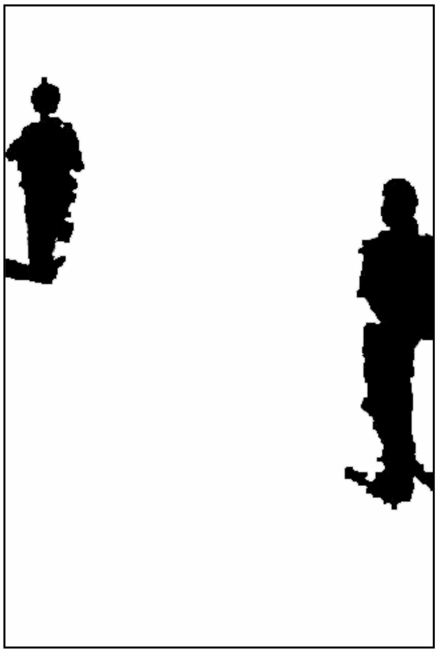
$$Pr(x_t) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} e^{-(x_t - x_i)\Sigma^{-1}(x_t - x_i)/2}$$

the density estimation is reduced to

$$Pr(x_t) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_t - x_i)^2}{2}}$$

Using this probability estimate the pixel is considered a foreground pixel if $Pr(x_t) < th$ where the threshold th is a global threshold over all the image. Next we find the connected components of the pixel whose $Pr(x_t) < th$. If the size of a connected component is too small, it would not be considered to the foreground. The smaller connected components could be affected by light or noise. Fig.2.1.1 shows some examples.

(a)

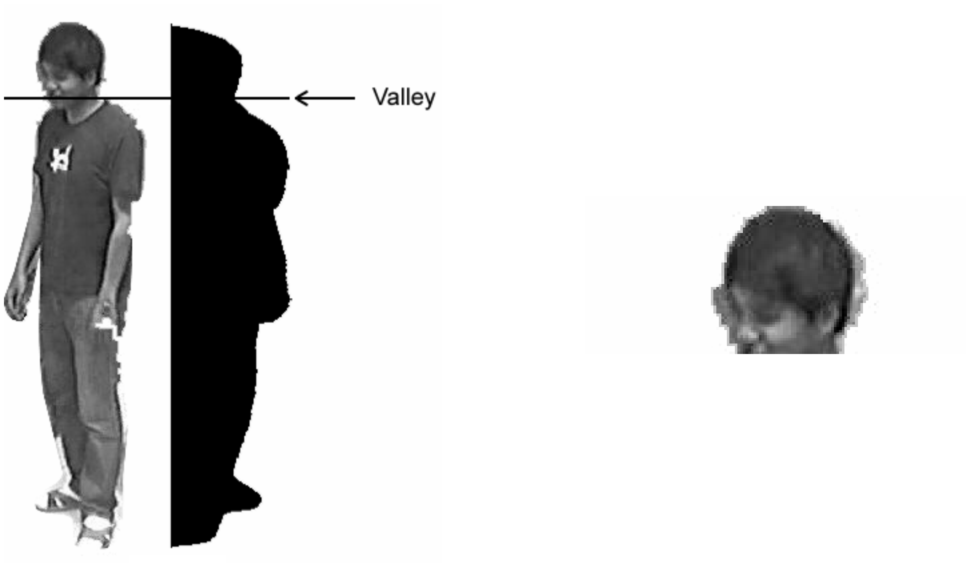


(b)

Fig.2.1.1. (a) An original image. (b) The foreground objects represent as white blocks.

2.2 Head Extraction

The head extraction is discussed in the section. An intuitive way to extract the head is that find the neck location of the foreground. The neck is narrower than its upper and lower components, such as, the head and the shoulder. We could find the neck location from the first valley in the horizontal projection profile, counting downward. If the head region is too large, we define the neck location in a fixed proportional location in the projection profile. This step is used to cope with the situation that the valley cannot be found reliably. The upper part from the location is taken as the head region. However, there are problems for extracting the head region by the neck location. If a person bows his head, a portion of the head could be lower than the neck location. The head information could be lost. Fig.2.2.1 shows an example. The extracted method by the neck location is not precise enough. It should be improved by other properties.



(a) (b)

Fig. 2.2.1 (a) The first valley of the horizontal projection could be the neck location. (b) The extracted head loses the mouth information.

2.2.1 Head extraction by ellipse

Because the shape of a head is an approximately ellipse, we could find an ellipse that is similar to the head region. Extracting the head by the ellipse would be more reliably. There are many methods to find a similar ellipse from a shape. Gravity-center-based and Hough transform are the common methods to detect the ellipse from a shape.

The gravity-center-based ellipse finding method is easy to implement. The procedure is as follows:

Gets the neck location from the horizontal projection.

1. Get the head region by the neck location.
2. Compute the horizontal and the vertical projection of the head region.
3. Average the location of all points along the contour, and get the location of gravity-center, (x_0,y_0).
4. Find the major axis a and minor axis b.
   - A. The former is defined as the line connecting the center point and the farthest point along the contour from the center.
   - B. The latter can be defined similarly.

The method is very fast, but it is not appropriate to find an ellipse for a human head. The gravity-center of the head is higher than the center of the head, such as the major axis would be incorrect.

Hough transform could find the most similar ellipse for the shape, but it costs much time to compute the all combinations of the ellipse function from the coordinates along the contour. An accelerated method [25] is proposed by P.S. Nair and A.T. Saunders. The procedure is as follows:

Gets the neck location from the horizontal projection.

1. Get the head region by the neck location.
2. Compute the cross point for the normal direction of the gradient of every two points around the shape.
3. A point is chosen from these points whose counter of location is the most maximum. The point (x0,y0) is the ellipse center.
4. Find the major axis *a* and minor axis *b*.
   A. The former is defined as the line connecting the center point and the farthest point along the contour from the center.
   B. The latter can be defined similarly.

The method can find the closest ellipse for a shape, but the time complexity is high, $O(N^2)$, N is the pixel number of a shape.

The both methods have some shortcomings to find the ellipse from the head, such that we propose another method to make it.

2.2.2 Our Ellipse Finding Algorithm

The ellipse finding algorithm consists of five steps, and the detail are described as follow:

1. Choose two horizontal lines (*A*,*B*) in the upper zone and lower zone. The upper zone for choosing line *A* is between 0.1**h* and 0.3**h*; the lower zone for choosing line *B* is between 0.7**h* and 0.9**h*, where *h* is the height of the segmented head by horizontal projection. (The illustration is shown in Fig.2.2.2.1.a)
   A. The head shape is not a perfect ellipse. We could take multiple lines in the two zones, it would find more ellipses.
   B. In our experiments, we take 2 lines of *A* and 2 lines of *B*.
2. Compute the intersection point between the foreground contour and two lines. Connect the midpoint in these two lines by a straight line *C*. (The illustration is shown in Fig.2.2.2.1.b)
3. Along the orthogonal direction on both sides of line *C* and get the distance to the contour. (The illustration is shown in Fig.2.2.2.1.c)
4. Choose the point (x_0, y_0) in line *C* whose distances to both sides are equal and the longest. This point (x_0, y_0) is regarded as the center of an ellipse. (The illustration is shown in Fig.2.2.2.1.d)
5. Find the major axis *a* and minor axis *b*. (The illustration is shown in Fig.2.2.2.1.e)
   A. The former is defined as the line connecting the center point and the farthest point along the contour from the center.
   B. The latter can be defined similarly.

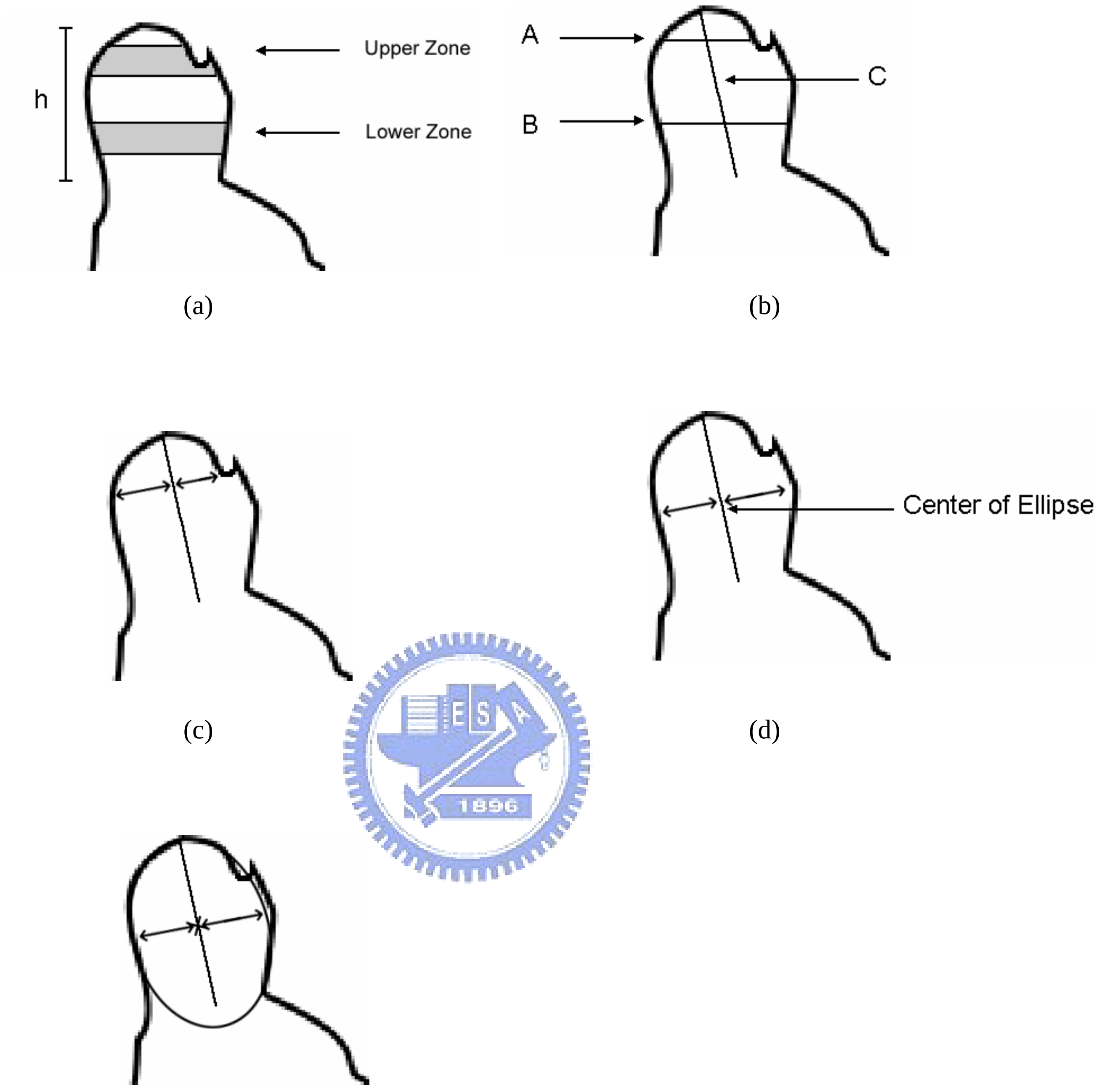


Fig. 2.2.2.1 (a)(b)(c)(d)(e) The illustrations of our ellipse finding algorithm.

This method could find the similar ellipse for the segmented head quickly, but it could find several ellipse candidates for the head region. We have to select the most appropriate ellipse to be the template. All ellipse candidates should be similar to the head region, such that we select the largest ellipse to extract the head for getting the most information. Fig.2.2.2.2 shows a result of our ellipse finding algorithm.

Fig. 2.2.2.2 The result of head extraction. The head is bounded by the white ellipse.

2.2.3 Ellipse Estimation Method Comparison

The gravity-center-based ellipse finding method is fast and easy to implement. However, the gravity-center for the head is usually higher than the center of the head. The prosperity would make the gravity-center-based method find an incorrect ellipse-center for the head. Fig.2.2.3.c shows an example.

Hough transform based ellipse detection algorithm could find the best ellipse for the head region, but it costs a lot of time. Fig.2.2.3.d shows an example.

Our ellipse finding algorithm is fast. It is not accurate, but bittern than the gravity-center method. Fig.2.2.3.e shows an example.

The comparisons of these methods are listed in Table 2.1.

Table 2.1 Comparison of Ellipse Detection Methods

	Advantage	Disadvantage
Gravity-center	▪ Fast ▪ Easy to implement	▪ Not appropriate to detect an ellipse for a head.
Hough transform	▪ Accurate ▪ Reliable	▪ Very slow ▪ Time complexity is $O(N^2)$
Our method	▪ Fast	▪ Not accurate, but better than gravity-center methods.

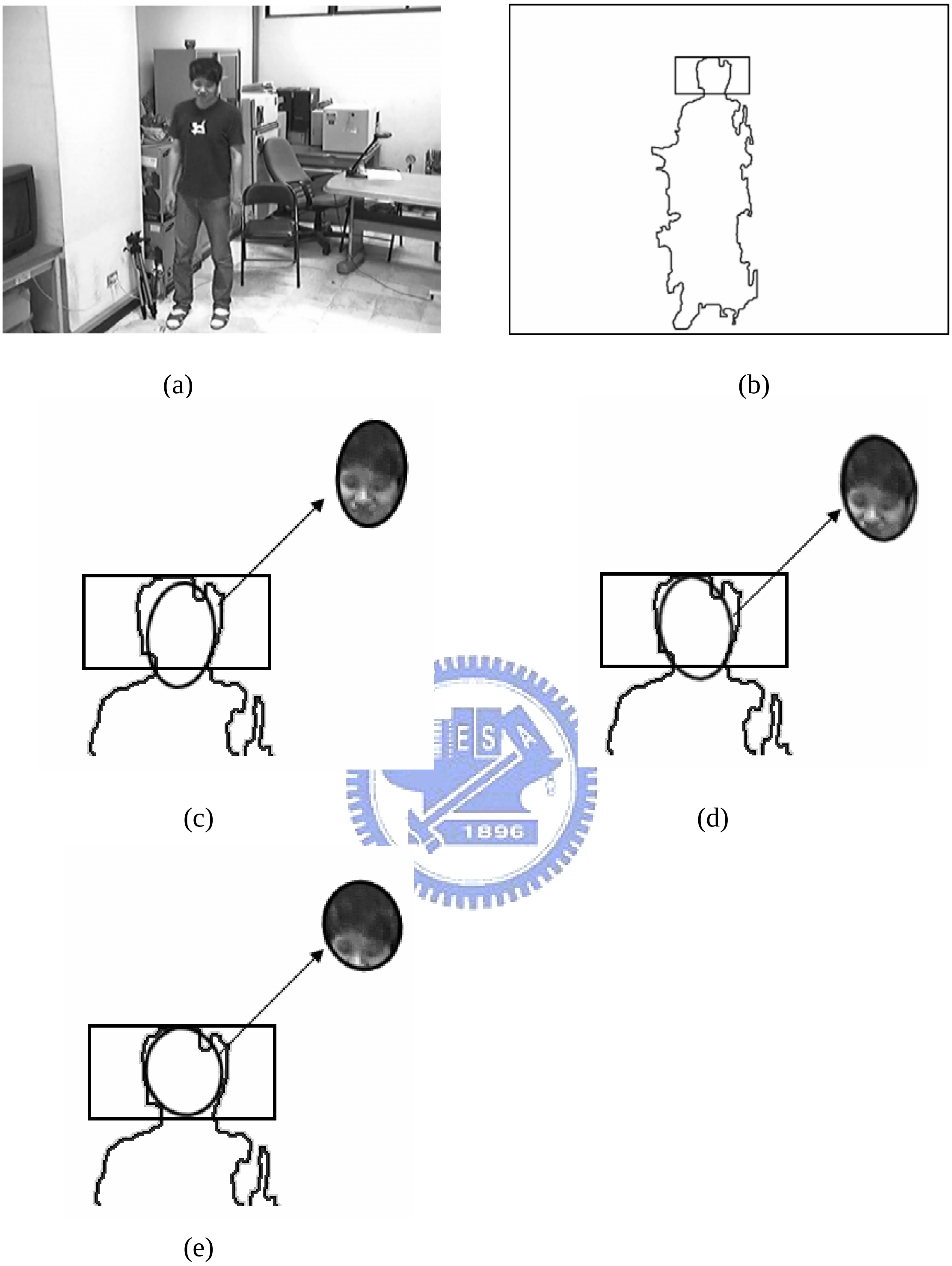


Fig.2.2.3 (a) The original image. (b) The contour of the foreground, the top box bounds the head region by the neck position. (c) The result of the gravity-center-based ellipse finding method. (d) The result of Hough transform based ellipse detection algorithm. (e) The result of our ellipse finding algorithm.

2.3 Facial Component Identification

Next we want to extract the facial components from the head. In the human faces, the orientations of each facial component are strong. For the upright and the frontal face, the facial components can be roughly divided into two groups: horizontal (for most organs such as eyebrows, eyes and mouth) and vertical (for nose and cheek). The facial components would form some obvious and darker lines in the face. If we could find the obvious lines in the face, we should locate the facial components.

Our facial component identification consists of five stages.

Stage 1. Face extraction

Stage 2. Orientation analysis

Stage 3. Line detection

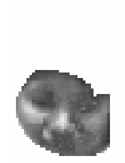
Stage 4. Line filtering

Stage 5. Facial component identification

2.3.1 Stage1: Segment face component

Firstly, the system needs to segment the face component from the head. When the light condition is stable, the intensity range of skin should be fixed. We could use the property to segment the face from the head. The intensity range of skin is between 65 and 145, trained from 40 faces in our experiments. Fig.2.3.1 shows an example.

(a)



(b)

Fig.2.3.1 (a) The head is bounded by the white ellipse. (b) The face image.

2.3.2 Stage2: Orientation Analysis

Zhou had presented an orientation analysis for rotated human face detection [6]. The orientation histogram is constructed from statistical analysis of the face images, which are combined by the orientation's distribution and its strength. We could compute the possible orientations of the facial components from the orientation histogram, because the strong orientations of the facial components.

For computing the orientation histogram, we have to calculate the gradient and the strength for every pixel.

It has been demonstrated that for a strong oriented intensity pattern along one direction the power spectrum of such a pattern clusters along a line through the origin in Fourier transformation domain and the direction of the line is perpendicular to the dominant spatial orientation. Instead of the actual computation in the Fourier transformation domain the orientation map, $\theta(x, y)$, and anisotropic strength map, $g(x, y)$, can be calculated directly from intensity image, $I(x, y)$, and its first order partial derivatives by

$$\theta(x,y) = \frac{1}{2}\tan^{-1}\frac{\int\int_{\Omega} 2I_x I_y dxdy}{\int\int_{\Omega}(I_x^2 - I_y^2)dxdy} + \frac{\pi}{2}$$

$$g(x,y) = \frac{(\int\int_{\Omega}(I_x^2 - I_y^2)dxdy)^2 + (\int\int_{\Omega} 2I_x I_y)^2}{(\int\int_{\Omega}(I_x^2 - I_y^2)dxdy)^2}$$

I_x and I_y are partial derivatives of $I(x,y)$. Ω is the small neighborhood of pixel, (x, y), whose size is related with the size of the object that we want to analyze. In our experiment, Ω is a tenth of head size. The value of $g(x,y)$ is between 0 and 1 (close to 1 for a strongly oriented pattern, and 0 for isotropic regions).

In order to analyze the distribution of orientations statistically, we use orientation histogram, by combining the orientation the orientation's distribution and its strength. α is the angle of any orientation between 0 and 180.

$$H(\alpha) = \sum_{\theta(x,y)=\alpha} g(x,y)$$

Fig.2.3.2.1 shows some examples for the orientation histogram. There are strongly orientations along to degree 45 and 135 in (a)(c), and the orientation histograms (b)(d) have two peaks at degree 45 and 135. (e)(g) is equal to (c), but add some Gussian noise. Their orientation histograms are still robust. It shows that the orientation histogram has a good ability of noise resistance.

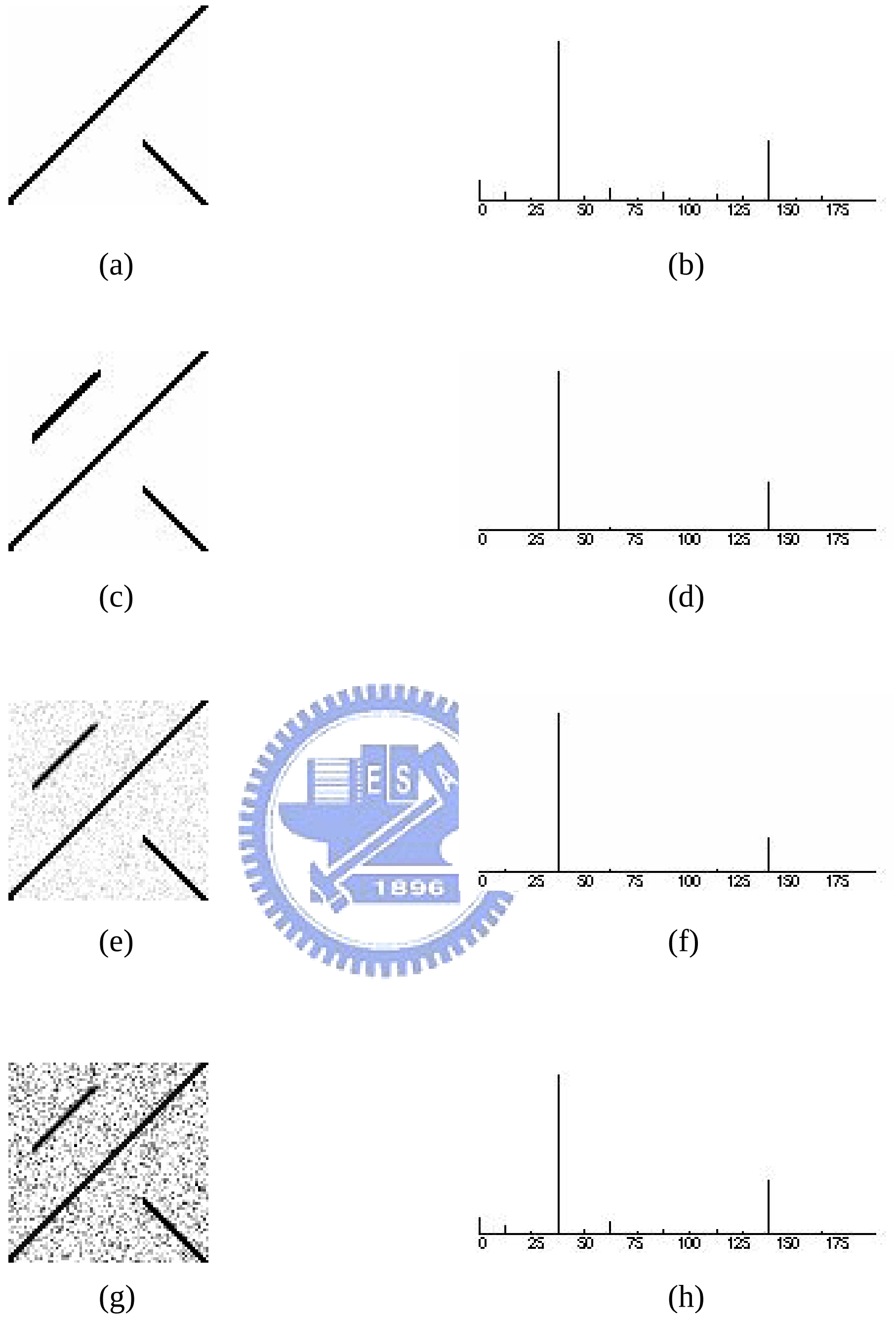
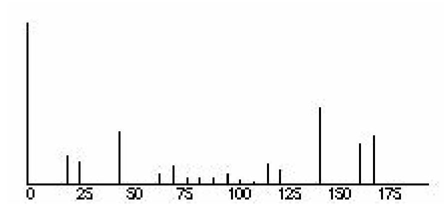


Fig.2.3.2.1 The examples of the orientation histogram. (a)(c) The original image. (e)(g) The image by adding some Gussian noise. (b)(d)(f)(h) The orientation histogram of (a)(c)(e)(g).

The face has several high peaks in the orientation distribution, called the major peaks. The face shown in Fig.2.3.2.a has the strong horizontal orientation and the orientation histogram shown in Fig.2.3.2.b that there is a peak at the angle 0. The orientation histogram could represent the orientation characteristic for the analyzed region.

(a)



(b)

Fig.2.3.2.2 (a) The face image. (b) The orientation histogram of the face.

2.3.3 Stage 3: Line Detection

The orientation distribution would be analyzed in Stage 2. According to obvious orientations, we would detect the lines along these orientations. The procedure of the line detection is as follows:

- Select k angles, $(t_1, t_2, \cdots, t_k)$ with the strongest distribution from the orientation histogram.
- For each angle t_i, $(i = 1, 2, \cdots, k)$

1. Detect the line for $point_1(x_1, y_1)$ by the order up to down and left to right.
2. Get $point_2(x_2, y_2)$ from $point_1$ and t_i . (d is 5 in our experiment)

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x_0 + d \\ y_0 + d \tan\theta \end{bmatrix}$$

3. If there is a point near $point_2$, whose angle is the same with t_i, then connect $point_1$ and $point_2$ as a line. The illustration shows in Fig.2.3.3.

4. Let $point_2$ as $point_1$, back to Step2, until the angle of $point_2$ is different with t_i.

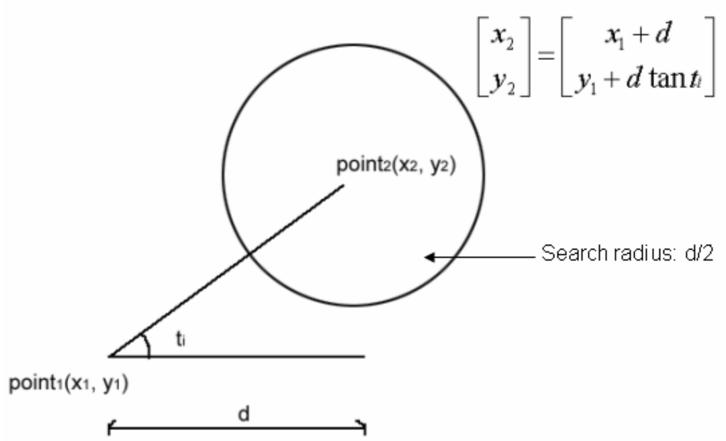


Fig.2.3.3.1 The illustration of the line detection.

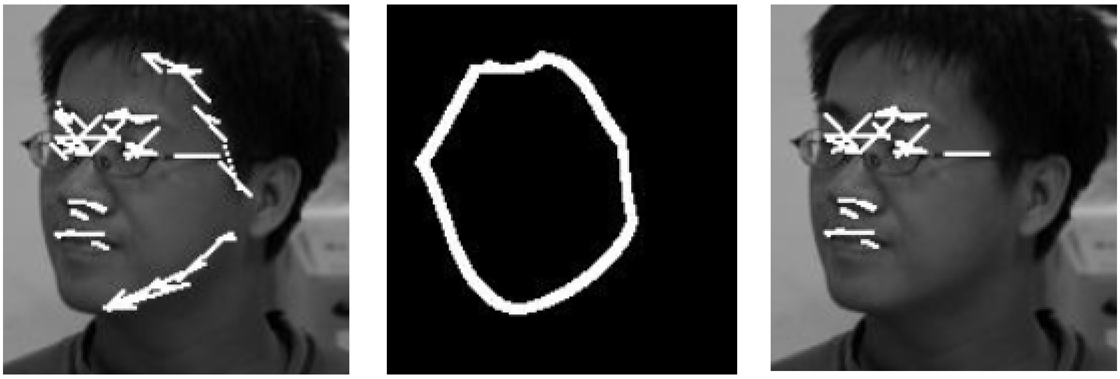
The examples of the line detection show in Fig.2.3.3.2.

Fig.2.3.4. The examples of the line detection.

2.3.4 Stage 4: Line Filtering

There are false lines detected in previous processes. For instance, the lines formed by face contour, cheek and eyeglass frame. The identification would be incorrect by interference with these false lines, such that we need to filter the result from the line detection.

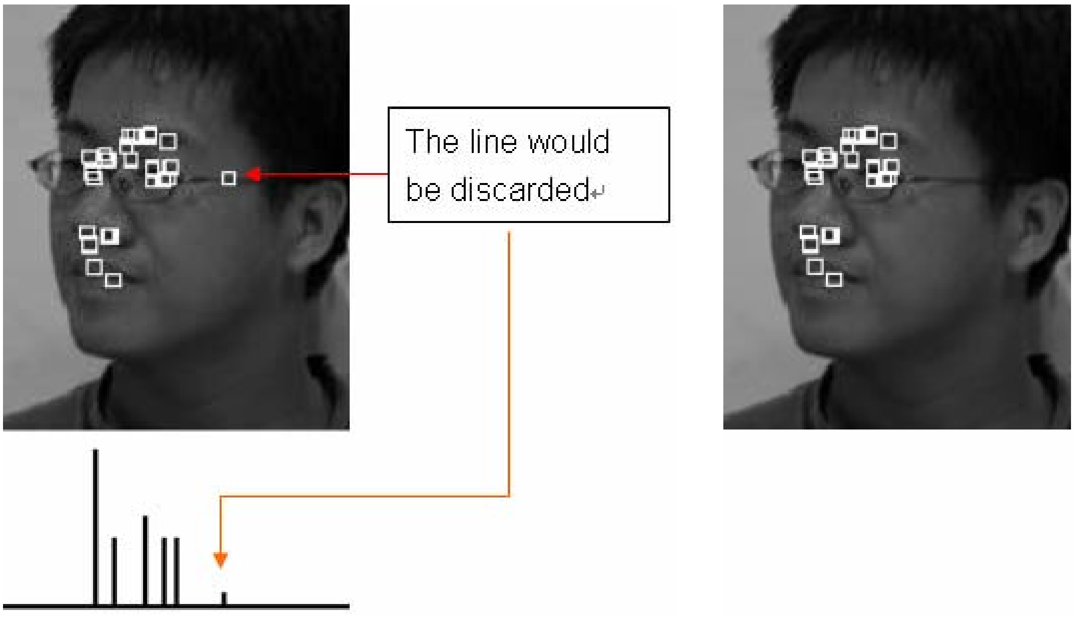
Since the lines formed by the shadow of the facial contour and the hair are near the head contour, we could remove the lines near the face contour. Fig.2.3.4.1 shows an example of line filtering based on the contour.



(a) (b) (c)

Fig. 2.3.4.1 (a) The result of the line detection. (b) The facial contour. (c) The result of the filter based on the contour.

The vertical distribution of line-centers formed around two eyes and mouth has higher values than that of cheek or glass frame. For removing these lines, we could delete the line-centers with low distribution values in the vertical projection. The remainder lines should belong to the facial components. Fig.2.3.4.2 shows the example of the line filtering based on the vertical projection.



(a) (b)

Fig.2.3.4.2 (a) The distribution of the line-center after filtering based on the contour. (b) The result of the line filter based on the vertical projection.

2.3.5 Stage 5: Facial Component Identification

For the remained lines, we wish to find the lines that could represent the facial components. The lines formed by their facial components have the same properties. For instance, the lines formed by eyes whose intensity should be darker and the location is higher, the lines formed by mouth whose intensity should be darker than the intensity around the mouth and the location is lower. The properties of lines formed by facial components are listed as follow:

The line formed by eyes

1. Location is higher.
2. Intensity is darker.

For the line formed by the mouth

1. Location is lower

2. Intensity of the line is deeper than intensity around this line

We could use these properties to identify the lines with their facial components. The procedure of the identification has three steps:

1. Remove the light intensity lines.
2. Identify the lines with the appropriate facial components.
3. Merge the lines with the same facial components.

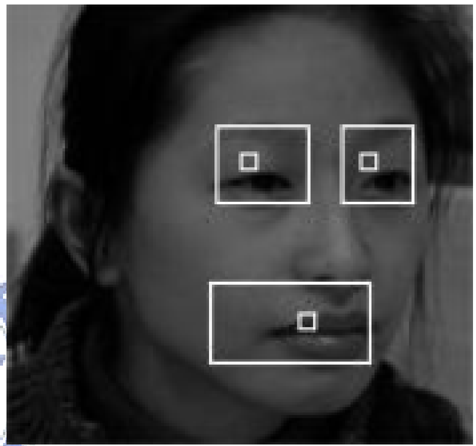
Fig.2.3.5 shows the example of the facial component identification.

There are exceptions that could occur in the facial component identification. For example, face with cover, long hair, moustache, changed expression and strong light effect. These factors would affect the orientation's distribution for face, and result in the incorrect result. These exceptions would be discussed in section 4.3.

(a)

(b)

(c)



(d)

Fig.2.3.5 (a) The result of the line detection. (b) The result of removing the light intensity lines. (c) The result of identifying the lines with the appropriate facial components. (d) The result of merging the lines with the same facial components. We use the average location of the line-centers to represent the location of the facial component.

CHAPTER 3. COMPUTATION OF FACE RECOGNIZABILITY

In this chapter, we would compute the recognizability from the facial features. It is necessary to analyze the factors that could affect the recognition. The analysis is described in section 3.1. The measurements of the recognizability are described in section 3.2. The computation for the recognizability is described in section 3.3.

3.1 Factors which Affecting Recognition Results

There are many factors that could affect the recognition. Whether a face could be recognized correctly depends on the extraction rate of the facial features. In general, a frontal face is the best candidate for the recognition. If the face is oriented, the facial features would be deformed or incomplete. In the other side, the size of the detected face would affect the recognizable result, too. The small face would be distorted, when it is normalized to the size of the recognition system. Before we recognize a human face using detected facial features, we use a simple measurement, called recognizability hereafter, to indicate the possibility of face recognition. If a lower measurement is obtained, the recognition rate may be poor and face recognition is not recommended.

We define the recognizability of a face according to its orientation and size. The situations of oriented face could be classified into three classes.

1. Frontal face. (Fig.3.1.1.a)
2. Oriented and rotated face. (Fig.3.1.1.b)
3. Non-frontal face, include face with covers. (Fig.3.1.1.c)

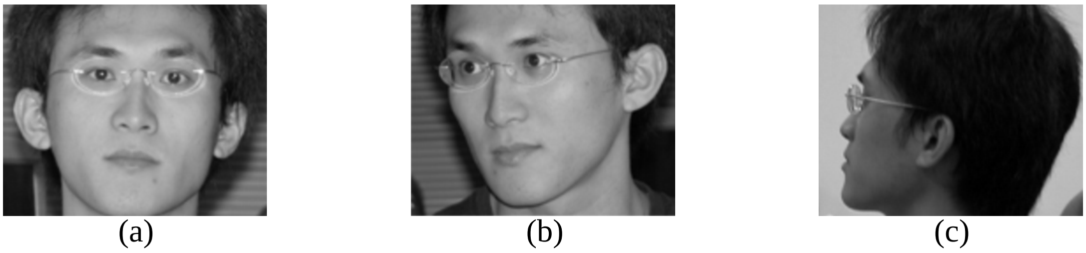


Fig.3.1.1. (a) The frontal face. (b) The oriented face. (c) The non-frontal face.

3.2 Recognizability Measurement

For measuring the recognizability, we should analyze the facial features under various orientations.

The properties of a frontal face:

1. The orientation distribution is symmetric.
2. The intensity distribution of the face region should be symmetric along a symmetric axis.
3. The symmetric axis could be computed from a symmetric measurement. [6]
4. The triangle formed by the centers of eyes and mouth, called eyes-mouth triangle, could be isosceles in the center region of the head. (Fig.3.2.1)

The properties of an oriented face:

1. The intensity distribution of face region should be asymmetric.
2. The eyes-mouth center triangle could not be an isosceles triangle.
3. While the frontal face turns to profile, the triangle deforms from isosceles triangle to right triangle.

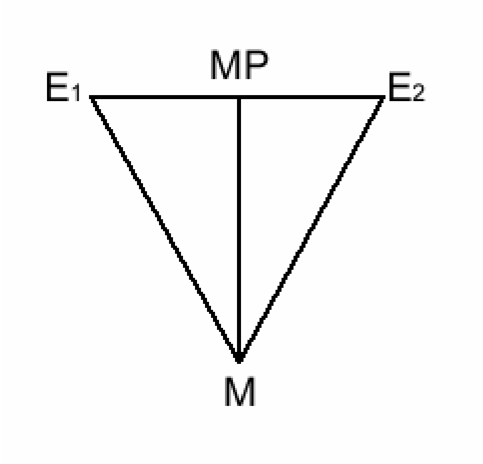


Fig.3.2.1 The triangle formed by eyebrows/eyes and mouth. E_1 is the location of left eyebrow or eye. E_2 is the location of right eyebrow or eye. MP is the location of mouth. MP is the projection of mouth in the E_1E_2.

The properties of a non-frontal face:

1. The eyes-mouth center triangle would not be formed because some facial components could not be detected.

After analyzing the frontal, oriented and non-frontal face, we could obtain the two measurements:

1. Similar Degree of a Triangle based on an Isosceles Triangle (*S*)
2. Region Asymmetric Degree (*RAS*)

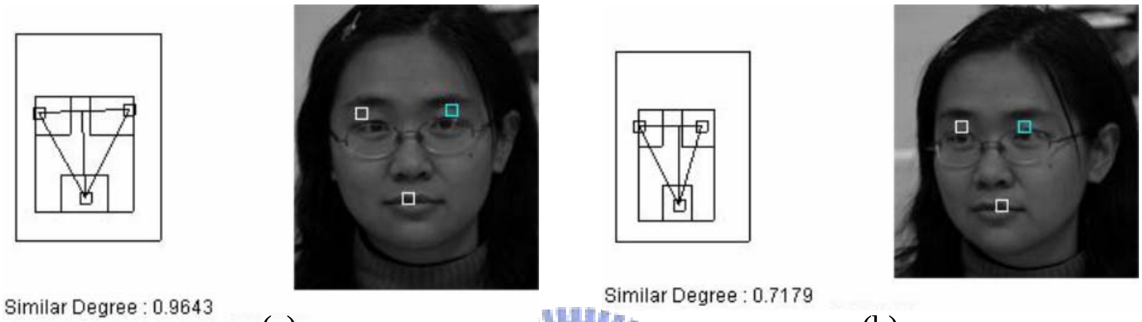
The detected face would be classified as frontal, oriented and non-frontal according to the two measurements.

3.2.1 Similar Degree of a Triangle based on an Isosceles Triangle

The triangle formed by eyes and mouth is shown as Fig.3.2.1. For measuring the degree of the facial orientation, we use an equation to compute to the similar degree of the triangle based on the isosceles triangle.

$$S = \frac{Min(\overline{E_1MP}, \overline{E_2MP})}{0.5\overline{E_1E_2}}$$

The value of S is between 0 and 1. S is high when the triangle is similar to isosceles triangle. S is low when the triangle is different from isosceles triangles. Fig.3.2.2 shows the example of the similar degree for the frontal and the oriented face.



(a) (b)

Fig.3.2.2 (a) The similar degree for the frontal face is 0.9643. (b) The similar degree for the oriented face is 0.719.

3.2.2 Degree of Region Asymmetry

The human face has the strong symmetry along the nose vertically, that is, the symmetric axis. The symmetric axis could be computed from the orientation histogram. The symmetric axis could be found from the orthogonal orientation of the line formed by eyes and the angle of the symmetric axis should be the orthogonal angle of the major peak for the orientation histogram (see Fig.3.2.3).

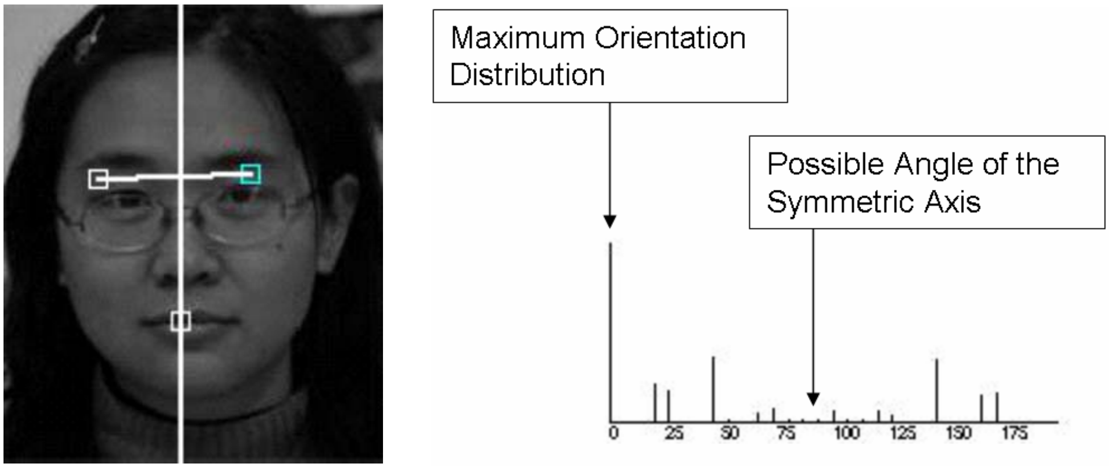


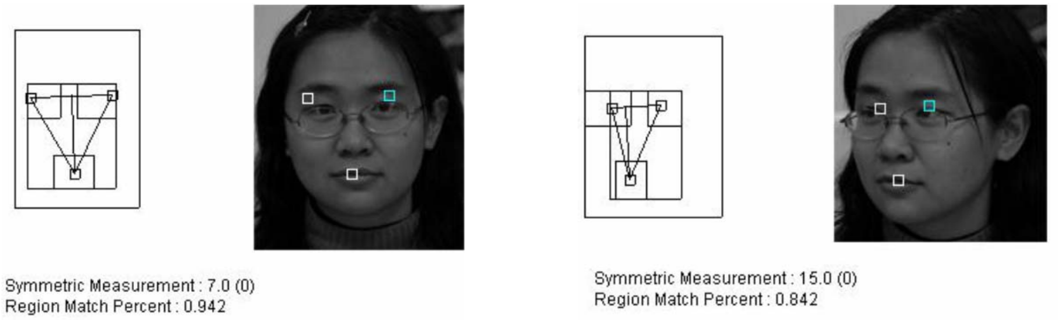
Fig.3.2.3. The angle of the symmetric axis could be found from the normal angle of the line formed by two eyes and the orientation histogram.

For computing the symmetry of the face, we use the two equations to compute the symmetry along the symmetric axis.

$$RAS = \frac{IntensityDifferenceOfMatchPixel(left, right)}{MatchPixelCounter(left, right)}$$

$$P = \frac{MatchPixelCounter(left, right)}{SumOfPixelCounter(left, right)}$$

where *left* is the left region, *right* is the right region of the face. *IntensityDifferenceOfMatchPixel* is the difference of the intensity between the left and the right face. *MatchPixelCounter* is the count of the match pixels between the left and the right face. *SumOfPixelCounter* is the sum of the left and the right face. If *RAS* <= 16 and P >= 0.90, the region is symmetric. Otherwise, the region is asymmetric. Fig.3.2.4 shows the example of the degree of region asymmetry.



(a) (b)

Fig.3.2.4. (a) The degree of region asymmetry for the frontal face. (b) The degree of region asymmetry for the oriented face.

3.2.3 Size Rate

The size of the detected face could be normalized to the size of the recognition format. The face could have the distortion during the normalization. The recognition result would be affected by the distortion. Therefore, the rate of the size of the detected face and the size of the recognition format, we define the rate as *size*. Fig.3.2.5 shows that the smaller face would have the greater distortion.

(a) (b) (c) (d)

Fig.3.2.5 (a) The original image. (b) The size of the image is half of the original image. (c) The size of the image is one fourth of the original image. (d) The size of the image is one eighth of the original image.

3.3 Recognizability Computation

In this section, we would introduce the recognizability computation. The recognizability, *Recog*, is defined as the product of similar degree, *S*, and the relation size, *size*. According to the eyes-mouth triangle, the orientation of the face could be classified into three classes: frontal, oriented and non-frontal. The relation between the measurements and the classes would be discussed later.

3.3.1 Class 1: Frontal Face

The frontal face would be classified into Class 1. The eyes-mouth triangle of the frontal face is in the center and the intensity of the face region is symmetric. The properties of Class 1 are listed as follows:

- The eyes-mouth triangle is in the center of the face region.
- $RAS \leq 16$
- $P \geq 0.9$
- $Recog = 1 * size$

Fig.3.3.1 shows the examples of Class 1.

Fig.3.3.1. The examples of Class 1.

3.3.2 Class 2: Oriented Face

The oriented face would be classified into Class 2. The triangle of the oriented face is not in the center of the face region and the intensity of the face region is asymmetric. The properties of Class 2 are listed as follows:

- The eyes-mouth triangle is not in the center of the face region.
- $S < 0.9$
- $RAS > 16$
- $P < 0.9$
- $Recog = S * size$

Fig.3.3.2 shows the examples of Class 2.

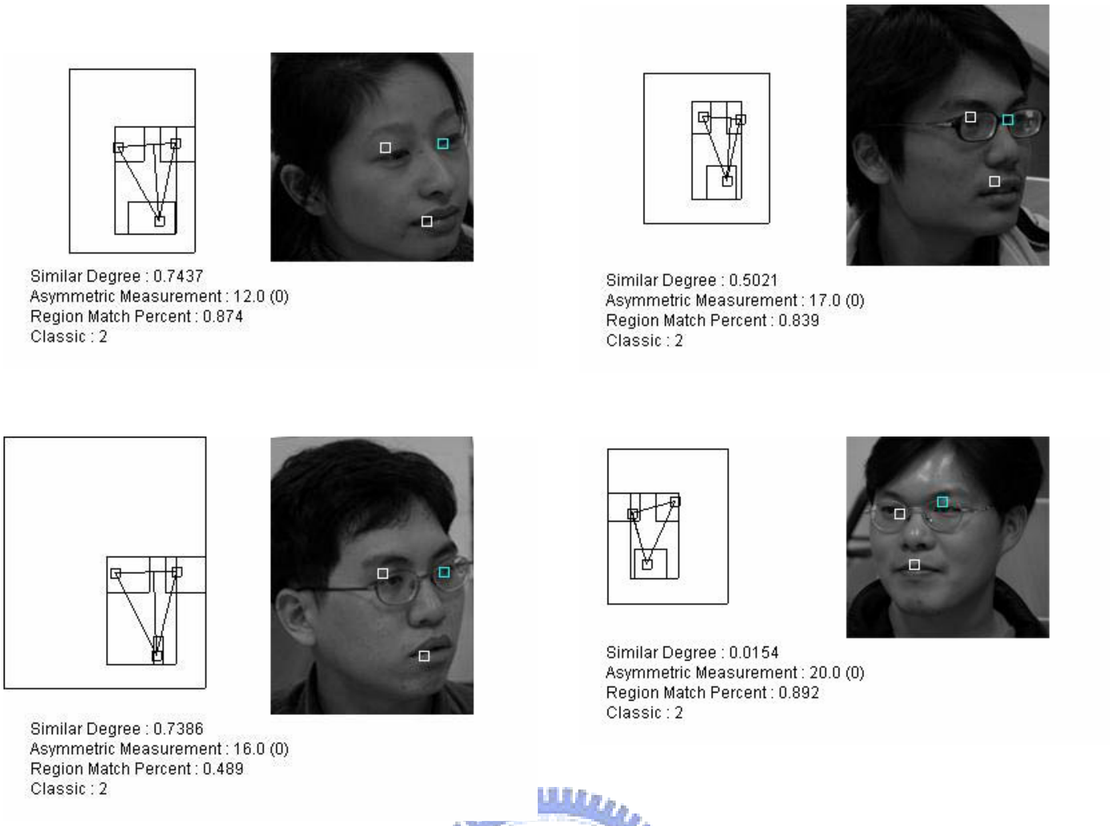


Fig.3.3.2. The examples of Class 2.

3.3.3 Class 3: Non-frontal Face

The non-frontal face and the face with covers would be classified into Class 3. The eyes-mouth triangle could not be found, because some facial components could not be detected. The properties of Class 3 are listed as follows:

- S is non-computable
- $Recog = 0$

Fig.3.3.3 shows the examples of Class 3.

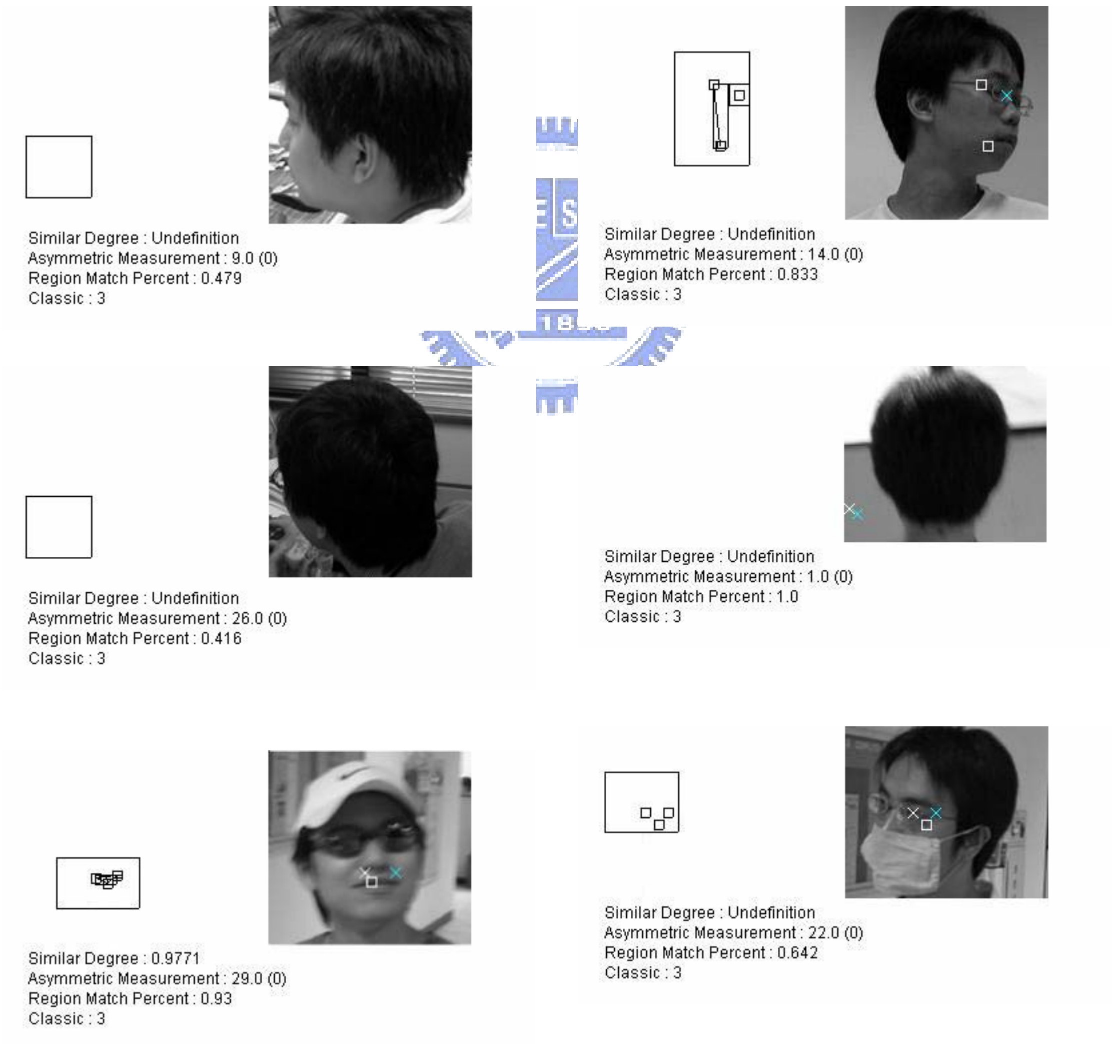


Fig.3.3.3. The examples of Class 3.

CHAPTER 4. EXPERIMENTAL RESULT AND DISCUSSION

In this chapter, we present the experimental results and the discussion of our system. The proposed approach has been implemented on the CPU, Pentium IV 2GHz, with 512 MB RAM. The software environment is: (1) Microsoft Windows XP (2) Borland JBuilder 9 (3) Microsoft Visual C++ 6.0.

The input images are gray scale, whose sizes are between 640x480 and 1024x768. The size of the detected faces is between 46x56 and 152x170. There are 229 images from 15 people. The recognition system references to [27]. We take 75 images from 15 people as the training sample, each person taken 5 images from different views and take others as the testing sample.

4.1 Result of Face Classification

In this section, we present the results of covers classification. There are 30 frontal faces, 104 oriented faces and 20 non-frontal faces. The result of the face classification is shown in Table 4.1. The incorrect result of the classification would be discussed in section 4.3.

Table 4.1 The result of the face classification

	Frontal (30)	Oriented (104)	Non-frontal (20)
Class 1	28	2	1
Class 2	2	97	0
Class 3	0	5	19

4.2 Relation between Recognizability and Recognition Rate

The faces with the different size rates would be tested in our experiments. The recognizability and the recognition rate with the size rate, 100%, 80% and 60% are listed in Table 4.2-4.

Table 4.2 The recognizability and the recognition rate with the size rate 100%.

	Class 1	Class 2			Class 3
Recognizability	1	0.8 ~ 1	0.4 ~ 0.8	0 ~ 0.4	0
Recognition Rate	90% (28/31)	82% (34/41)	67% (25/37)	30% (8/26)	0% (0/19)

Table 4.3 The recognizability and the recognition rate with the size rate 80%.

	Class 1	Class 2			Class 3
Recognizability	0.8	0.64 ~ 0.8	0.32 ~ 0.64	0 ~ 0.32	0
Recognition Rate	64% (22/34)	52% (20/38)	32% (12/37)	24% (6/25)	0% (0/20)

Table 4.4 The recognizability and the recognition rate with the size rate 60%.

	Class 1	Class 2			Class 3
Recognizability	0.6	0.48 ~ 0.6	0.24 ~ 0.48	0 ~ 0.24	0
Recognition Rate	59% (19/36)	35% (14/39)	28% (11/38)	23% (7/30)	0% (0/20)

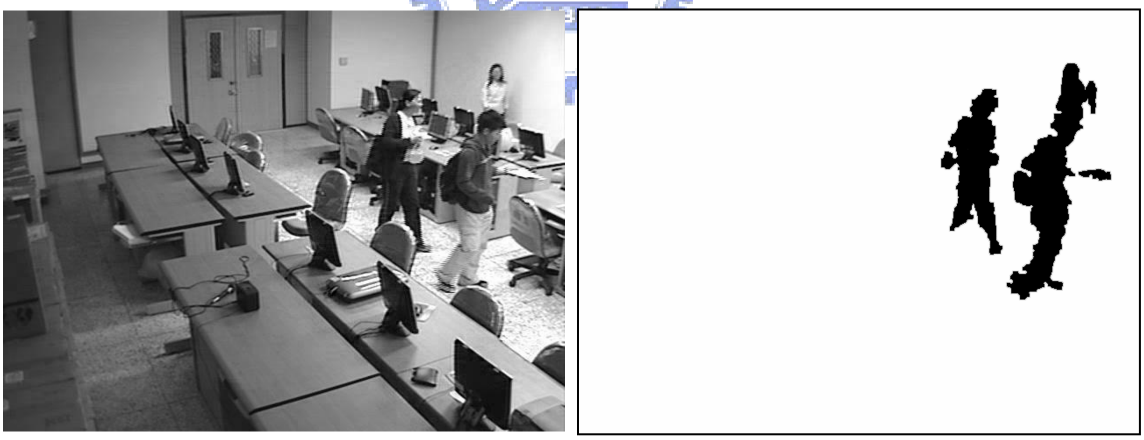
From the Table 4.2-4, we could found that the recognition rate of the face with the same recognizability is lower while the size of the detected face is smaller. While the orientation of the face is larger, the recognition rate and the recognizability of that are lower. The recognizability is related to the recognition rate, and measure the correct recognition rate for the face.

4.3 Error Analysis

The error analyses would be discussed in the section.

4.3.1 Error Analysis: People in Group

Our system could only solve the single person. If there are people overlapped in the image, the head extraction would not work correctly (Fig.4.3.1).



(a) (b)

Fig.4.3.1. The example of people in group. (a) The input image. (b) The result of the foreground segmentation.

4.3.2 Error Analysis: Tilted Head

Because the features of the facial component identification only have tiny changes, the system could not detect the tilted head efficiently (Fig.4.3.2).

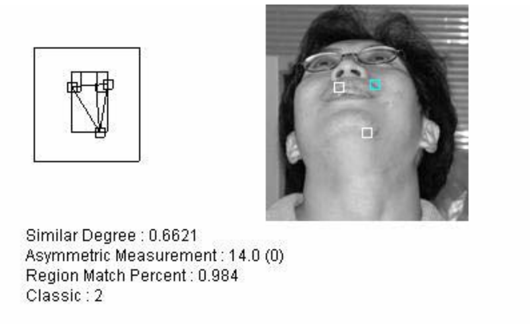


Fig.4.3.2. An example of the tilted head.

4.3.3 Error Analysis: False Facial Component Identification

There are some errors that could not be handled properly. For instance, the light effect would affect the result of the face segmentation (Fig.4.3.3).

Fig.4.3.3. The recognizability measurement is in the dark light condition.

CHAPTER 5. CONCLUSION AND FUTURE WORK

In this thesis, we design a system of computation of face recognizability. The system consists of four stages: foreground segmentation, head extraction, facial component identification and recognizability computation.

We apply the statistical background model and a background subtraction approach to segment the foreground from the image. The method could segment the foregrounds efficiently.

An ellipse finding algorithm method is proposed to extract human head. The method could find a suitable ellipse to extract the head region from the foreground. The extraction method is fast and efficient.

A method is proposed to extract the facial components under various orientations. The method analyzes the orientation distribution for the face by the orientation histogram. It detects the lines with the major peaks of the orientation histogram, and using the line-centers to estimate the locations of the facial components.

A method is proposed to estimate recognizability. The recognizability would be computed from the facial features, for instance, the triangle formed by the facial components, the degree of direction symmetry and the degree of region symmetry. The experimental results show that the recognizability could supply the reliable measurement to the recognition system.

For the future work, we suggest several directions to improve the performance of our system.

The foreground segmentation could be more efficient by adding the color information. The problem of people in group could be solved by using the tracking information.

The facial component identification could be more accurate by adding color information. For instance, the mouth could be extracted correctly. The facial component identification could be more accurate by using other approaches.

The recognizability measurement could be more credible by adding more features. For example, the recognition of human action could be helpful to classify the facial orientations.

REFERENCE

[1] H. Yang, M.D. Levine, "The background primal sketch: an approach for tracking moving objects," *Machine Vision and applications*, pp. 17-34, 1992.

[2] C. StauOer and W. Grimson, "Adaptive background mixture models for real-time tracking," *Proceedings of the IEEE CS Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 246–252, 1999.

[3] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric Model for Background Subtraction," *Proc. IEEE Frame Rate Workshop*, 1999.

[4] M.H. Yang, D.J. Kriegman, and N. Ahuja, "Detecting faces in images a survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Volume: 24 Issue: 1, pp. 34-58, Jan. 2002.

[5] G. Yang and T. S. Huang, "Human Face Detection in Complex Background," *Pattern Recognition*, vol. 27, no. 1, pp. 53-63, 1994.

[6] J. Zhou, X. Guang Lu, D. Zhang and C.Y. Wu, "Orientation analysis for rotated human face detection," *Pattern Recognition*, Volume: 20, Issue: 4, pp. 257-264, April, 2002.

[7] S.A. Sirohey, "Human Face Segmentation and Identification," *Technical Report CS-TR-3176, Univ. of Maryland*, 1993.

[8] K.C. Yow and R. Cipolla, "A Probabilistic Framework for Perceptual Grouping of Features for Human Face Detection," in *Proc. Second Int'l Conf. Automatic Face and Gesture Recognition*, pp. 16-21, 1996.

[9] K.C. Yow and R. Cipolla, "Feature-Based Human Face Detection," *Image and Vision Computing*, vol. 15, no. 9, pp. 713-735, 1997.

[10] M.F. Augusteijn and T.L. Skujca, "Identification of Human Faces through Texture-Based Feature Recognition and Neural Network Technology," *Proc.*

IEEE Conf. Neural Networks, pp. 392-398, 1993.

[11] Y. Dai and Y. Nakano, "Face-Texture Model Based on SGLD and its Application in Face Detection in a Color Scene," *Pattern Recognition*, vol. 29, no. 6, pp. 1007-1017, 1996.

[12] M. Jun, Y. Baocai, W. Kongqiao, S. Lansun, and C. Xuecun, "A hierarchical multiscale and multiangle system for human face detection in a complex background using gravity-center template," *Pattern Recognition*, Volume: 32, Issue: 7, pp. 1237-1248, July, 1999.

[13] L. Chiunhsiun and K.C. Fan, "Triangle-based approach to the detection of human face," *Pattern Recognition*, Volume: 34, Issue: 6, pp. 1271-1284, June, 2001.

[14] R. Bruneli and T. Poggio, "Face recognition: Features versus templates," *IEEE Transactions, Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042-1052, 1993.

[15] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71- 86, 1991.

[16] H. Rowley, S. Baluja, and T. Kanade, "Human Face Detection in Visual Scenes," *Advances in Neural Information Processing Systems 8, D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo, eds*., pp. 875-881, 1996.

[17] H. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 203-208, 1996.

[18] H. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23-38, Jan. 1998. Color and Local Symmetry Information," *Proc. Third Int'l Conf. Automatic Face and Gesture Recognition*, pp. 130-135, 1998.

[19] K.K. Sung and T. Poggio, "Example-Based Learning for View-Based Human

Face Detection," *Technical Report AI Memo 1521, Massachusetts Inst. of Technology AI Lab*, 1994.

[20] O. Ayinde and Y.H. Yang, "Region-based face detection," *Pattern Recognitio*n, Volume: 35, Issue: 10, pp. 2095-2107, October, 2002.

[21] L.L. Huang, A. Shimizu, Y. Hagihara and H. Kobatatke, "Gradient feature extraction for classification-based face detection," *Pattern Recognition*, pp. 2501-2511, 2003.

[22] E. Osuna, R. Freund, and F. Girosi, "Training Support Vector Machines: An Application to Face Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 130-136, 1997.

[23] Jeffrey Ng and Shaogang Gong, "Composite support vector machines for detection of faces across views and pose estimation," *Image and Vision Computing*, pp. 359-368, 2002.

[24] J. Wang and T. Tan, "A new face detection method based on shape information," *Pattern Recognition*, pp.463-471, 1999.

[25] P.S. Nair and A.T. Saunders, "Hough transform based ellipse detection algorithm," *Pattern Recognition*, pp.777-784, 1995.

[26] R.C. Gonzalez and R.E. Woods, "Digital Image Processing," *Prentice Hall*, 1992.

[27] W.F. Chen, "Face recognition using edge distance in sampling lines," 2004.