國立交通大學

資訊科學與工程研究所

碩 士 論 文

影 帶 中 特 定 人 物 的 搜 尋

Suspect Retrieval from Videos

研 究 生：黃一哲

指導教授：李錫堅 教授

中 華 民 國 九 十 五 年 九 月

影帶中特定人物的搜尋

Suspect Retrieval from Videos

研 究 生：黃一哲　　Student：Yi-Jhe Huang

指導教授：李錫堅　　Advisor：Hsi-Jian Lee

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

September 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年九月

影帶中特定人物的搜尋

學生：黃一哲　　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建立一個以特定人物為目標的搜尋系統。近年來，隨著犯罪案件的增加，警方在於找尋嫌疑犯上必須花費大量的時間與人力。因此，我們希望建立一個人物協尋系統，可以減少警方因人眼觀察比對而投入的大量人力。系統的目標是給定一個人物影像後，能自動的將在其他錄影片段中找到與此人相似的人物，而系統的主要目的為自動過濾掉非相似的人物，減少花在比對非相似人物的人力與時間。我們所提出的系統，分成四個部份：影帶中前景人物的偵測、人物的身體部位切割與特徵抽取、特徵選取和人物尋找。

第一個部分，影帶中前景人物的偵測。我們使用最簡單的方法來偵測影帶中的移動物體---連續畫面相減，不過我們做了修改，可用來抑制影子。另外，我們提出一種機制來決定影帶中移動人物的位置。然而，因為背景顏色的與前景人物的衣著顏色相似，會有偵測出的前景區域破碎的問題，所以我們使用邊緣偵測的方式，來將遺失的前景區域補回。最後，我們處理多人物平行移動的情況，來將平行移動的人物分開來。

第二個部份，人物的身體部位切割與特徵抽取。當抽取出移動人物後，為了要比對影帶中的人物與目標人物，我們需要做特徵的抽取以進行比對，然而，根據身體部位的不同，我們比對時的權重也有所不同，因此我們將人物分為三個主要部位。這一部份描述我們如何將人物切成三個身體部位，以及如何對各別的身體部位做特徵的抽取。

第三個部份，特徵選取。在前一部份，我們提出了許多不同部位的特徵，然而並非所有的特徵皆是有鑑別性的，也就是說，我們必須選出較有鑑別性的那些特徵來作為比對的依據，因此，這一部份主要介紹我們選取特徵的依據與方法，針對不同的身體部分，該選哪些特徵以及如何選取。

第四個部份，人物尋找。有了具有鑑別性的特徵後，這一部份主要介紹如何做單張影像中人物的相似度測量，其中包含同一身體部分不同特徵間的整合，及不同身體部分的相似度整合。有了人物比對的相似度定義與衡量，我們就可以從許多影帶中找出與目標人物相似的人。

實驗的部份，我們測試了實際情況中影帶中前景人物的偵測、人物的身體部位切割和人物比對。實驗結果發現我們所提出來的機制有著不錯的效果，我們可以利用這個機制來協助警方找尋特定人物。

Suspect Retrieval from Videos

Student：Yi-Jhe Huang Advisors：Dr. Hsi-Jian Lee

Department of Computer Science and Information Engineering
National Chiao Tung University

ABSTRACT

The purpose of this thesis is to construct a specific person searching system when given a target suspect image. With the increasing crimes, the police will waste much time to search and match the suspects in videos manually. In this thesis, we aim to develop a suspect searching system to decrease the manpower in matching with video suspects. We will filter dissimilar video suspects. The system consists of four stages: human detecting, human body decomposition and feature measurement, feature selection, and human searching.

In the first stage, human detection, we use the frame differencing procedure to detect moving persons in a video. We propose a modified frame difference method to suppress shadows. We also propose a mechanism to decide the suspect positions in the video sequence. Because of the color similarity of the suspect's dress and background, fragmental foreground regions may be detected. To solve this problem, we use an edge-based method to fill the lost foreground regions. Finally, connected persons are spited. In the second stage, human body decomposition and feature measurement, after detection of the moving suspect in a video sequence, we need measure feature values for suspect searching. Since the weights for matching the body

parts may differ, we decompose the human body into three parts. Because not all the features are discriminate, we need select discriminate features for human matching. Hence, we propose a mechanism to select different features in different body parts. In the fourth stage, human searching, we measure the similarity of the suspect in a video frame, including the combination of different features in a body part and the combination of the similarities in different body parts. According to difference measurements we define we can find suspects in videos.

In the experiments, we test cases of human detection, body part segmentation and human searching. The experimental results show that our system is very effective for human searching.

Acknowledgements

I am in hearty appreciation of patient discussion and proper guidance received from my advisor, Dr. Hsi-Jian Lee, not only in the course of this thesis study, but also in every aspect of my personal growth.

I think to Mr. Shen-Zheng Wang and Shan-Lung Zhao for their suggestions and encouragement. Appreciation is given to the colleagues of 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 my dissertation to them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

1.1 Motivation

With the increasing crimes, the police will waste much time to search the suspects in videos manually. In this study, we aim to develop a mechanism to search the suspect automatically or semi-automatically. Even though searching a suspect manually can reach an acceptable recognition rate, the police need to waste a lot of time on matching dissimilar target persons with the suspect.

According to reasons mentioned above, the objective of this study is to filter those dissimilar suspects. In this way, we can decrease the manpower on matching suspects.

The suspect searching system contains several modules: the detection of foreground people, the segmentation of human body parts, the judgment of face and dress type, the human searching part, and so on. The foreground person detection module can segment moving objects in videos. The module of body part segmentation is used for segmenting the body parts according to the ratios given by the suspect image. Face and dress type can also be used to match the video object with the target suspect. The module of human searching can define similar suspects and filter dissimilar ones.

1.2 Problem Definition

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

1.2.1 How to search a suspect in videos?

Given a suspect image, how can we to use the information contained in the

image to search similar person in other videos?

1.2.2 Using the information of target suspect to search similar suspects

We can extract different features from different body parts. If we can extract several discriminate features from suspects, we can use these features to find similar suspects in other video sequences.

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 CBIR (Content-Based Image Retrieval)

CBIR is the abbreviation of content-based image retrieval. It is the application of searching for digital images in large databases. The objective of CBIR systems is to find similar images from an image database according to the query image given by a user. Most systems are text-based or image-based searching. Research efforts have led to the development of methods that provide access to image and video data and we

can refer to [4]. Several famous systems are described briefly below:

- **QBIC [IBM's Query By Image Content]**
  (http://wwwqbic.almaden.ibm.com/)

- **WebSEEK [A Content-Based Image and Video Search Engine and Catalog for the Web]**
  (http://ei.cs.vt.edu/~mm/cache/WebSeek.htm)

- **MARS [**the Multimedia Analysis and Retrieval System**]**
  (http://www-db.ics.uci.edu/pages/research/mars/index.shtml)

- **Photobook**
  (http://vismod.media.mit.edu/vismod/demos/photobook/)

- **Digital Library Project**
  (http://elib.cs.berkeley.edu/)

- **PicToSeek: Combining Color and Shape Invariant Features for Image Retrieval**
  (http://www.science.uva.nl/research/isla/)

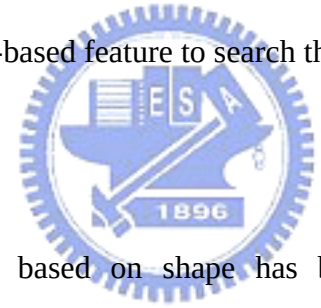
- **Video Google: A Text Retrieval Approach to Object Matching in Videos** [19]

1.3.3 Feature Extraction

Feature extraction plays an important part for the searching system. Many research studies had been proposed for features. Mainly features are divided into three types as we refer in [4].

- Color-based features

Color has been the most widely used feature in CBIR systems. It is a strong cue for retrieval of images and also is computationally least intensive. Swain and Ballard [5] used histogram intersection to match the query image and the database image. Histogram intersection is robust against these problems and they described a reprocessing method to overcome change in lighting conditions. Chahir and Chen [6] segmented the image to determine color-homogeneous objects and the spatial relationship between these regions for image retrieval. In Ref. [7], the authors presented a color clustering based approach to determine of image similarity. The above mentioned famous systems all use the color as the main feature to retrieval images and we also use color-based feature to search the suspects.



- Shape-based features

An approach to CBIR based on shape has been through use of implicit polynomials for effective representation of geometric shape structures [8]. Adoram and Lew [9] used gradient vector Mow (GVF) based active contours (snakes) to retrieve objects by shape. They note that deformable templates are highly dependent on their initialization and are unable to handle concavities. The authors present results by combining GVF snakes with invariant moments. GTunsel and Tekalp [10] defined a shape similarity based directly on the elements of the mismatch matrix derived from the eigenshape decomposition. The MARS, Photobook, PicToSeek system have used shape as feature to implement their systems. The shape feature can be used for the classification of the lower-body into the skirts and pants in our system.

- Texture-based features

The visual characteristics of homogeneous regions of real-world images are often identified as texture. Typically, textures have been found to have strong statistical and/or structural properties. The textures have been expressed using several methods. One system uses the quadrature mirror filter (QMF) representation of the textures on a quad-tree segmentation of the image [11]. Fisher's discriminant analysis is the used to determine a good discriminant function for the texture features. The Mahalanobis distance is used to classify the features. An image can be described by means of different orders of statistics of the gray values of the pixels inside a neighborhood [12]. In the Texture Photobook due to Pentand et al. [13] the authors use the model developed byLiu and Picard [14] based on the World decomposition for regular stationary stochastic processes in 2D images. The QBIC, MARS, Photobook use feature as feature to implement their system. In our system, we use texture to describe the dress content of suspects.

1.4 Assumptions

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

1. Unconstrained environments
2. Different cameras
3. Input images are color with resolution of 640×480.
4. The frame rate of videos is 30 frames/sec.

1.5 System Description

In this thesis, we develop a system to search the similar suspects when given a suspect image. The main modules include human detection, human body part

segmentation and feature measurement, feature selection, and human searching.

In the module of human detection, we introduce our proposed method to detect foreground person in videos. After obtaining the moving suspect, we segment the suspect's body into several parts and extract the features of the corresponding parts.

In the module of feature selection, we use training videos to select those discriminate features. By using our proposed model and training videos from many features, we can judge which discriminate features can be used to find the similar suspects in videos more effectively. Finally, we can use these discriminate features to search the similar suspects in videos. The system flow diagram is shown in Fig.1.5.1. The main modules of the system are described as follows:

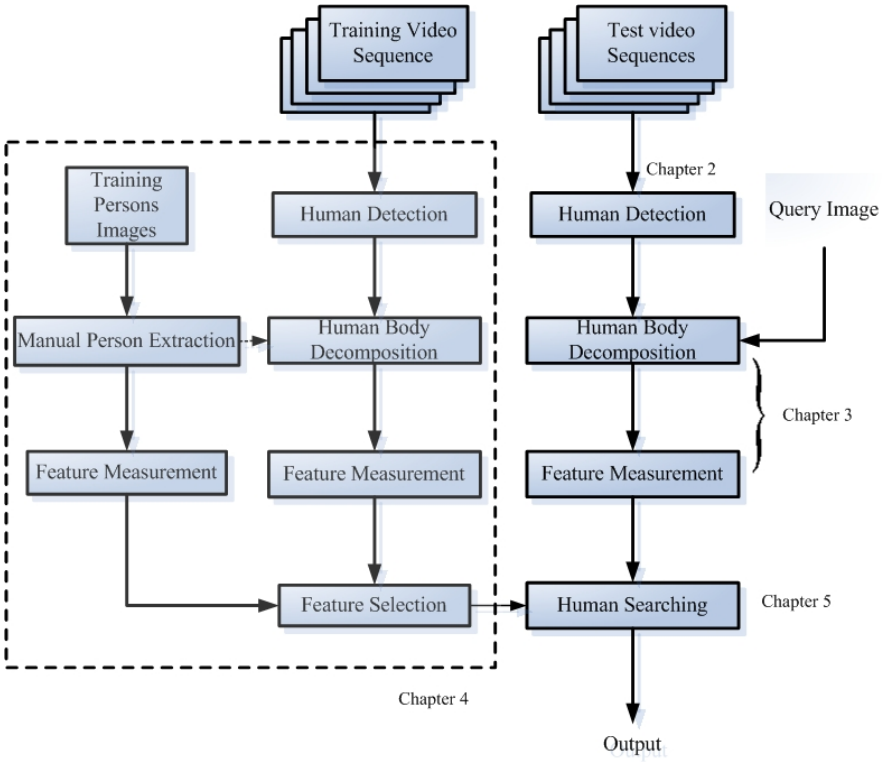


Fig.1.5.1 The system flow diagram

1.5.1 Human detection

We propose a modified frame difference method to detect the foreground person

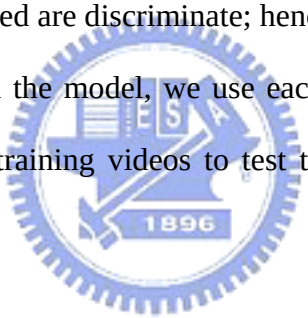
in videos. The color space we used can suppress shadows. However since the foreground person regions may be connected, we also propose a method to split the connected persons.

1.5.2 Human Body Decomposition and Feature Measurement

After the suspect in a video was extracted, we need to segment the body into several parts according to part significances. Additionally we need extract features of different body parts for human searching.

1.5.3 Feature Selection

Not all features we extracted are discriminate; hence we propose a model to select those discriminate features. In the model, we use each feature to match the training persons with the suspects in training videos to test the discriminative capability of each feature.



1.5.4 Human Searching

With the discriminate features, we determine the combination of features to measure the similarity between the video suspects and the target suspect. According to the similarities of frames, we decide which suspects are similar to the target person.

1.6 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 describes the method of human detection. Chapter 3 describes the module of human body part segmentation and feature measurement. Chapter 4 describes our approach feature selection. Chapter 5 describes the method of human searching. Chapter 6 describes

experimental results and their analyses. Finally, chapter 7 presents some conclusions and suggestions for future work.

CHAPTER 2 HUMAN DETECTION

In order to search similar suspects, we need detect the moving persons in videos. There are several existed methods to detect human. The main methods can be divided into three types: background subtraction, statistical background modeling, and frame differencing. As shown in Fig.2.1, the background subtraction method [22] can obtain foregrounds quickly. The second method is to construct the statistical background model, as Stau, et al. [2] and Elgammal, et al. [3] proposed. The results of foreground detection using statistical background models are generally acceptable. The third method, frame differencing, can be also used to detect the foregrounds for further human searching.

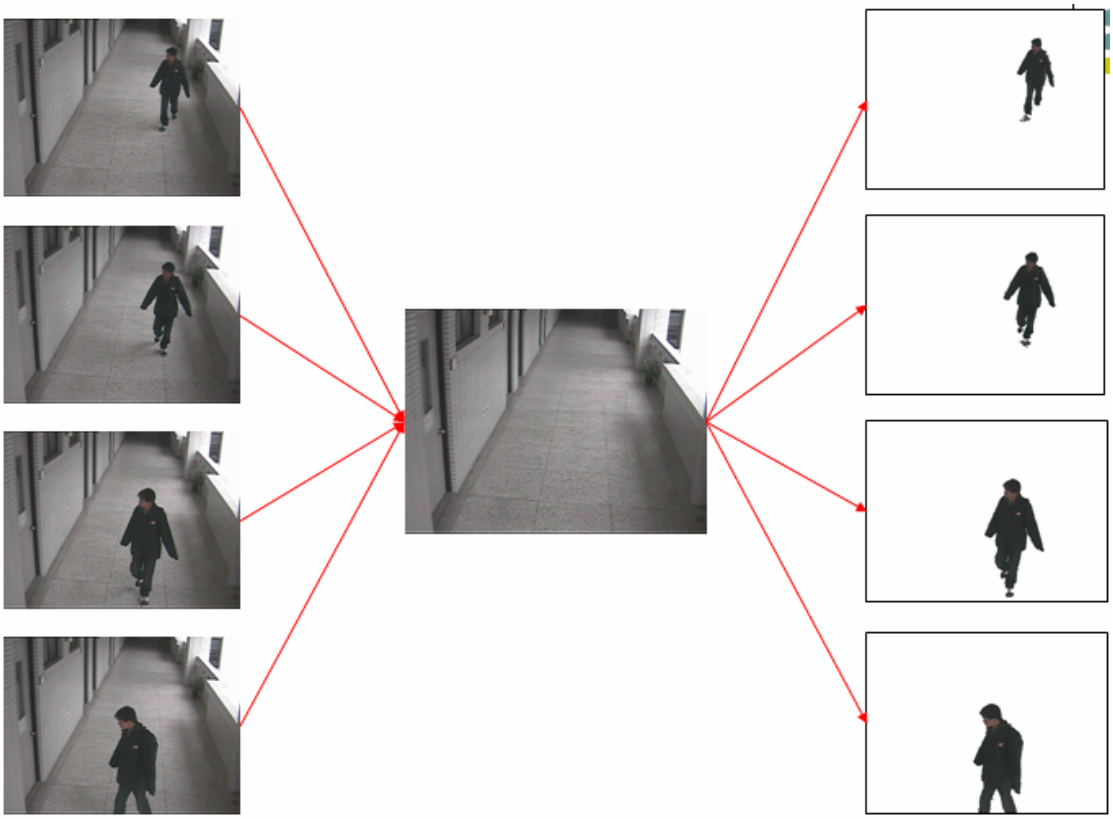


Fig. 2.1 An example showing the result after applying the background subtraction method.

However, most video recording systems only record those clips when moving objects appear in and pass by the scene of the camera. It is difficult to train the

background model. Consequently, the method of statistical background models can not be applied in our system. Moreover, when searching a similar suspect, the dubious videos were collected from unknown video recoding systems. We cannot know the scenes of backgrounds. It is hard to obtain the information of background, so we can not use the method of background subtraction to detect the moving persons. The third method, frame differencing, can be applied to find the moving regions in a video sequence. In this study, we propose a modified method of frame difference to resolve the situation. Without available information about background, we use our modified frame difference method to obtain moving persons in videos.

2.1 Foreground Detection

In this section, we describe the proposed method for foreground detection. Firstly, the moving regions in each frame can be known by using frame difference, discussed in section 2.1.1. We introduce the color space we use in Sec. 2.1.2. Moreover, we use the color space to suppress shadows from inaccurate foreground detection in 2.1.3. In Sec. 2.1.4, we obtain the foreground blobs through a four-stage process. Finally, the regions of moving suspect can be detected by our modified frame difference method, discussed in section 2.1.5.

2.1.1 Frame Differencing

Conventionally the frame differencing method subtracts the current frame from the next frame. The subtraction results are the moving regions in the current frame. The method can be modeled using the following formula:

$$if \quad |f_t(x,y) - f_{t-1}(x,y)| < threshold$$
$$g_t(x,y) = 1$$
$$else \quad g_t(x,y) = 0$$

The advantage of this method is that it can obtain moving regions in video frames very quickly. However the problem is that the regions of moving objects may be duplicated or fragmental. When a moving person passes by, if the interval between the current frame and next frame is short, the differencing regions are fragmental. Once the moving person stops, it can not obtain any moving regions. If the movement of the person is slow, the region will become small. In other words, the result is velocity dependent. In order to solve this situation, we propose a method of modified frame differencing.

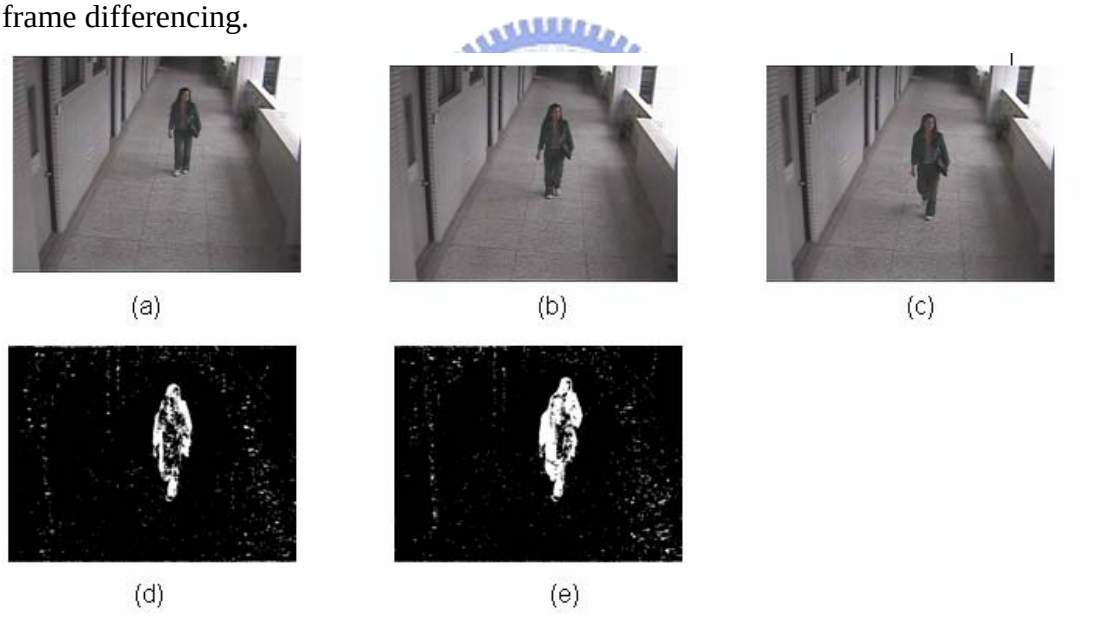


Fig. 2.1.1.1 Example of the conventional method of frame differencing.

2.1.2 Color Spaces

Here we introduce the perceptually Uniform Color System (UCS). It can be used in our system. Because the color representation is similar to the color sensitivity of human eyes and UCS implements the representation, we adopt the Farnsworth's UCS

to model our color space. To model our color space, the RGB color information is converted to CIE XYZ color domain and the chromaticity (x,y) was computed, as shown in the following formula:

$$\begin{cases} X = 0.619R + 0.177G + 0.204B \\ Y = 0.299R + 0.586G + 0.115B \\ Z = 0.000R + 0.056G + 0.944B \end{cases}, \quad \begin{cases} x = \dfrac{X}{X+Y+Z} \\ y = \dfrac{Y}{X+Y+Z} \end{cases}$$

Secondly, we convert the chromaticity (x,y) is to Farnsworth's UCS (u_f,v_f) with a nonlinear transformation mapping table. The (u_f,v_f) controls the chromaticity and the Y of CIE XYZ is used as luminance.

2.1.3 Suppression of Shadows

In some cases, when the foreground regions in video sequences were detected, shadows were also detected as foreground regions. Shadows detected as foreground regions could make further analysis inaccurate. So we need the handle this situation. The color space we used can resolve this problem.

Since chromaticity has no information about luminance, it is not sensitive to illumination changes due to shadows. In order to suppress shadows, (u_f,v_f) is used as shown in the following formulas.

$$\begin{aligned} &\textit{if} \quad \sqrt{\left(u_t(x,y) - u_{t-1}(x,y)\right)^2 + \left(v_t(x,y) - v_{t-1}(x,y)\right)^2} < \textit{threshold} \\ &\qquad g_t(x,y) = 1 \\ &\textit{else} \qquad g_t(x,y) = 0 \end{aligned}$$

.

We can obtain the effective results of suppressing shadows as shown in the figure 2.1.3.1. However some foreground regions were not be detected.

Fig. 2.1.3.1 Results of the frame difference using different methods

Usage of luminance

As shown in the above figure, the problem is that the over suppression of shadows. It caused some foreground regions were not be detected. So we propose a method to resolve this problem. The problem is caused by the similarity of chromaticity when two colors have similar chromaticity. Luminance can be used to differentiate different colors when two colors have the same chromaticity. For instance, the color of gray and white have the same chromaticity, but we also can use their luminance to differentiate them. So, a luminance constraint is added [3]:

$$r(x,y) = \frac{Y_t(x,y)}{Y_{t-1}(x,y)},$$
$$\alpha \le r(x,y) \le \beta,$$

where parameters α and β are fixed over all the images.

Here we fix the formula to model the foreground detection. After the constraint was applied, we can obtain quite effective results as shown in the Fig 2.1.3.2. Our

proposed formula is fixed, as shown in the following formula:

if $\sqrt{(u_t(x,y)-u_{t-1}(x,y))^2+(v_t(x,y)-v_{t-1}(x,y))^2} \geq$ threshold, $g_t(x, y)=1$, else
if $r(x, y)$ does not satisfy the luminance constraint, $g_t(x, y)=1$

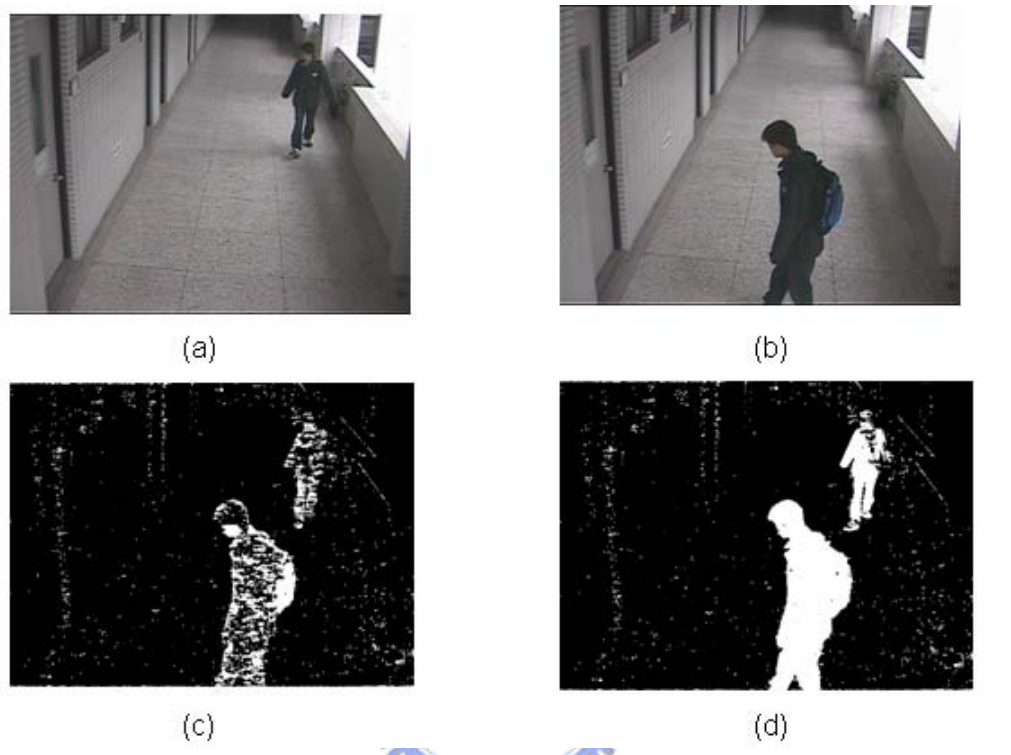


Fig. 2.1.3.2 Results of the frame difference using our method

2.1.4 Blob detection

After obtaining the results of suppressing the shadows, we need further process. The further process is to detect foreground blobs. Foreground blobs are segmented from each image by a four-stage process. As shown in the Fig. 2.1.4.1, the result of blob detection helps us to determine the foreground regions. The four stages of blob detection were listed as follows:

- noise eliminating
- morphological filtering
- connected-component extraction, and

- hole filling

Fig. 2.1.4.1 Result of blob detection

2.1.5 Modified Frame Differencing

We propose our modified frame differencing method in this section. The detected blobs are the candidates of foreground regions. We use an accumulation map to determine the foreground regions. The sub-flow diagram of this section is shown in Fig. 2.1.5.1.

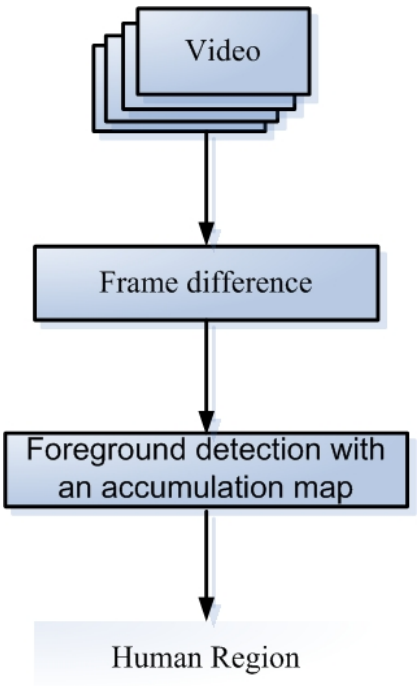


Fig. 2.1.5.1 The flow diagram of our modified frame differencing method

For a video sequence, we use the frame difference method to find the candidates of foreground regions. Then each pixel of the frame-differenced image will be decided if it is a foreground point. For a pixel (point), if the following formula was satisfied, then a point can be considered as a foreground point.

$$if \quad \frac{1}{3}n \quad \leq \quad \sum_{i=1}^{n} g_{t-i}(x,y) \leq \quad n$$

$$\text{then} \quad h_t(x,y) \in \text{foreground}$$

We use a flow diagram as shown in Fig. 2.1.5.2 to demonstrate our accumulation map mechanism. In the following, we describe how to detect the foreground region in the current frame. Firstly, we perform a frame differencing stage in a video sequence. In this stage, we select n image frames from this video sequence in every m frames. Note that the video sequence has n×m frames, but we select only n frames in a constant interval, m frames. Second, we subtract the current frame from each previous picked frame to obtain the detected blobs in the n-1 frame-differenced images. Then based on the accumulation map, we determine the foreground region in the current frame. For each frame-differenced image containing detected blobs, if a blob appears in each frame-differenced image, it is regarded as a foreground region. In summary, we construct an accumulation map to determine foreground points in the current frame of a video sequence.

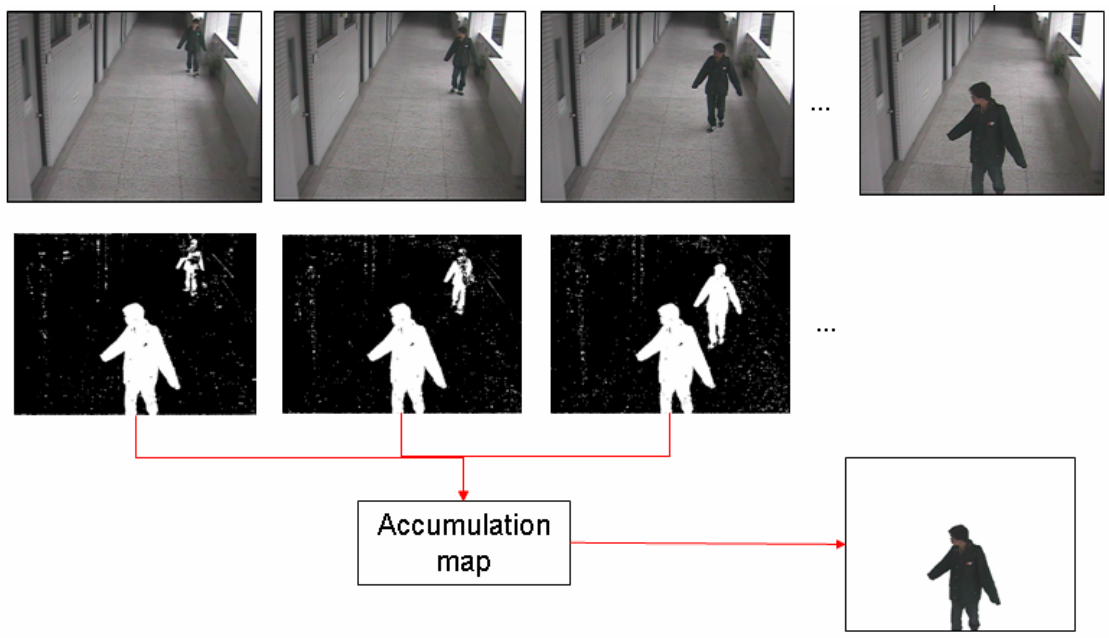


Fig. 2.1.5.2 The flow diagram of the accumulation map mechanism.

2.2 Group of Fragmental Regions

In some cases, the detected foreground regions are fragmental. This is because the color similarity between the dress of person and background pixel is too low to differentiate them. As shown in Fig. 2.2.1, we will obtain fragmental foreground regions in some situations.

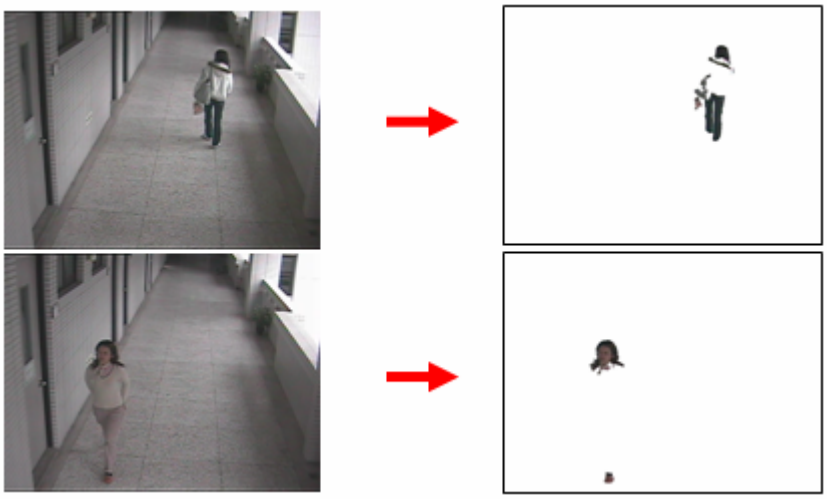


Fig. 2.2.1 Examples of fragmental regions cases.

In this section, we propose a method to solve this problem. We use the following

flow diagram to depict how we fill the fragmental foreground regions.

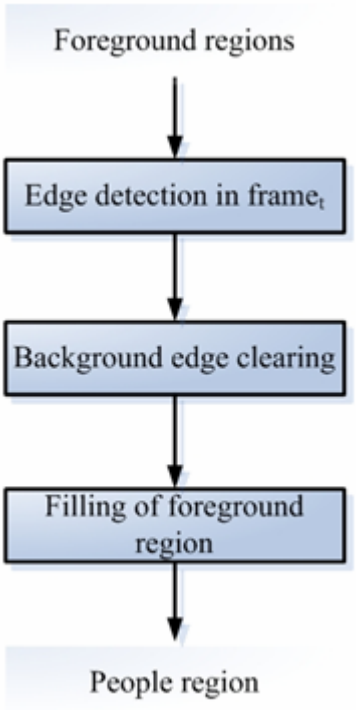


Fig. 2.2.2 The flow diagram of grouping fragmental regions.

In 2.2.1, we use edge detection to find the contour of a person. By finding the contour of the person, we can regain the missing regions. However there are too many edges to be detected when detecting edges, so we use background edges to filter those unneeded edges in subsection 2.2.2. Then we can locate the regions of missing persons. Finally, in subsection 2.2.3, we fill the fragmental regions of the missing parts to obtain the person region.

2.2.1 Edge detection

We detect edges because of the low color similarity between dress of person and background pixel. A portion of human bodies are missed by a pixel-wise differencing method (the method of modified frame differencing). However, the contour of a human can be detected by a template-based edge detector.

Fig. 2.2.1.2 The results of using pixel-wise differencing method.

The above figures show us the results of a pixel-wise differencing method in some cases of low color similarity. We will obtain the contour of the person to fill the missing parts of the body. We will use edge information [20] to obtain the contour of the person.

Because the Prewitt edge detector is more sensitive to edges than other edge detectors, we use it to detect edges. Additionally, we choose the vertical edges to locate the position of the person since most man-made buildings contain many vertical edges and human shape contains more vertical edges than horizontal ones. Fig. 2.2.1.3 shows the results of vertical and horizontal edge detectors.

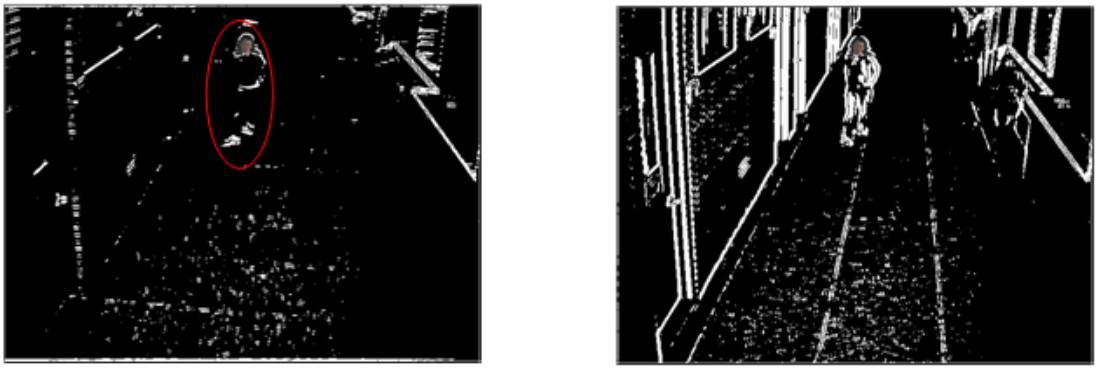


Fig. 2.2.1.3 The results of vertical and horizontal edge detections.

Although the detected results of the foreground regions are fragmental, it can be used to know the positions of the fragmental regions to be filled.

2.2.2 Background edge removal

There are many edges detected as shown in Fig. 2.2.2.1. The edges belonging to the person are our targets. The remaining edges need to be cleared, which include all non-human edges. In this section, we present our method to clear those redundant edges.

We use the background edges belong to several frames to clear the background edge of the current frame. To avoid the slow movement of the person, the interval between those frames and the time of the current frames must be long enough. In other words, if the person moves slowly and the interval is not far enough, we may clear the target edges erroneously. As shown in Fig. 2.2.2.2, we clear the background edges of the current frame by using picked frames.

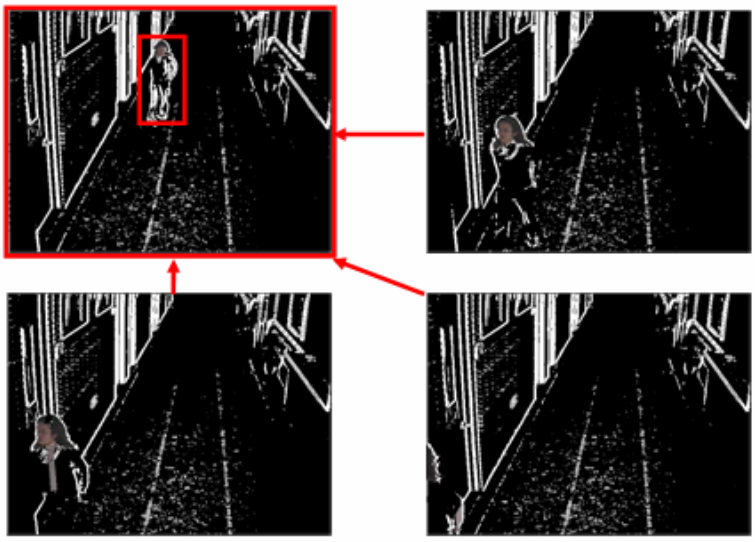


Fig. 2.2.2.1 Images showing our background edges removal method.

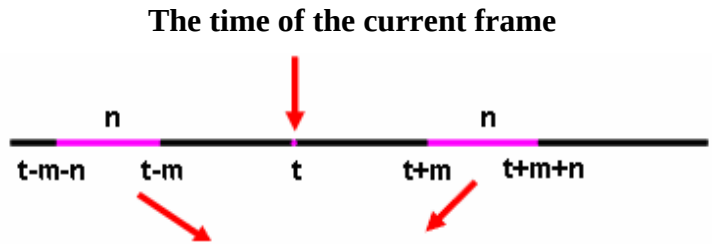


Fig. 2.2.2.2 A diagram showing the removal frames and current frame.

After clearing the background edges, the contour of person can be obtained and we will fill the lost foreground regions. The results after applying the method of this section are shown on the following figure:

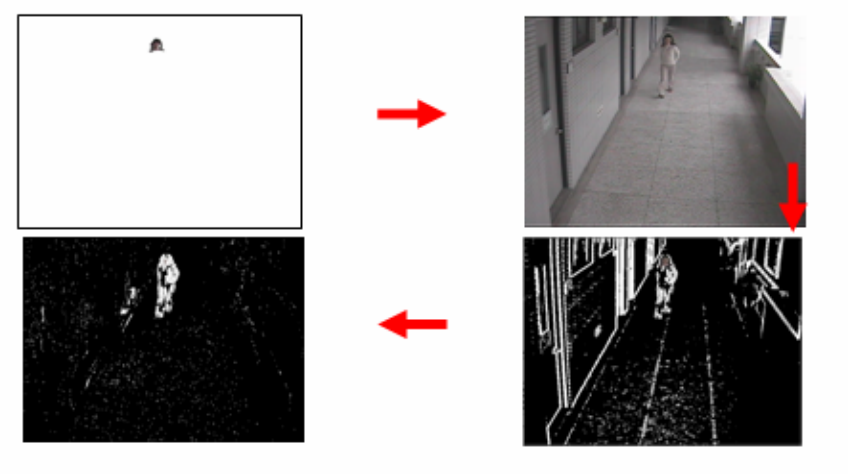


Fig. 2.2.2.3 The results of our background edges removal method.

2.2.3 Lost Foreground Filling

After the background edges were cleared, we use the information of detected foreground to locate the range of regions we want to fill. As shown in Fig. 2.2.3.1, we check a range near the person. In this range, we detect edge points between the left and right boundary of the range in each row. We then find a left and a right edge point in each row and define the foreground run as the closed interval between the left and right edge points. Finally, we fill each foreground run to obtain the region of the person.

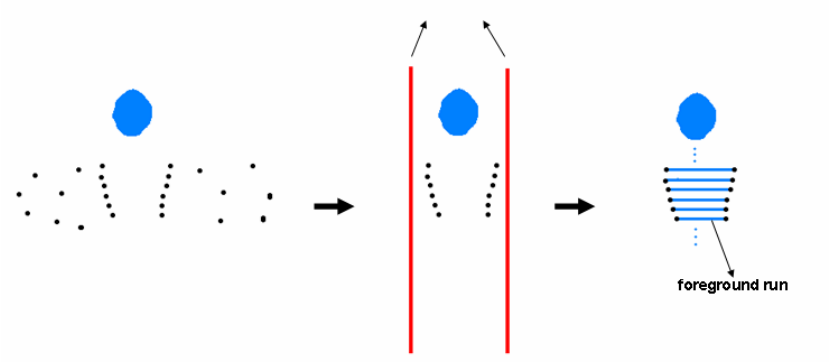


Fig. 2.2.3.1 A diagram showing our background edges removal method.

Fig. 2.2.3.2 depicts the result after using our method.

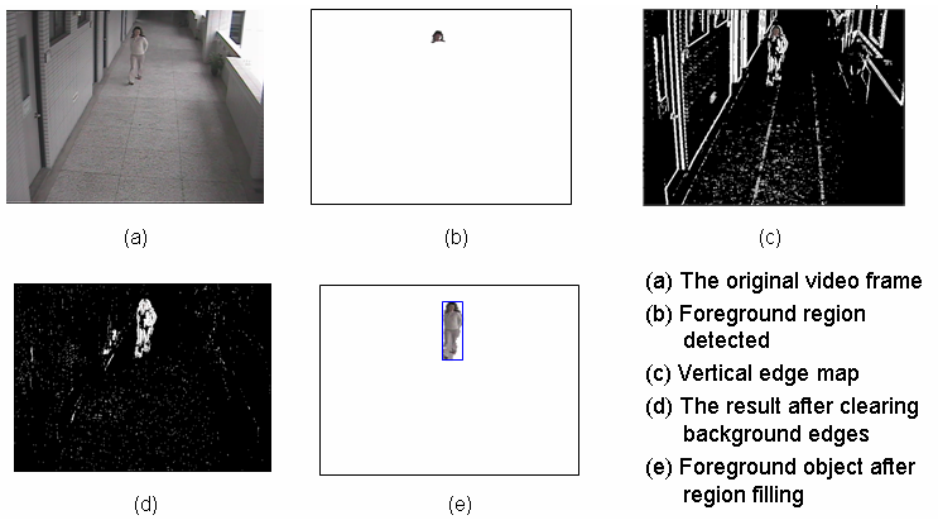


Fig. 2.2.3.2 The results of our lost foreground filling method.

2.3 Multiple Persons Segmentation

Considering the case of connect persons, the aspect ratio of the foreground region does not meet the ratio of a person. If the aspect ratio of a foreground region exceeds a threshold, then this region may contain multiple persons. As shown in Fig 2.3.1, we try to find a vertical line to split the region of multiple persons.

Fig. 2.3.1 An example of connected persons case.

In this section we present a method to split the detected region of connected persons. The following diagram shows how we segment a region of connected persons.

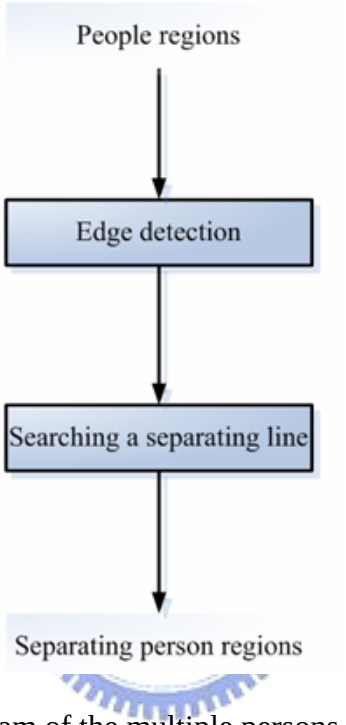


Fig. 2.3.2 The flow diagram of the multiple persons segmentation algorithm.

Here we introduce our method to split the region of the connected persons briefly. Firstly, we define a vertical scanning line and find the distance between the top and bottom edge points in the edge map for each vertical scanning line. Second, we define the separating line as the vertical scanning line with the shortest distance. As shown in Fig. 2.3.3, the vertical blue line is the separating line of this edge map.

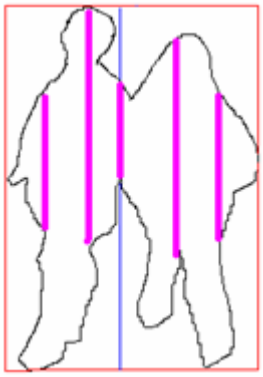


Fig. 2.3.3 A diagram to show the separating line.

CHAPTER 3 HUMAN BODY DECOMPOSITION AND FEATURE MEASUREMENT

After detecting the suspect in the videos, the next step is to decompose the body of the suspect and to extract the features of the suspect. In this chapter, we use our method to segment the body of the suspect into three parts because the different parts have different weights. When the query suspect was given, the user can assign different weights to different body parts. Sec. 3.1 introduces how to decompose the body of the suspect into three parts. Then we propose a method to extract the features in different body parts in Secs. 3.2, 3.3 and 3.4.

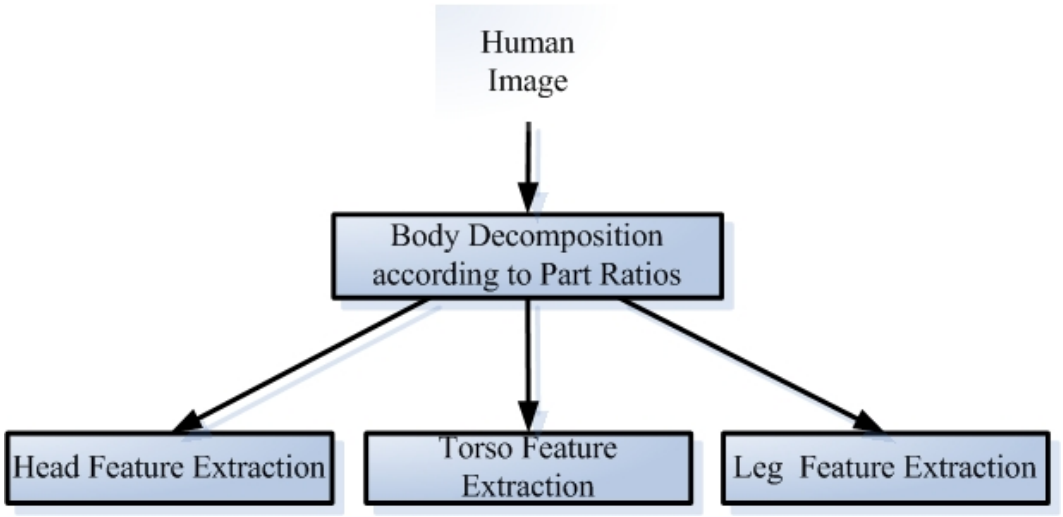


Fig. 3.1 The diagram of human body decomposition and feature measurement.

3.1 Body Part Decomposition

In this section, we introduce our method to decompose the body of the detected person. When the query suspect was given, the user can manually divide the body of the query suspect into three parts and obtain the ratio of the three parts. We then decompose the bodies of the suspects in a video according to the three ratios of the parts in the query suspect. As shown in Fig. 3.1.1, we decompose the body of detected

person into three parts: head part, upper-body part and lower-body part.

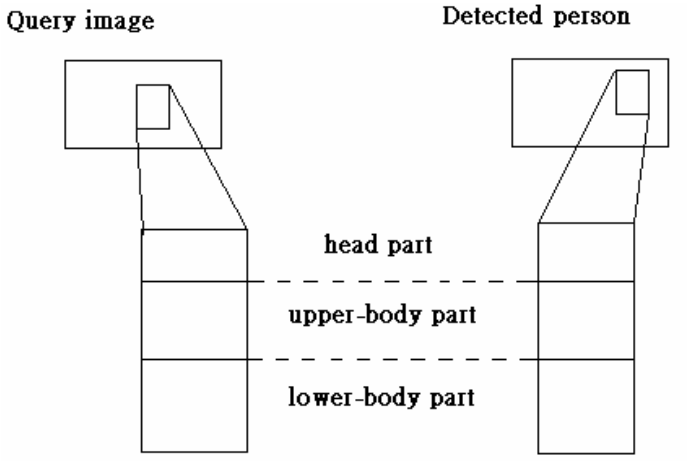


Fig. 3.1.1 A diagram showing body part decomposition.

3.1.1 Coarse Decomposition

When a query suspect was given, we can decompose the human body interactively and obtain the ratio of the parts according to the part ratio of the query suspect.

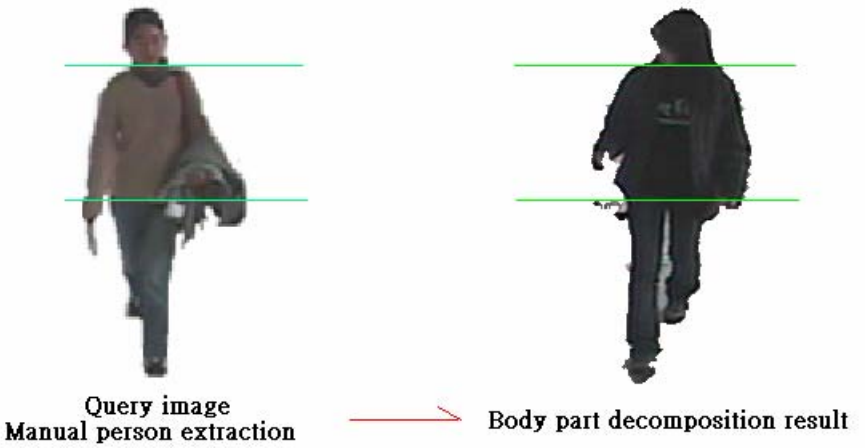


Fig. 3.1.1.1 An example of coarse decomposition of body parts.

3.1.2 Part Adjustment

After segmenting the video suspects into three parts coarsely, we need to do part adjustment. This is because the body ratios of the video suspects may different from

the body ratios of the query suspect. A part after segmentation may contain a portion of other parts, so part adjustment is needed. As shown in Fig. 3.1.2.1, the upper and lower splitting line will be adjusted.

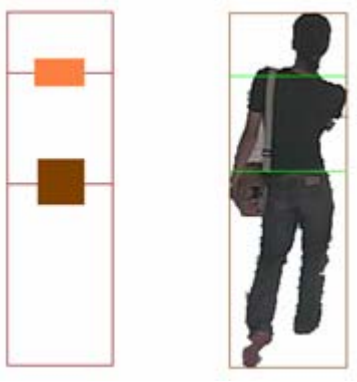


Fig. 3.1.2.1 An example for adjusting body parts.

For the upper splitting line, we detect the skin colors in the orange regions as the shown in Fig. 3.1.2. The upper splitting line will be adjusted to the lower boundary of the detected skin region in the orange region of the person. The adjustment is according to the fact that the regions with skin colors are a head part of the person. Finally, we clear the non-face regions in the head part by filtering the pixels of non-hair and non-skin color.

For the lower splitting line, we detect the heavy intensity of edges in the brown region in Fig. 3.1.2.1. The lower splitting line will then be adjusted to the position with the heaviest edge pixels. As shown in 3.1.2.2, we detect the position with the most horizontal edge points in the region of the mid 80 percent of the body region, because we want to remove other carrying objects. The results after the adjustment is shown in Fig. 3.1.2.3.

Detection of the heaviest edge points in the region of the mid 80 percent of the body region.

Fig. 3.1.2.2 An example showing the detection of the heaviest edge points.

Fig. 3.1.2.3 The results of body part decomposition.

3.2 Color Feature

The color feature is the main feature when we compare two suspects. It can be used for represent the dress content of a person. For human searching, the skin color and hair color are used as major features. We use the skin color [21] and hair color [16] to detect the faces of suspects from video.

3.2.1 Skin and Hair Detection

Here we describe the range of skin colors and hair colors when detecting skin and hair. As shown in Fig. 3.2.1.1, we list the range and the detected results of skin colors and hair colors.

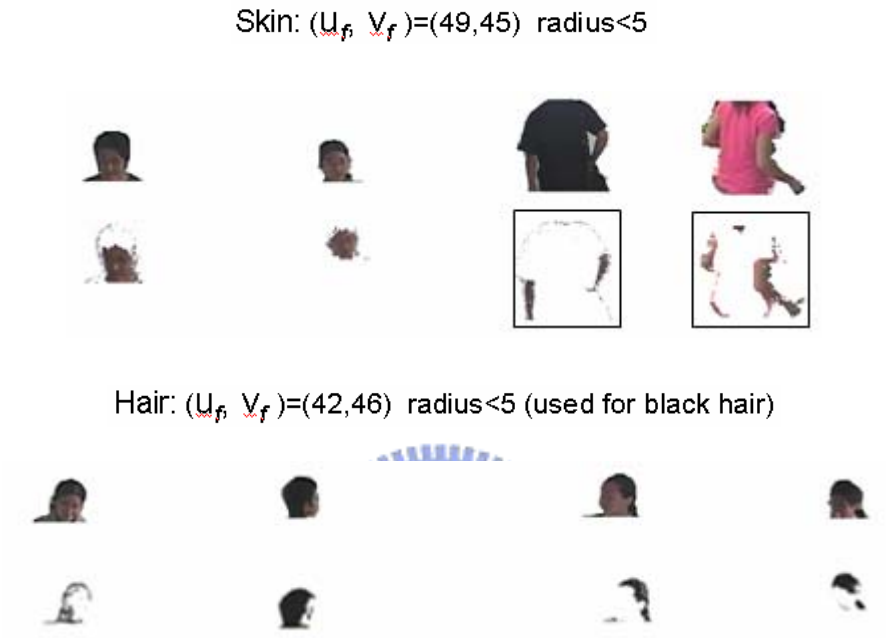


Fig. 3.2.1.1 The results of detected skin and hair regions.

1. Projection Profiles of skin colors

We use the projection of skin colors as our features. The vertical and horizontal projections of skin colors are used for comparing the similarity of two suspects.

Fig. 3.2.1.2 Examples of projections of skin colors

2. Hair Projection

The feature of hair types is usable for judging the face direction and the gender [17] of one person. When two persons have the same face directions, we can compare their head to judge if they are the same person. In this study, we use the projection of hair colors as our features. The vertical and horizontal projections of hair colors are used for comparing the similarity of two suspects.

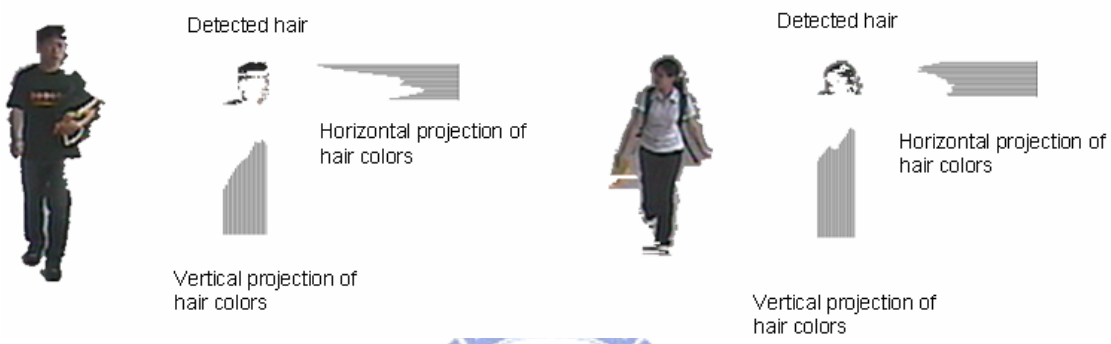


Fig. 3.2.1.3 Examples of projections of hair colors.

3. Color ratio

Color ratio contains two types: the skin and hair color ratio. The skin color ratio is used to determine the face view (front or back view) and the dress type of the clothes and pants. The hair color ratio describes a distribution of the hair type of the person. The skin and hair color ratio are used as face features when comparing the heads of two persons.

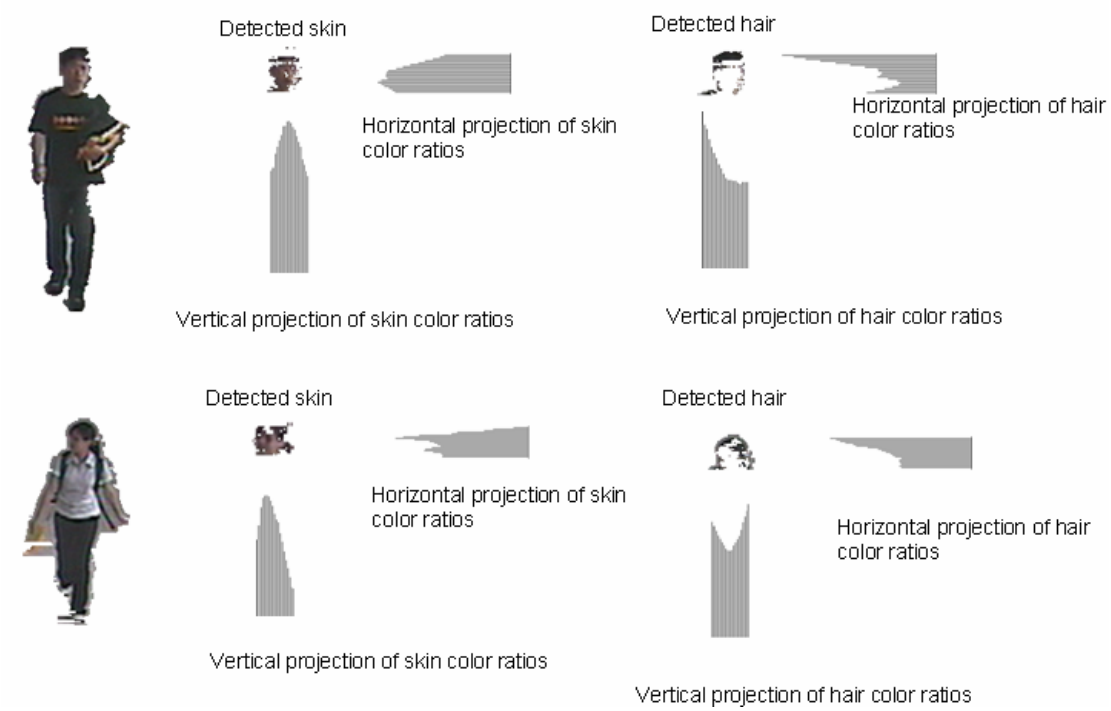


Fig. 3.2.1.4 Examples of projections of skin and hair color ratios

A face view can be classify into the front and back views. As shown in Fig. 3.2.1.5, we use the face mask to determine the view of a face. We count the skin color ratio in the orange region of the face mask. If the ratio is greater than 0.11, then we judge a face is front view.

Fig. 3.2.1.5 Examples of used face mask

3.2.2 Color histogram

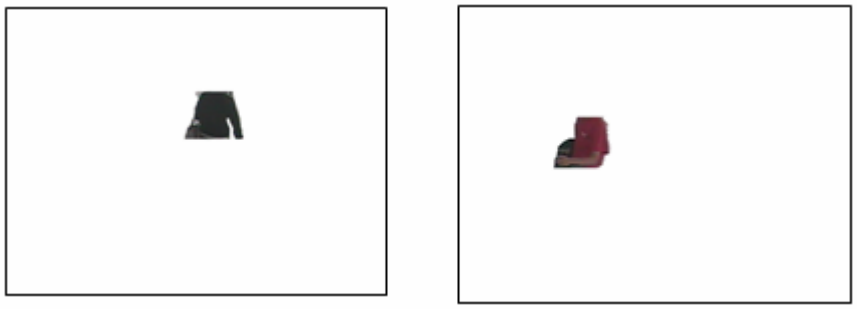


Fig. 3.2.2.1 Examples of different colors of upper-bodies

The color information is the major conspicuous characteristic when comparing two suspects in the upper-body and lower-body part. We describe how we compare the dresses of two suspects here. When comparing dresses of two suspects, we compare their chromaticity of their dresses. In the color space, we divide the u_f and v_f into equal parts and compare the u_f and v_f histogram distributions individually. The u_f was divide into nine parts and the v_f was divided into fourteen parts. Finally, we sum the measured distances of u_f and v_f as the color comparison distance.

3.3 Texture Feature

As shown in Fig. 3.3.1, the content of the dress of a person is another characteristic to discriminate different people. The contents of a dress can be described by the edge distribution. The edge distribution is also a kind of texture representation. In this study, we use the gradient (edge) as the texture feature when comparing the dresses of two suspects.

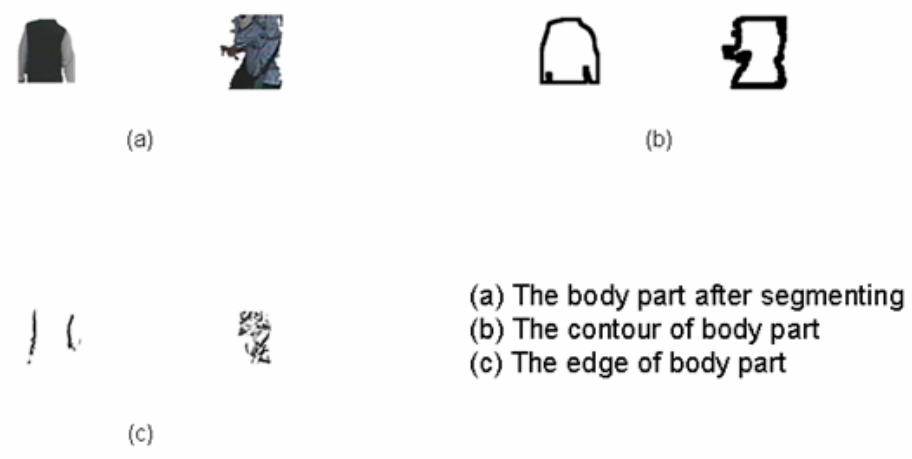


Fig. 3.3.1 Examples of different textures of upper-bodies.

The projection of detected edge pixels can be used as the features of upper-body and lower-body. By comparing the gradient distributions of two persons, we can discriminate the different persons. The projection of edge pixel ratios is also used to describe the edge pixel ratio distribution of the dress and pants on a suspect. Both the edges and edge ratios can be used to describe the dress style of a person. The vertical and horizontal projections are used as the texture features.

3.4 Features used in Body Parts

In this section, we summarize the used features in the three different body parts. In the head, we divide the face into the front and back views. Because the low resolutions of video recording systems, it is hard to recognize the face [18] of a suspect. As a result of this problem of impossible recognition of faces, we just compare the hair type. Before determining the hair type of a person, we need judge whether a head is in a front or back view. The hair type can be divided into two types: the long and short hair. The back view is not our targets when matching two persons. Thus we divide the hair type into long hair type and short hair type only in the front

view cases.

For the features used in the head, using the face mask, we can determine the view of a face (front or back view), and then judge the hair type of a person (long or short hair). The projection of the skin colors and hair colors can be used as the features in the head. The projection of the skin color ratios and hair color ratios are also used as the head features.

For the upper-body and the lower-body part, we use the same features. The used features include the color histogram, the skin color ratio and the edge projection profiles. The color histogram is used to compare the dress colors of two suspects. With the estimation of skin color ratio, we can determine the dress type of the clothes and pants (short or long type). At last, we use the projection profiles of the edges and edge ratios as the features in both upper-body and lower-body. The following figures summarize the features used in different body parts:

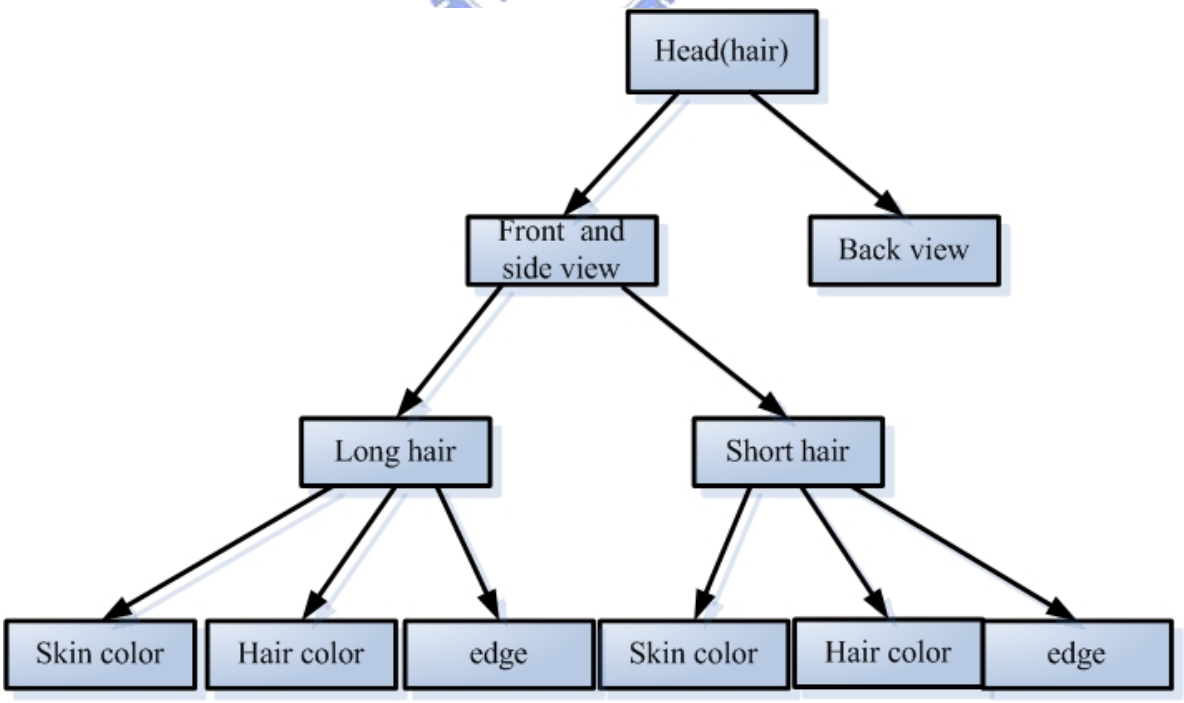


Fig. 3.4.1 The features used in the head.

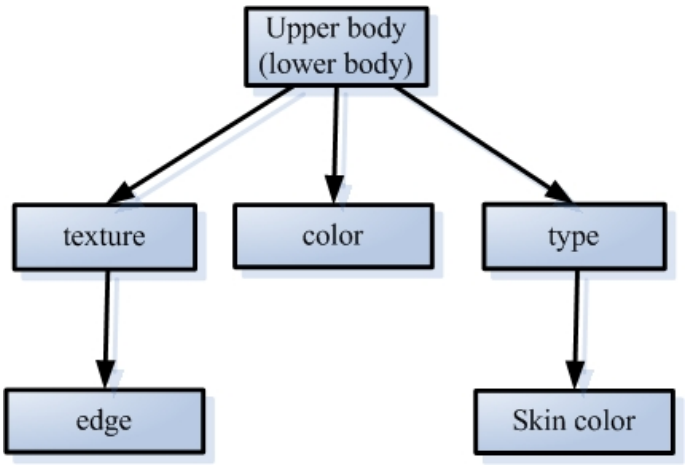


Fig. 3.4.1 The features used in the head.

CHAPTER 4 FEATURE SELECTION

After defining the features we will use in different parts, we need to select some of them to search a suspect in a given video because not all the features are have enough discriminant capability. In this chapter, we propose a model to determine which features are discriminant and will be used to search suspects. We depict our mechanism in the following figures.

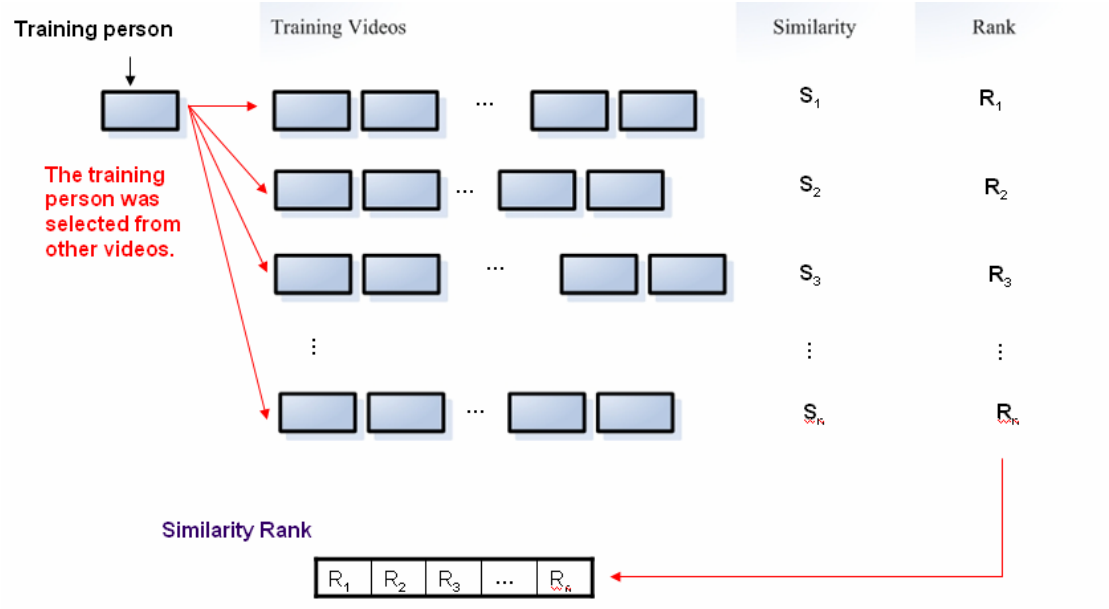


Fig. 4.1.1 The similarity rank for a training person.

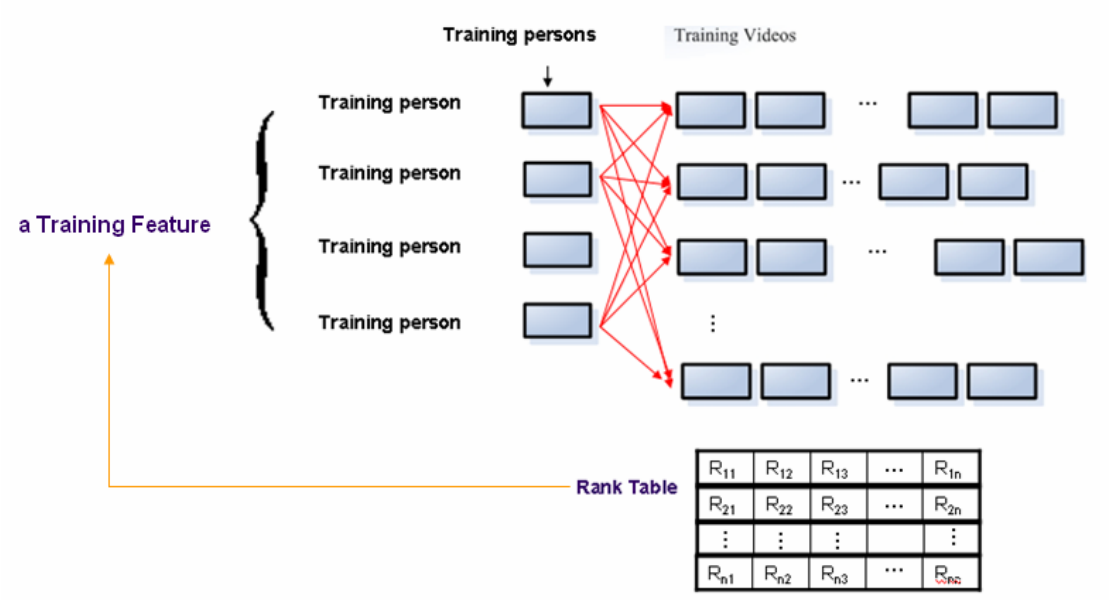


Fig. 4.1.2 The rank table for a training feature.

To test the discriminative capability of a feature, we use *m* training videos and select *n* training persons from other videos. These *n* training persons must appear in the *m* training videos, where *n*<=*m*. We use *n* training persons to test if a feature can be used to find the person in the training videos.

When testing a feature, suppose that the selected training person is the suspect in video *x*. If the test feature has a good discriminative capability, then the similarity measurement of the video x must be highest among all training videos when using this feature to measure the similarities between the training person and each training video.

For a training person, we measure the similarity measurements between all *m* training videos and the selected training person. We have *m* similarity measurements and then sort them. The result of sorting ranks can be record in a vector whose size is 1 by *m*. Now we have *n* training persons. It means we have *n* vectors. The results of *n* vectors can be recorded in a table whose size is *n* by *m*. If the feature has nice

discriminative capability, then the records of similarity rank on the diagonal entities will all be 1's, because each training person can be found the corresponding training video by using the test feature. In other words, each corresponding training video will have the highest similarity rank among all training videos. Accordingly, if the feature is discriminative excellently, then the sum value of the diagonal entities in the rank table which belong to this feature will be *n*.

Consequently, we can obtain the conclusion: for a feature, the lesser the value of rank-sum of one feature has, the more discriminative capability it has.

4.1 Similarity Measurement

The histogram difference measurements can be measured by either the absolute distance or the Bhattacharyya distance. The formulas of the two distances are given as follows:

$$A(P \| Q) = \sum_{i=1}^{n} |p_i - q_i|$$

$$B(P \| Q) = \sum_{i=1}^{n} \sqrt{p_i q_i},$$

where the *pi* and *qi* is the corresponding bins when comparing two histogram.

4.2 Similarity of a Video

In order to determine the discriminant features, we need to define the similarity of a video. If we have similarity measurements of *m* training videos, then we can know the similarity rank and test a feature if is discriminant. After defining the similarity of a video, we can compare the similarity rank among all training videos.

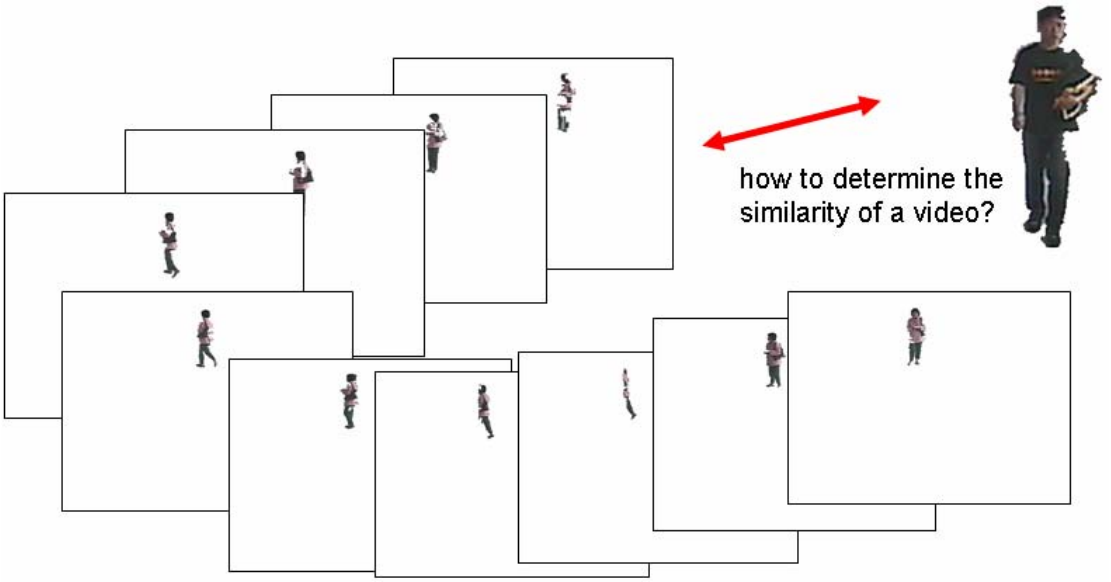


Fig. 4.2.1 We need to determine the similarity of a video.

4.2.1 Frame Similarity

When a training person was compared to each training videos, for each training video, it is composed of a corresponding frame sequence. First, we need to record each similarity measurement of each frame in this training video sequence. As shown in Fig. 4.2.1.1, postures of a moving person will change as time goes by. We compare each video frame with the suspect image and record this similarity measurement. Then we can determine the similarity of each video.

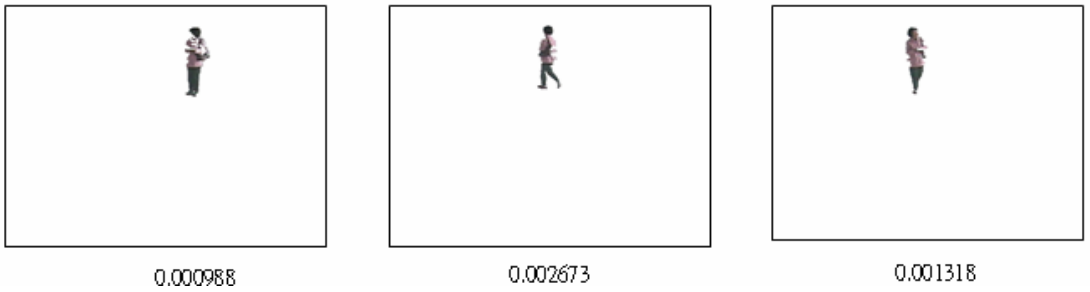


Fig. 4.2.1.1 Examples of recording similarity measurements

4.2.2 Similarity Determination

Because the similarity rank of each training video is needed to be determined when testing a feature, we define the similarity of training video and then we can compare the similarity measurements to get the similarity rank. The similarity of a video is determined when we take the mean of the top 3 measurements in all frames as the similarity of a video.

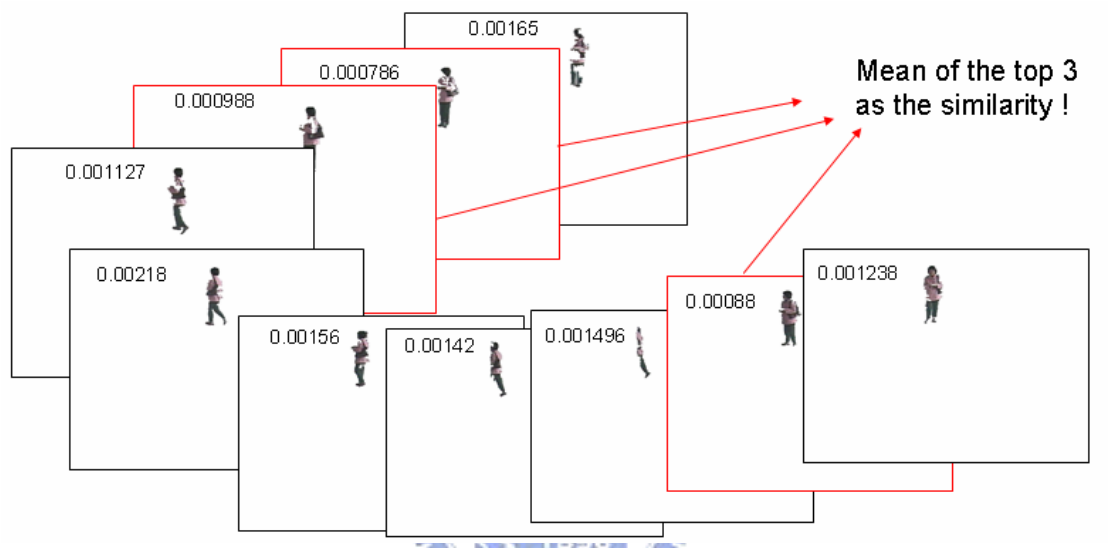


Fig. 4.2.2.1 An example of determining the similarity of a video

4.3 Similarity Rank

After determining the similarity of a training video, by comparing each similarity of each training video, we can obtain the similarity rank of each video. The similarity rank is used to determine the discriminant capability of a feature. In this section, we propose a rank-sum model to select those discriminant features. In subsection 4.3.1, we introduce the usage of similarity rank and the similarity rank-sum model in all features. In 4.3.2, we use the feature selection model to select those discriminant features. Finally, we show the results of discriminant features which will be used for suspect searching.

4.3.1 Similarity Rank-sum Model

The similarity rank is used to test the discriminative ability of a feature. First we select a training person in other videos and this training person is needed to appear in one of the training videos. The reason is that we aim to use the test feature to find the corresponding video suspect. For a training person, if the test feature has quite discriminative capability, we can use this feature to find the corresponding suspect and the corresponding video will have the highest similarity rank. We measure the similarity measurements between the training person and all training videos by using this test feature. Subsequently, each training video will have a corresponding similarity measurement and we can obtain the similarity rank for each training video. Finally we record the similarity rank with a vector whose size is 1 by *m.*

For *n* training persons, we use the same way to find the similarity rank of all training videos. The size of similarity rank vector for one training person is 1by *m*, and the size of all similarity rank vectors for *n* training person is *n* by *m*. We collect all the similarity rank vectors to obtain a similarity rank table (size: $n \times m$) which record each similarity rank after comparing the *n* training persons with the *m* training videos.

The rank-sum model is implemented by summing all the values of diagonal entities in the rank table of a test feature. If the value of the summation is *m*, it means the test feature is an excellent discriminant feature. This is because we have *m* training videos and the best case is that each corresponding training video was found. In other words, all the values of diagonal entities in the rank table are all 1s.

4.3.2 Feature Selection Model

The feature selection model succeeds by the mechanism of the rank-sum,

introduced in the previous section. When we want to select those discriminant features, we just compare their rank-sum values. The less value of rank-sum is, the more discriminative capability the test feature has.

When selecting the discriminative features from many features, we set a threshold value for the rank-sum mechanism. For a feature, if the rank-sum value is less than the threshold, it will be selected as a discriminative feature. By setting a threshold, we compare each rank-sum value with this threshold and then determine which features are discriminative. This is the feature selection model and we use it to select the discriminative features for further suspect searching.

4.3.3 Feature Selected

We adopt the absolute distances to measure the feature differences. In the following, we show the results of our feature selection model.

1. Head
    - Vertical projection of hair percentage
    - Vertical projection of skin/hair percentage
    - Skin percentage when a face is divided into four quadrants (left-top, right-top, left-down, right-down)
    - Hair percentage when a face is divided into four quadrants (left-top, right-top, left-down, right-down)
    - Color histogram
2. Upper-body

- Projection of edge
- Projection of edge ratio
- Vertical projection of edge ratio
- Color histogram

3. Lower-body
   - Projection of edge
   - Horizontal projection of edge
   - Horizontal projection of edge ratio
   - Color histogram

CHAPTER 5 HUMAN SEARCHING

With the discriminative features, we can use these features to compare the similarities between the query suspect and the video suspects. In this chapter, we explain how to combine these discriminative features in different body parts.

Although we divide the human body into three parts, the measurement of head is hard to use. The feature in the head contains only the hair type. We can compare the face similarities between the query suspect and the video suspect when they have the same heads direction. The comparison of face similarity will increase the searching performance of our system. As shown in Fig. 5.1, we show the flow of our system.

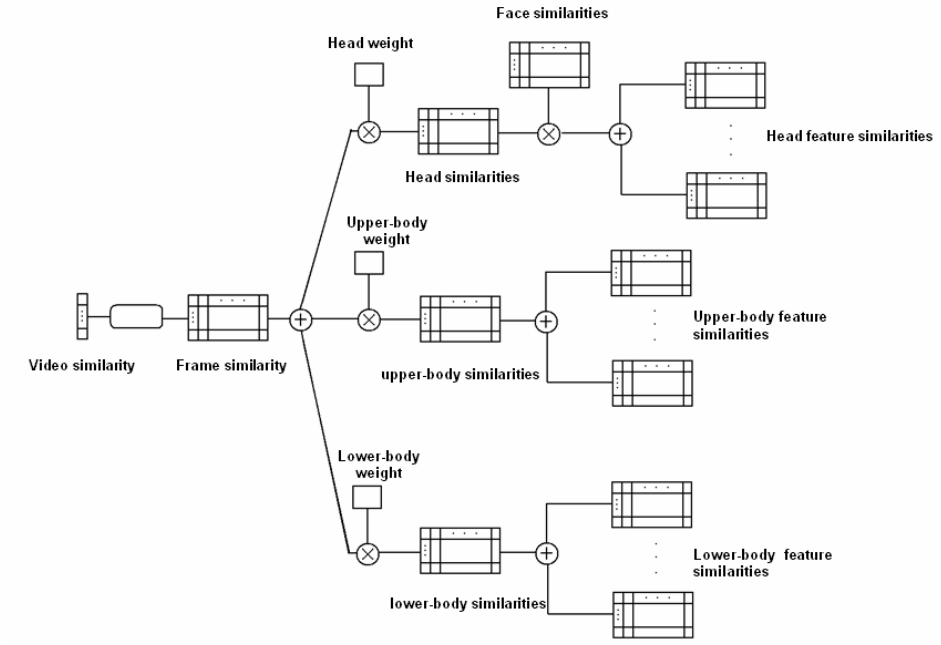


Fig. 5.1 The flow diagram of our searching system.

5.1 Judgment of Head Direction

In order to determine whether the face directions of two persons are the same, we have to judge the view of the faces. We use the face mask to determine the view of their faces (front or back). If they both have the front views, then we judge whether their faces have the same direction.

We divide the face directions into two types: front and side. In the following we describe how to judge the two types. If the vertical projection of skin color ratios is symmetrical, then the face is taken as front. If the vertical projection of skin colors ratios skew to one side and the vertical projection of hair colors skew to another side, then the face is regarded as side. The judgments are shown in the following figures:

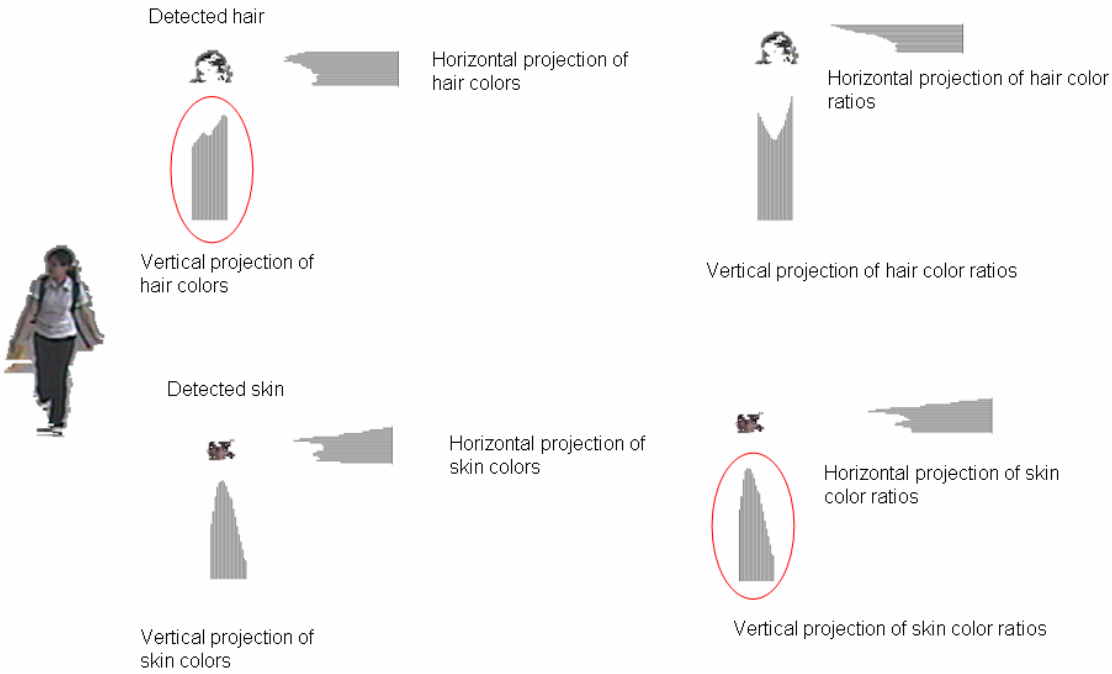


Fig. 5.1.1 The judgment of the side face.

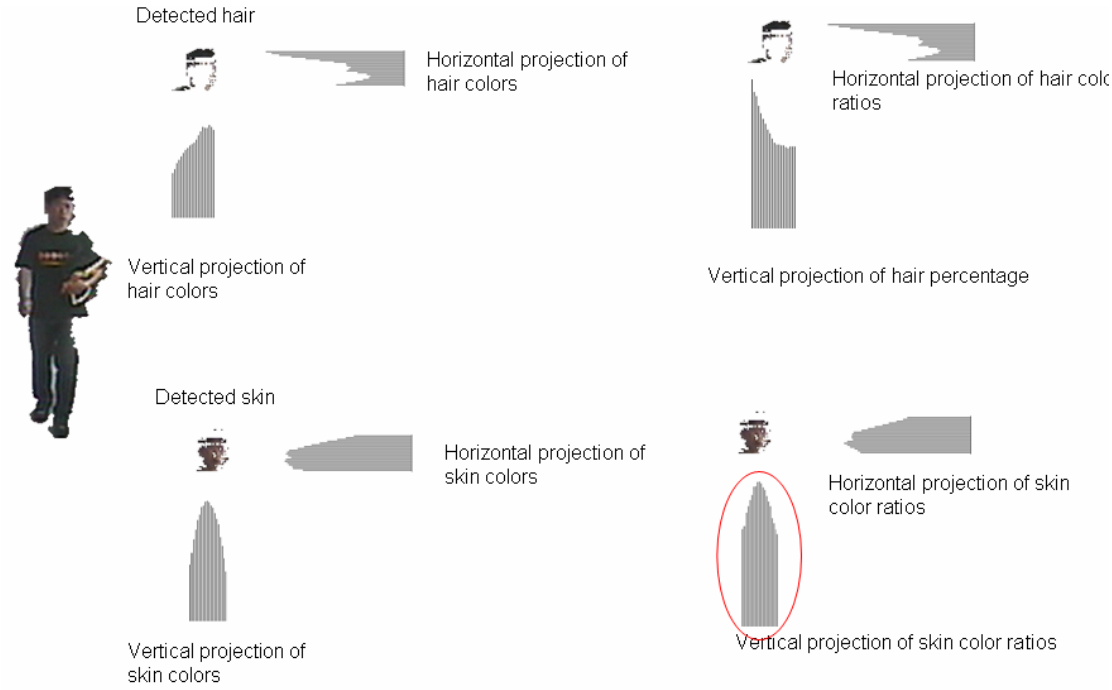


Fig. 5.1.2 The judgment of the frontal face.

5.2 Body Similarity

According to different body parts, we have different numbers of features. We use the equal weights for the selected features in the same body part. As shown in the following formulas, the different body part similarity measurement are defined, where the weights were assign to 1.

$$d_h = \sum_{i=1}^{5} w_{hi} \times d_{hi}$$

$$d_u = \sum_{i=1}^{4} w_{ui} \times d_{ui}$$

$$d_l = \sum_{i=1}^{4} w_{li} \times d_{li}$$

5.3 Frame Similarity

After defining the different body similarities, we can combine the three similarities of different body parts to obtain the similarity of a suspect. Consequently,

for a video sequence, a frame similarity exists in each frame for the suspect of the video sequence. The following formula is used to decide the frame similarity, where w_h, w_u and w_l are the different weights in head, upper-body, lower-body part, respectively, and they can be defined interactively by the user when the user inputs a query suspect image. Once the frame similarity was computed, we compare the similarity measurement to a predefined threshold. If the measurement is less than the threshold, the video may contain a suspect. In this way, we can judge whether the video suspect is similar to the target suspect.

$$f_d = w_h \times d_h + w_u \times d_u + w_l \times d_l$$

CHAPTER 6 EXPERIMENTAL RESULTS AND DISCUSSION

In this chapter, we present the experiment results and the discussion of our system. The proposed approach has been implemented in a personal computer with Pentium IV 3GHz CPU and 1G RAM. The software environment is Microsoft Windows XP and Microsoft Visual C++ 6.0

The input images are color, whose resolutions are 640 x 480. We separate our experiments into five parts: human detection, segmentation of body parts, front or back view, face direction, and human searching.

6.1 Experiment Result

6.1.1 Human detection

We test five different cameras as shown in Fig. 6.1.1.1.

Fig. 6.1.1.1 The five experimental scenes

We test 135 videos (1842 frames). 1675 frames images are extracted correctly.

The accuracy can reach 90.93%. Some results were shown as follows:

Fig. 6.1.1.2 Results of some detected persons.

6.1.2 The segmentation of body parts

We test 135 videos (1842 frames). 1568 frames images are extracted correctly. The accuracy can reach 85.12%.

6.1.3 Front or back view

We test 72 videos (1016 frames). 985 frames images are extracted correctly. The accuracy can reach 96.94%.

6.1.4 Face direction

We test 679 frames. 635 frames images are extracted correctly. The accuracy can

reach 93.51%.

6.1.5 Human searching

For human searching, we collect 40 videos from different four cameras and test the performance. For each camera, we collect 10 videos individually for testing. The experiments were divided into three parts: close test, accuracy rate, and false alarm. There are 40 target images in the close test. In this part, the 40 target images were selected from 40 videos, in which each video contains a target image. Secondly, there are 80 test images for testing accuracy rate. For each camera, we use 20 test images individually. However the 80 target images were selected from other videos. In the third part, we test 80 target images from other videos different from the suspects in the 40 videos. The results were shown as follows:

1. Close test

In this part, we can reach the accuracy rate of 100%.

2. Accuracy rate

In this part, we test the accuracy rate of our system by testing 5, 10, 15 and 20 target persons in each camera. In order to test the performance of adjusting different body weights, we test four situations.

- The weights (0.3, 0.4, 0.4), where the values were represented for the head, upper-body, lower-body weights individually.
- weights (0.3, 0.4, 0.4) and the addition of the face similarity
- weights (0.1, 0.5, 0.4)

- weights (0.1, 0.5, 0.4) and the addition of the face similarity

The result showed us that the accuracy rate can exceed 80%.

	Video1	Video2	Video3	Video4	Ave.
5	4/5	5/5	4/5	4/5	85%
10	8/10	9/10	8/10	7/10	80%
15	13/15	14/15	12/15	12/15	85%
20	17/20	18/20	17/20	16/20	85%

Table 6.1.5.1 The accuracy rate with weights (0.3, 0.4, 0.4).

	Video1	Video2	Video3	Video4	Ave.
5	4/5	5/5	5/5	4/5	90%
10	8/10	9/10	8/10	8/10	82.5%
15	13/15	14/15	12/15	13/15	86.7%
20	17/20	18/20	18/20	17/20	87.5%

Table 6.1.5.2 The accuracy rate with weights (0.3, 0.4, 0.4) and the addition of the face similarity.

	Video1	Video2	Video3	Video4	Ave.
5	4/5	5/5	4/5	4/5	85%
10	8/10	9/10	8/10	7/10	80%
15	13/15	14/15	12/15	13/15	86.7%
20	18/20	18/20	16/20	16/20	85%

Table 6.1.5.3 The accuracy rate with weights (0.1, 0.5, 0.4).

	Video1	Video2	Video3	Video4	Ave.
5	4/5	5/5	4/5	4/5	85%
10	8/10	9/10	8/10	7/10	80%
15	13/15	14/15	12/15	13/15	86.7%
20	18/20	18/20	17/20	16/20	86.25%

Table 6.1.5.4 The accuracy rate with weights (0.1, 0.5, 0.4) and the addition of face similarity.

3. False positive

In this part, we test the false positive of our system by testing 20 target persons in each camera. In order to test the effects of face similarity and different body weights, we test four situations:

- The weights (0.3, 0.4, 0.4)
- weights (0.3, 0.4, 0.4) and the addition of the face similarity

- weights (0.1, 0.5, 0.4)
- weights (0.1, 0.5 0.4) and the addition of the face similarity

# of false positive	Video 1		Video 2		Video 3		Video 4	
Never appear	1	1.6	2	1.5	1	1.5	1	1.3
	0		1		2		2	
	1		0		2		1	
	2		1		1		0	
	2		2		1		0	
	1		0		2		1	
	2		1		0		2	
	1		2		1		1	
	2		1		2		1	
	2		2		1		2	
	2		2		1		2	
	1		2		2		1	
	3		2		1		2	
	1		3		3		1	
	2		1		2		2	
	3		2		1		2	
	2		1		2		1	
	2		2		1		1	
	1		1		2		2	
	2		2		2		1	

Table 6.1.5.5 The results of false positive with weights (0.3, 0.4, 0.4).

# of false positive	Video 1		Video 2		Video 3		Video 4	
Never appear	1	1.5	2	1.35	1	1.35	1	1.3
	0		1		1		2	
	1		0		2		1	
	2		1		1		0	
	1		2		1		0	
	1		0		1		1	
	2		1		0		2	
	1		1		1		1	
	2		1		2		1	
	2		2		1		2	
	2		2		1		2	
	1		1		2		1	
	3		2		1		2	
	1		2		2		1	
	2		1		2		2	
	2		2		1		2	
	2		1		2		1	
	2		2		1		1	
	1		1		2		2	
	2		2		2		1	

Table 6.1.5.6 The results of false positive with weights (0.3, 0.4, 0.4) and the addition of face similarity.

# of false positive	Video 1		Video 2		Video 3		Video 4	
Never appear	1	1.6	2	1.45	1	1.4	1	1.35
	1		1		2		2	
	1		0		2		1	
	2		1		1		0	
	2		2		1		0	
	1		0		1		1	
	2		1		1		2	
	1		2		1		1	
	2		1		2		1	
	2		2		1		2	
	2		2		1		2	
	1		2		2		1	
	3		2		1		2	
	1		3		2		1	
	2		1		2		2	
	2		1		1		3	
	2		1		2		1	
	2		2		1		1	
	1		1		2		2	
	2		2		1		1	

Table 6.1.5.7 The results of false positive with weights (0.1, 0.5, 0.4).

# of false positive	Video 1		Video 2		Video 3		Video 4	
Never appear	1	1.6	2	1.45	1	1.35	1	1.35
	1		1		1		2	
	1		0		2		1	
	2		1		1		0	
	2		2		1		0	
	1		0		1		1	
	2		1		1		2	
	1		2		1		1	
	2		1		2		1	
	2		2		1		2	
	2		2		1		2	
	1		2		2		1	
	3		2		1		2	
	1		3		2		1	
	2		1		2		2	
	2		1		1		3	
	2		1		2		1	
	2		2		1		1	
	1		1		2		2	
	2		2		1		1	

Table 6.1.5.8 The results of false positive with weights (0.1, 0.5, 0.4) and the addition of face similarity.

6.2 Analysis of Erroneous Results

6.2.1 Human detection

The main reason of erroneous human detection may be the color similarity between the body of person and the pixel of background. It will cause incorrect results of edge detection. Some error results were shows as follows.

Fig. 6.2.1 Error results of detected persons.

6.2.2 Human segmentation

The reasons of erroneous human segmentation can be divided into two cases. In case one, some detected persons are not complete; this will affect the results of human segmentation. This kind of error comes mainly from the error of human detection. In case two, the color of the upper-body is similar to the color of lower-body. When using the edge information to segment the upper-body and lower-body, it is hard to discriminate the upper and lower body parts when the two parts have similar color. It is essentially ambiguous to differentiate the upper and lower body when the two parts have similar colors. Some error results were shown as follows:

Fig. 6.2.2 Error results of human segmentation.

6.2.3 Face direction

The reasons of error detection of face direction seem the same as the error reason of human detection. This is because some heads were not detected completely. It is mainly caused by errors of edge detection. Some error results were shown as follows:

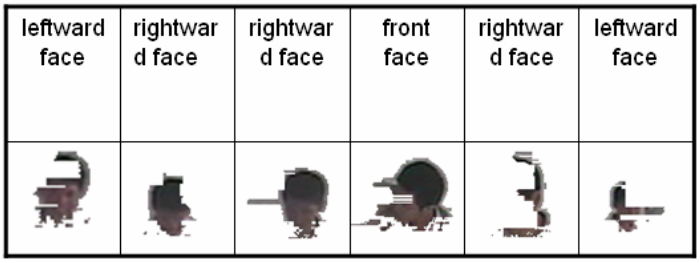


Fig. 6.2.3 Error results of face direction judgments

6.3 Discussion

1. The accuracy will increase when two faces have the same directions. If two faces have the same directions, we can add the face similarity to enhance our searching performance.

Frontal information of faces can be used to compare the query suspect with the video suspect when searching video suspects. If we compare two heads which both are back view, it is hard to promote the searching rate. This is because the information of the head is too deficient to provide enough strength to search suspects when the back view of a face was detected.

2. If the head of a query suspect was judged as a back view, then decreasing the weight of the head part and increasing the weights of other two parts will automatically increase the accuracy rate.

We discuss two kinds of situations below. First, using the front view of the query suspect to compare the front view of the video suspect, we can obtain very acceptable performance. Second, using the back view of the query suspect to compare the front

view of the video suspect, it has lower accuracy rate, because we have less information of the head. If we decrease the weight of the head part and increase weights of other body parts in this situation, we can promote our searching performance. This means we can put the emphasis on the comparison of the upper-body and lower-body parts when we meet the second situation.

CHAPTER 7 CONCLUSION AND FUTURE WORK

7.1 Conclusion

In this thesis, we have proposed the suspect searching system which consists of four main phases: human detection, human body decomposition and feature measurement, feature selection, and human searching. The proposed mechanism can be used for searching the specific person and decrease the time wasted on matching dissimilar persons.

In human detection, we have used the modified frame differencing to detect the moving persons in videos. In some cases, the detected human may be fragmental. We use the edge information to fill the missing foreground. As the experimental results shown, the accuracy rate can reach 90%.

In human body decomposition and feature measurement, we segment the human body into three parts and measure the different features in the different parts. Our correct rate of human body decomposition can achieve above 85%. In the part of feature measurement, we describe the different features we want to test in different parts. These features are mainly color-based and texture-based. In different body parts, the features used are different.

In feature selection, we use the rank-sum model to select the discriminative features. These selected features can be used for the next phase to search suspects. In this part, we describe the method how we measure the discriminate capability of each feature and determine those discriminate features.

In human searching, we define the similarity combination and measure the similarity between the query suspect and video suspects. As shown by the experimental results, the accuracy rate can reach above 80%.

From our experiments we can reach the following conclusions. First, the addition

of frontal face similarity helps to enhance the searching for the similar persons. Second, when the face of the query suspect was judged as a back view, if the proposed mechanism can decrease the weight of the head part automatically, it can reach more accuracy rate.

7.2 Future Work

In the future, we can include the unstable light conditions in outdoor environments and design a light-resistant method to handle variable intensity of suspects resulting from changing light. Further, more features are needed. For example, the shoes may increase the discriminative power. The addition of other features may provide more statistics to compare the query suspect with the video suspects.

References

[1] H. Yang and M.D. Levine, “The Background Primal Sketch: an Approach for Tracking Moving Objects, ” *Machine Vision and Applications*, vol. 5, pp. 17-34, 1992.

[2] C. Stauffer and W.E.L. Grimson, “Adaptive Background Mixture Models for Real-Time Tracking,” *Proceedings of the IEEE CS Conference on Computer Vision and Patter Recognition*, vol. 2, Fort Collins, Colorado, pp. 246-252, 1999.

[3] A. Elgammal, D. Harwood, and L. Davis, “Non-Parametric Model for Background Subtraction,” *Proceedings of International Conference on Computer Vision,* Kerkyra, Greece, September 1999, pp 751-767.

[4] S. Antani, R. Kasturi, and R. Jain, “A survey on the use of pattern recognition methods for abstraction, indexing, and retrieval of images and video,” *Pattern Recognit.*, vol. 35, no. 4, pp. 945-965, 2002.

[5] M.J. Swain and D.H. Ballard, “Color indexing,” *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[6] Y. Chahir and L. Chen, “Efficient content-based image retrieval based on color homogenous objects segmentation and their spatial relationship characterization,” *IEEE International Conference on Multimedia Computing Systems*, Vol. 2, 1999, pp. 705–709.

[7] M. S. Kankanhalli, B. M. Mehtre and J. K. Wu, “Cluster based color matching for image retrieval,” *Pattern Recognition*, vol.29, no.4, pp. 701–708, 1996.

[8] Z. Lei, T. Tasdizen, and D. B. Cooper, “Object signature curve and invariant shape patches for geometric indexing into pictorial databases,” *Proceedings of IS&T=SPIE Conference on Multimedia Storage and Archiving Systems II*, Dallas, TX, Vol. 3229, Nov. 1997, pp. 232–243.

[9] M. Adoram and M.S. Lew, “IRUS: image retrieval using shape,” *IEEE International Conference on Multimedia Computing Systems*, Vol. 2, Florence, Italy, July 1999, pp. 597–602..

[10] B. GTunsel and A. M. Tekalp, “Shape similarity matching for query-by-example,” *Pattern Recognition*, vol. 31, no.7, pp. 931–944, 1998.

[11] J.R. Smith and S.-F. Chang, “Quad-tree segmentation for texture-based image query,” *ACM International Conference on Multimedia*, San Francisco, California, October 1994, pp. 279–286.

[12] M. Borchani and G. Stammon, “Use of texture features for image classification and retrieval,” *Proceedings of IS&T=SPIE Conference on Multimedia Storage and Archiving Systems II*, Vol. SPIE 3229, 1997, pp. 401–406.

[13] A.P. Pentland, R.W. Picard, and S. Sclaroff, “Photobook: content-based manipulation of image databases,” *International Journal Computer Vision*, vol. 18, no. 3, pp. 233–254, 1996.

[14] F. Liu and R.W. Picard, “Periodicity, directionality, and randomness: Wold features for image modeling and retrieval,” *IEEE Transactions Pattern Analysis and Machine Intelligence* vol. 18, no. 7, pp. 722–733, 1996.

[15] T. Gevers and A.W.M. Smeulders, “Pictoseek: combining color and shape invariant features for image retrieval,” *IEEE Transactions on Image Processing*, vol. 9, no.1, pp. 102–119, 2000.

[16] Y. Yacoob and L. Davis, “Detection, Analysis and Matching of Hair,” *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference*, Oct. 2005, 741-748 Vol. 1

[17] K. Ueki and *et al.*, “A Method of Gender Classification by Integrating Facial, Hairstyle, and Clothing Images,” *Pattern Recognition, 2004. ICPR 2004 Proceedings of the 17th International Conference*, Aug. 2004, pp. 446- 449

Vol.4.

[18] B.Froba and C. Kublbeck, “Robust Face Detection at Video Frame Rate Based on Edge Orientation Features,” *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference*, Washington, DC, USA, May 2002, pp. 327-332

[19] J. Sivic and A.zisserman, “Video Google: A Text Retrieval Approach to Object Matching in Videos,” *Proc. Ninth IEEE Int’l Conf. Computer Vision*, vol. 2, Nice, France, Oct. 2003, pp. 1470–1477.

[20] S. Jabri, and *et al.*, “Detection and Location of People in Video Images Using Adaptive Fusion of Color and Edge Information,” *Proceedings International Conference on Pattern Recognition*, vol. 4. Barcelona, Spain, Sep. 2000, pp. 627-630.

[21] S. Jayaram and *et al.*, “Effect of Colorspace Transformation, the Illuminance Component, and Color Modeling on Skin Detection,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR'04)*, Vol.2. Washington, DC, USA, 27 June-2 July 2004, pp. 813-818.

[22] W.J. Kuo, ”Archiving of Human-Based Images from Video Sequences,” Department of Computer Science and Information Engineering National Chiao Tung University, master, July 2005.