

國立交通大學

電機與控制工程學系

碩士論文

利用可變編碼比對人形辨識系統

實現於 DSP 平台

Human Recognition System

Using Deformable Codebook Matching and

Realization on DSP Platform

研究生：李亞書

指導教授：林進燈 教授

中華民國九十六年七月

利用可變編碼比對人形辨識系統實現於 DSP 平台
Human Recognition System Using Deformable Codebook
Matching and Realization on DSP Platform

研 究 生：李亞書

Student：YA-SHU LEE

指導教授：林進燈 教授

Advisor：Dr. Chin-Teng Lin

國立交通大學

電機與控制工程學系



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

中華民國 九十六 年 七 月

利用可變編碼比對人形辨識系統 實現於 DSP 平台

學生：李亞書

指導教授：林進燈 教授

國立交通大學電機與控制工程研究所

中文摘要

本論文提出了一組快速而且計算量低的即時人形辨識偵測系統，其用途為在影像監視系統中提供人與非人物體之判別，並且可用在提供紅外線夜視攝影的攝影機上。本系統包含前景取得(Foreground)，人形辨識(Human Detection)，軌跡判別(Trajectory Tracking)等。由於本系統以建立在非支援浮點運算之 DSP 平台上為前提來進行研究，即時處理(Real-Time)的要求極為嚴苛，計算量以及精準度成為了本論文的第一要求。系統的第一部分在於取出特定場景中的移動前景，在這裡我們使用背景相減法(Background Subtraction)來做為取出移動物體的基礎。為了可以適合各種情況及不同的取像設備，我們使用了一個簡單快速的背景相減二值化閾值(Threshold)設定。第二部分提出了一些簡單的軌跡與狀態判別模式，在之後的人形辨識部分上提供一些必要資訊，以及降低誤判(False-Alarm)，誤判情況偵測等等的處理。而系統的第三部分，人形辨識的判斷法，基於在運算量的要求下，我們在許多的判斷方法之中選擇的以外形樣版為基礎的 Codebook 判斷方法來實現分辨人形與其他物體的不同。而為了解決室內常有的場景內人物下半身被遮蔽現象，我們提出了 Deformable Codebook Matching 的方法，可以提供半身以下不同比例的人形辨識機制，以完成雖有部分遮蔽仍有其辨識效果的系統。更進一步的延伸其用法到並排重疊的多人辨識機制。

Human Recognition System

Using Deformable Codebook Matching and Realization on DSP Platform

Student: Ya-Shu Lee

Advisor: Prof. Chin-Teng Lin

Department of Electrical and Control Engineering
National Chiao Tung University

Abstract

In this thesis, a fast real-time human detection system with low computing power is proposed. The purpose of this system is to provide the human detection and tracking for video surveillance which can be used in the environment with infrared rays lighting. This system consists foreground segmentation, human tracking, and human detection. The system will be implemented on the real-time DSP system which is not supporting the floating-point computation. The requirement of low computing power and accuracy becomes the major condition that we are very concerned. The first part of our human detection system is to segment the moving object from the scenes. We use the background subtraction here to segment the moving blob. We provide a simple and fast function to calculate the binarization threshold for the varying environments and videos taken by different cameras. In second part of our system, we use simple trajectory tracking and condition judgment to provide some data for human detection algorithm and to decrease the false-alarm rate. The final part is human detection. Because of the requirement of low computing power, we choose the shape-based method by codebook to classify human being from the other objects. The people walking indoor are sometimes covered by furniture such as desks or chairs. To solve this kind of problem, we provide deformable codebook matching, a human detection algorithm for first half body with different height/width ratio. With deformable codebook matching, when someone's bottom half body is covered, the system can still work. Further, we use deformable codebook matching to implement the human detection for multiple people walking side by side.

致 謝

首先，在這兩年之中最爲感謝的當然是我們的指導教授 林進燈教授，這段時間以來給予我的幫助與指導，讓我學習到許多寶貴的知識與經驗，在學業及研究方法上也受益良多。而傅立成，郭耀煌等口試委員們在口試時的建議與指教，使得本論文得以更加完備，在此深感致謝。

其次，要不是超視覺實驗室的學長鶴章、剛維及得正，博班學長子貴、建霆、肇廷、Linda 等眾位學長姐的從旁協助，本論文的許多環節將無法緊緊相扣，許多實驗也無法順利進行，實在是非常感激。而同學育弘、立倬、及訓緯的相互砥礪，以及 IC LAB 的靜瑩，智文及德瑋的友情相挺，更是讓我在研究所這條路上不會孤單不會寂寞。當然也不會忘記諸位學長姐、學弟妹們在研究過程中所給我的鼓勵與協助，交接人采蓉及晟輝的負責與認真。最後更是要感謝鶴章學長及剛維學長，在理論及程式技巧上給予我相當多的幫助與建議，讓我獲益良多。

最後感謝我的父母親對我的教育與栽培，並給予我精神及物質上的一切支援，使我能安心地致力於學業。

謹以本論文獻給我的家人及所有關心我的師長與朋友們。

Contents

Chinese Abstract	ii
English Abstract	iii
Contents	v
List of Tables.....	vii
List of Figures.....	viii
Chapter 1 Introduction.....	1
1.1 Motivation.....	2
1.2 Related Work.....	3
1.3 Thesis Organization	6
1.4 System Architecture	7
1.4.1 Software Architecture	7
1.4.2 Hardware Introduction	9
Chapter 2 Moving Object Extraction	11
2.1 Background Construction and Dynamic Background Update.....	11
2.2 Adaptive Foreground Extraction.....	14
2.2.1 Optimal Threshold Finding Function.....	14
2.2.2 Noise Elimination Filter.....	17
2.2.3 Connect Components Labeling.....	20
2.3 Moving Object Tracking	21
Chapter 3 Human and Non-Human Detection	24
3.1 Codebook Classification	25
3.2 Training Algorithm	28
3.2.1 Pre-classify (K-Means)	29
3.2.2 Training Algorithm	30
3.3 Deformable Codebook Matching.....	32
3.3.1 Full Body Matching	34
3.3.2 First Half Body Matching	35
3.3.3 Multiple Occlusive Human Detection	38
Chapter 4 Experimental Result	42
4.1 Simulation Result of Optimize Threshold Finding Algorithm	42
4.2 Simulation Noise Elimination Filter	46
4.3 Simulation Result of Deformable Codebook Matching.....	48

4.3.1 Human and Non-Human Detection	49
4.3.2 First Half-Body	53
4.3.3 Object Tracking Table	55
4.3.4 Multiple Occlusive Human Detection	56
4.4 Testing Environment	57
4.5 Accuracy of DCBM algorithm.....	58
4.6 Discussion	62
Chapter 5 Conclusion	64
References.....	66



List of Tables

Table 1 : The items of object tracking table.....	22
Table 2 : H/W ratio selecting table.....	34
Table 3 : Half body matching table.....	37
Table 4 : The accuracy of our system.	58
Table 5: The statistic of the accuracy (Video with only full body human).....	61
Table 6: The statistic of the accuracy (Video with lots half body human).....	61
Table 7: The average of accuracy above.....	61



List of Figures

Figure 1-1 : System architecture.....	7
Figure 1-2 : Human detection system.....	8
Figure 1-3 : Blackfin A-V EZ-Extender.....	9
Figure 1-4 : System overview.....	10
Figure 2-1 : Example of ITU-656 data format.....	12
Figure 2-2 : Arrangement of ITU-656 Format.....	12
Figure 2-3 : The flow chart of background model.....	14
Figure 2-4 : The flow chart of moving object extraction.....	15
Figure 2-5 : 4-connected and 8-connected sets.....	18
Figure 2-6 : Dilation diagram.....	19
Figure 2-7 : Erosion diagram.....	19
Figure 2-8 : Example of connected components labeling.....	20
Figure 2-9 : Object tracking table update process.....	22
Figure 3-1 : The flow chart of human detection.....	24
Figure 3-2 : The procedure of the comparison with the codebook.....	26
Figure 3-3 : Feature word extraction: shape information.....	27
Figure 3-4 : Feature word extraction: histogram of projection information.....	28
Figure 3-5 : Indoor video.....	32
Figure 3-6 : Outdoor video.....	32
Figure 3-7 : The flow chart of deformable codebook matching.....	33
Figure 3-8 : Threshold finding result.....	35
Figure 3-9 : Comparison with full and half body.....	36
Figure 3-10 : Two persons walk side by side.....	38
Figure 3-11 : Flow chart of multiple-occlusive human detection.....	39
Figure 3-12 : Projection histogram of Figure 3-10.....	39
Figure 3-13 : Projection histogram checking and separating.....	40
Figure 3-14 : Histogram analysis of a tree waved shadow.....	41
Figure 3-15 : Result of multiple-occlusive human detection.....	41
Figure 3-16 : Histogram analysis of three occlusive human.....	41
Figure 4-1 : The plot of threshold adjustment.....	43
Figure 4-2 : Example of utilization of the threshold adjustment function.....	44
Figure 4-3 : Scene of Figure4-4.....	44
Figure 4-4 : Example of utilizing three kinds of threshold.....	45
Figure 4-5 : Examples of noise elimination filter.....	46
Figure 4-6 : Other examples of noise elimination filter.....	47
Figure 4-7 : Snap shot of the output of our System.....	48

Figure 4-8 : Snap shot of the output of our system on PC.....	49
Figure 4-9 : Results of normal indoor environment.....	49
Figure 4-10 : Result of normal outdoor environment.	50
Figure 4-11 : Results of non-human object shown in the outdoor environment.....	51
Figure 4-12 : Results of non-human object shown in the indoor environment.	51
Figure 4-13 : Results of complex human detection.	52
Figure 4-14 : Results of multiple human detection.....	52
Figure 4-15 : Moving objects with human and non-human objects.	52
Figure 4-16 : Results of complex human detection.	53
Figure 4-17 : Results with only half-body (indoor).....	53
Figure 4-18 : Results with only half-body (outdoor).....	54
Figure 4-19 : Result of object tracking table.	55
Figure 4-20 : Results of multiple-occlusive man detection.	56
Figure 4-21 : List of testing environment.	57
Figure 4-22 : Example of system fail #1.....	63
Figure 4-23 : Example of system fail. #2.....	63
Figure 4-24 : Example of system fail. #3.....	63



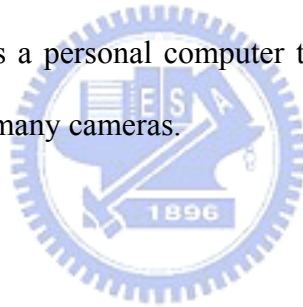
Chapter 1

Introduction

In recent years, visual surveillance systems play an important role in security. We can easily find visual surveillance systems everywhere, not only in the street and open store but also in many houses and companies. There are two common types of video surveillance system: one is to record video from video surveillance system without human monitoring. It is already prevalent in commercial establishment with camera output being recorded periodically or stored in video archives. The other is sending video to the monitors with continuously human monitoring.

The disadvantage of the given surveillance system is that it will take lots of storage equipments, and even waste money and resource, if we only record the video from surveillance TV camera as data. For example because of the limitation of storage space in the video surveillance system, the data can not be reserved for a long time. If there was something happened and we can't get the related video information in time, it will be difficult to recover the interesting extract from mass video data. In the building or some close environment, the video sequences from many remote areas can be presented to watchmen at a time. However, looking at many TV monitors for a long time is hard work for watchmen. Research shows that normal adults can only focus on the monitor less than fifty minutes. Further more, it is expensive to find available human resources to sit and watch the video. Therefore the automatic detection of moving objects and the classification of the detected objects are required in video surveillance system. In particular, distinguishing human from other objects is an indispensable function for security systems.

The advanced video surveillance system needs to analyze the behaviors of people in order to prevent the occurrence of the potential dangerous case. The analysis of behaviors of people requires the human detection and tracking system. In recent years, the developments of human detection and tracking system have been going forwards for several years, and many real time systems have been developed. However, there are still some challenging technologies need more researches: most of systems need heavy computing load because of foreground extraction and classify algorithm. If a video surveillance system can provide occlusion people detection or shadow handle algorithm, it always comes to the problem of heavy computing load. There have been some real-time video surveillance systems, but most of them are developed on personal computer platform with fast modern CPU. It cost too much when every input video needs a personal computer to analysis, if we have much of area need to be monitored by many cameras.



1.1 Motivation

In recent years, visual surveillance systems play an important role in security. A problematic issue is to distinguish people from other moving objects such as animals or traffic. Although many researches have been studied on this problem, it is still a leading issue in visual surveillance systems.

Human detection is one major issue of object detection. Its main idea is to find whether there are human in the video (or image) or not. A successful human detection can tell us how many people are there in the one single image. This kind of human detection system scans whole image to find out if there is any human feature. There are several systems which can do this kind of job and provide a good search result. The question is this kind of human detection system takes very long time to calculate

and search. It is not suitable for real time human detection.

Real-time human detection systems are always separated into two parts: finding the moving object in the video sequence and then analyzing the moving object. When a moving object get into the range of interesting, we need to locate its position and size, and even, its speed and direction. We call this process foreground extraction or foreground segmentation. It is difficult to extract the moving object from background perfectly. Foreground extraction can be affected by light, shadow, camera shake, noise and background object...etc. The second part is to classify what the moving object is. At this part, human detection systems exact all kinds of features and use mathematic equations or advance algorithm such as neuron networks to achieve the goal. But to distinguish human from other object is still a challenge when two or more people walking too close or when people are covered by some background object.

The several reasons mentioned above motivate us to develop a fast real-time human detection system with low computing load on the DSP platform. It can distinguish human from other object in a low cost and easy to set up. In human detection we use shape-based model to modeling the human. This human model can be used to detect moving human that has variation of size after normalizing, and provide a deformable matching algorithm to handle the situation when half body of human appear or few people walk too close.

1.2 Related Work

In recent years, many human detection approaches have been developed. There are two parts of human detection system: segmentation of moving object from background and human detection by distinguishing the human with other moving objects. Several methods for moving object segmentation are optical flow method [1],

[2], and [3], stereo based vision, and temporal difference method. Optical flow is used to detect independently moving objects but it has complex computation and sensitive to change of intensity. Optical flow used in [2], [3] was used to detect vehicle. Zhao et al [4] exploited stereo based segmentation algorithm to extract object from background and to recognize the object by neural network based recognition. Although stereo vision based technique have been proved to be more robust it requires at least two cameras and can be used only for short and middle distance detection. Carlos Orrite-Urunuela [8] used multiple cameras to analyze the 3D skeletal structure in gait sequences. They used 3D skeletal structure to make sure the shape of moving human can be completely extracted. And then they followed a point distribution model (PDM) approach using a Principal Component Analysis (PCA) to establish the shape of human. The fitting was carried out by selecting the closest allowable shape from the training set by means of a nearest neighbor classifier. They developed a human gait analysis to take into account temporal dynamic to track the human body. But this system need to use 3D skeletal structure, it needs multiple cameras to generate. And they can only successful extract whole shape in the simple and clean environment. The size of human must be large enough for their algorithm, which is a disadvantage for video surveillance system. Smith et al [5] used background subtraction method to segment isolate human. The serious problem of this approach is the changeable background or the illumination that is almost different in each frame. Zhuo-Lin Jiang [9] also used background subtraction method to segment isolate human. To avoid shadow they use the homogenous of shadow and background to eliminate the shadow. A time study algorithm was able to exclude the background object such as window curtains and indoor plants. Area thresholds can avoid sudden change of light or illumines interfere the moving object extraction. However they only eliminate shadow, animal and background object and then take rest of moving objects

as human. It is faster but less accurate. Y.L.Tian and A.Hampapur combine these two techniques together [10]. They firstly use the background subtraction to locate the motion area, and then perform the optical flow computation only on the motion area to filter out false foreground pixels. The background subtraction is popularly used in foreground segmentation. The motion information is extracted by thresholding the difference between the current image and background image. The background can be modeled as Gaussian distribution $N(\mu; \sigma^2)$, this basic Gaussian model can adapt to gradual light change by recursively updating the model using an adaptive filter. However, this basic model will fail to handle multiple backgrounds, such as water wave and tree shaking. To solve the problem of multiple backgrounds, the models such as the mixture of Gaussian, Nonparametric Kernel [11], and codebook [12] are provided recently. Although these algorithms are effective for modeling multiple backgrounds, they require more memories and more computation.

To distinguish human with other object (human recognition), several method have been implemented such as shape-based, motion-based, and multi-cue based methods. The shape-based approach uses shape feature to recognize human. Motion based approach use Fast Fourier Transform and its periodicity against time [6]. Some system integrates multiple features to recognize human such as shape pattern, motion pattern, skin color, etc. Curio et al [7] used the initial detection process that is based on geometry feature of human. Then, motion patterns of limb movements are analyzed to determine initial object hypotheses. There is another way for human recognition, such as neural network based approach. The neural network is powerful tool for pattern recognition. In [13], the BP network was used to recognize the pedestrian. The model based human recognition system analyzes the shape of object and classify the people from other objects. In order to recognize the people, we utilize the codebook method to model the shape of human, and propose the distortion

sensitive competitive learning algorithm to design the codebook. Sang Min Yoon, [14] used robust skin color, background subtraction and human upper body appearance information. They extracted the human candidate regions using color transform and background subtraction and update. To classify human and other objects that have similar skin color region or motion, an efficient incorporation of geometric pixel value structure and model based image matching using Hausdorff distance are implemented. Daoliang Tan, [15] they dealt with the problem of night gait recognition via thermal infrared imagery. First of all, human detection was accomplished, based on the Gaussian mixture modeling of the background. Then, human silhouettes were extracted on the basis of preceding detection results. Moreover, a new gait representation called HTI was proposed to characterize gait signatures for recognition.

1.3 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 1-4 describes system overview including software and hardware architecture. Chapter 2 describes the foreground extraction, adaptive thresholding and moving object tracking. Chapter 3 shows detection system including human detection and deformable matching system. Chapter 4 shows the experimental results. Chapter 5 makes the conclusions of this thesis and the future works.

1.4 System Architecture

1.4.1 Software Architecture

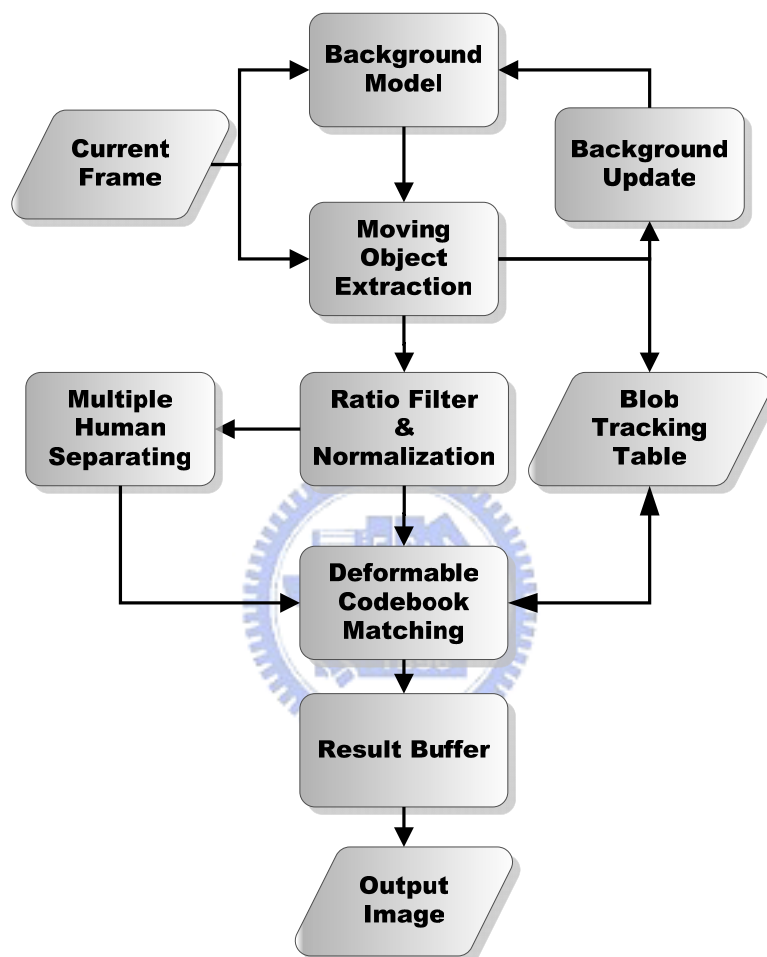


Figure 1-1 : System architecture.

Figure 1-1 shows the overview of our human detection system. The static camera captures the current frame and feeds into our system. Our human detection system can take TV video sequence from camera and show the result to the TV monitor. After some de-noise filter, the system construct the background model for moving object extraction. Moving object extraction process extracts the foreground objects by subtracting with background model, and establishes a object tracking table for object

tracking and providing the information for classification. After using some filters and normalization, the deformable codebook matching algorithm can distinguish human from other objects.

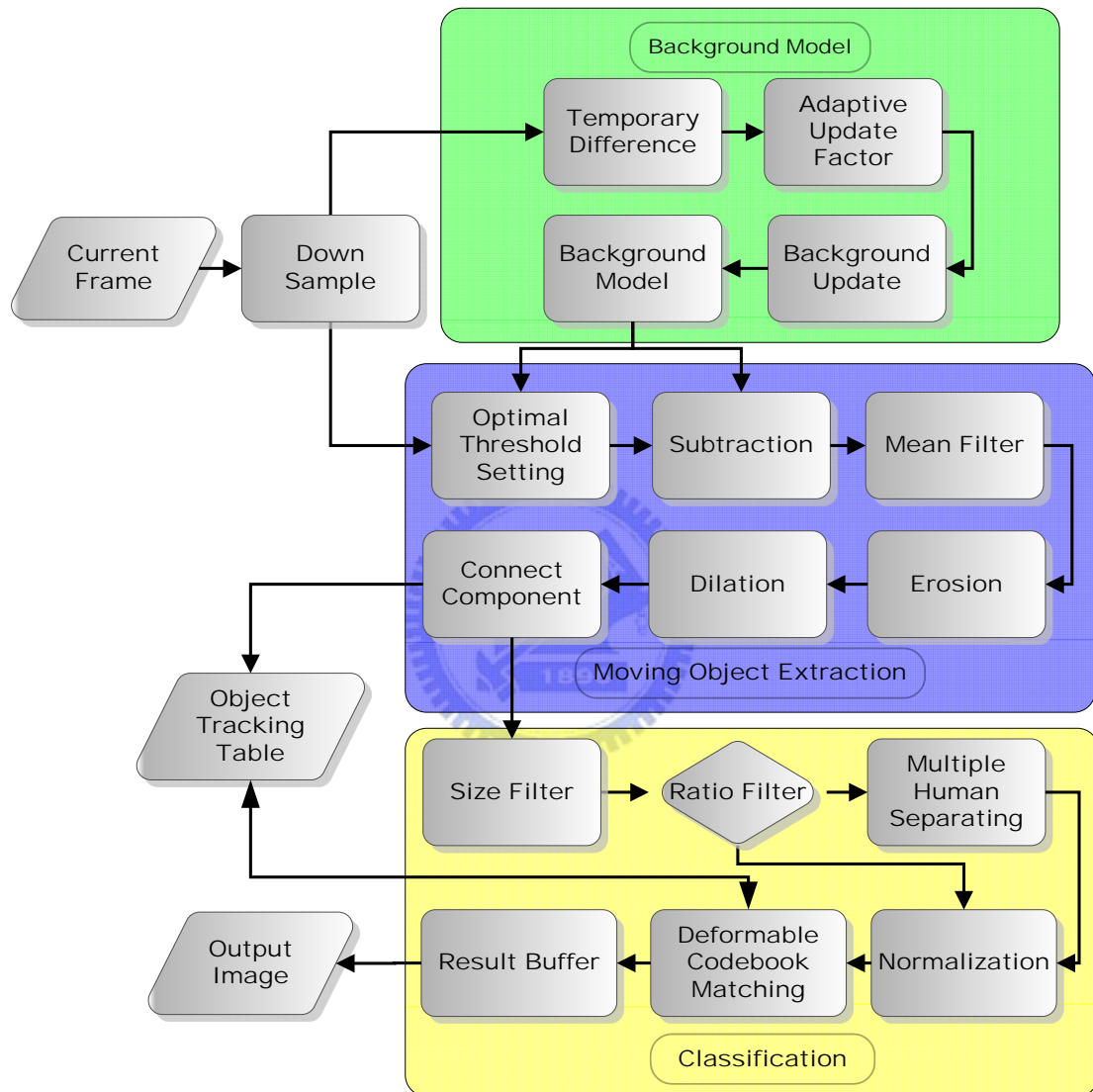


Figure 1-2 : Human detection system.

Figure 1-2 is the detailed flow chart of our human detection system. The system is divided into three blocks: background model block, moving object extraction and object tracking block, and classification block. After down sample, current frame is used to establish the background model and update the background frame and we call

this block background model block. Moving object extraction use optimized threshold setting to support background subtraction method. Erosion and dilation operation can decrease the noise influence and get a more complete shape. The connect component process provide us a full region of moving object, and the object tracking table are set up when regions of moving object confirmed. The classification block provides us full body human detection, first half body human detection and multi-human detection with deformable codebook matching algorithm. And the data from ratio filter and multi-human separating part is necessary for deformable codebook matching algorithm.

1.4.2 Hardware Introduction



Figure 1-3 : Blackfin A-V EZ-Extender.

Our human detection system is built on the DSP platform of Analog Device ADSP-BF561 EZ-KIT Lite. The Blackfin A-V EZ-Extender daughter board is used to allows us to evaluate a diverse set of peripherals on the ADSP-BF561 processor. The EZ-Extender contains video connectors allowing connection to camera sensor. We can

develop our algorithm on the Analog Device VisualDSP ++ and execute the code on the development kit and processor. And Figure 1-4 shows the overview of our system.

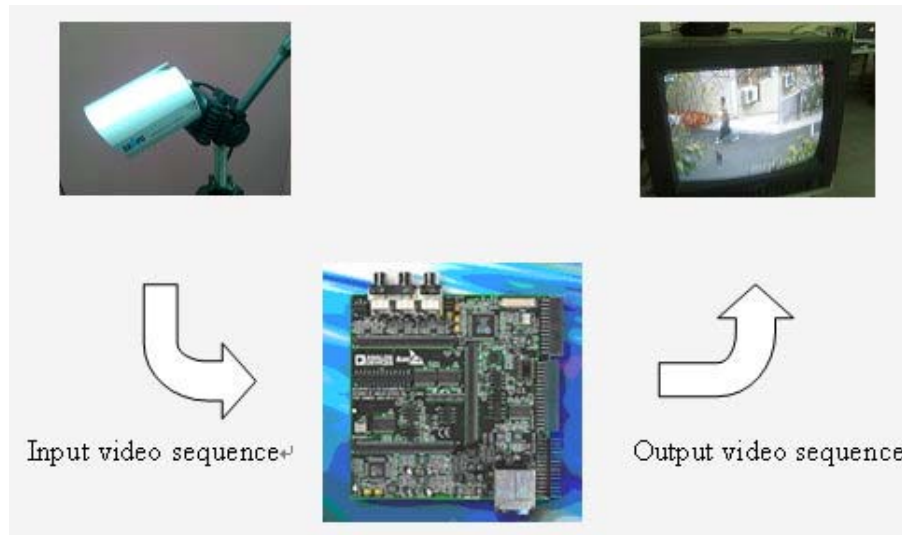
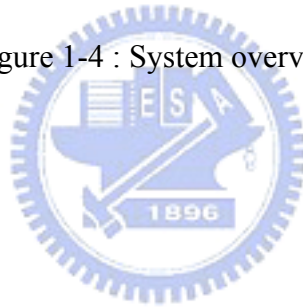


Figure 1-4 : System overview.



Chapter 2

Moving Object Extraction

2.1 Background Construction and Dynamic

Background Update

Before we construct our background, there are something about the video stream format we should know.

The video sequence format comes from DSP development kit is ITU-656. ITU-656 data stream is a sequence of 8-bit or 10-bit bytes, transmitted at a rate of 27 Mbyte/s. Horizontal scan lines of video pixel data are delimited in the stream by 4-byte long SAV (Start of Active Video) and EAV (End of Active Video) code sequences. SAV codes also contain status bits indicating line position in a video field or frame. Line position in a full frame can be determined by tracking SAV status bits, allowing receivers to 'synchronize' with an incoming stream. Individual pixels in a line are coded in YCbCr format. After a SAV code (4 bytes) is sent, the first 8 bits of Y (luminance) data are sent then 8 bits of Cb (chroma U), followed by 8 bits of Y for the next pixel and then 8 bits of Cr (chroma V), the 4:2:2 digital video encoding parameters.

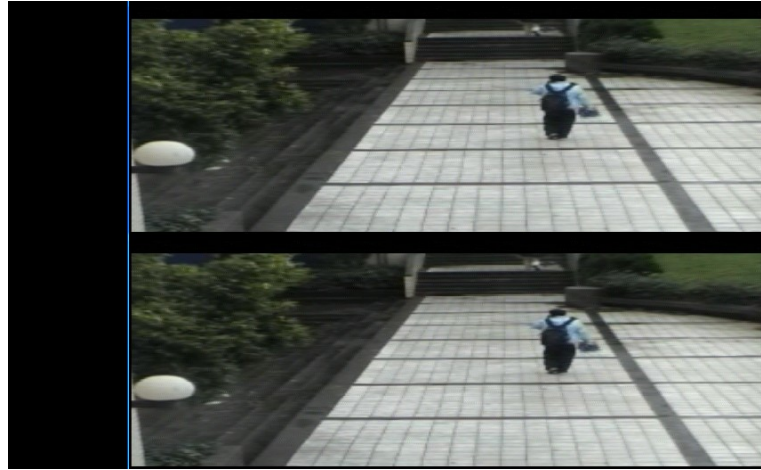


Figure 2-1 : Example of ITU-656 data format.

An ITU656 video frame is divided into two fields, field 0 and field 1. Figure 2-1 shows that what one frame actually looks like, and Figure 2-2 shows the arrangement of ITU-656 in each frame. The black part is head data, the others are the video sequence. The ITU-656 video format size is 858*525 and the actually pixel number is 720*486, which is too large for real-time processing and take too many space in the memory. Therefore, the first step is to down sample the current frame. We down sample the current frame from 720*486 to 180*122, about quarter size of each side. The efficient way is to use only one field and down sample the width two times than the height.

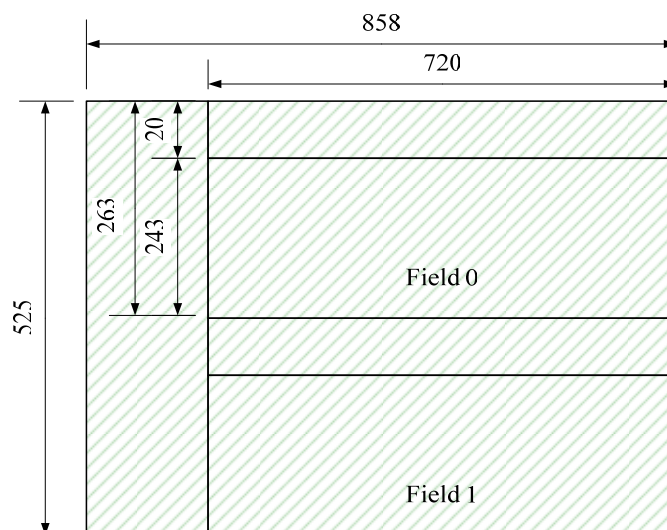


Figure 2-2 : Arrangement of ITU-656 Format

After down sample the current frame, we can build the background. The system will delay for few seconds while system starting, because of booting the DSP and waiting for the stable video sequence from camera. The system will delay for few seconds. The first frame is taken after whole system comes to stable. In order to construct the background for moving object extraction, the first current frame will be taken for the original background model. Here we only use the Y (luminance) channel for our system. If we only focus on the shape and the movement of foreground objects, using the gray level image but colorful image can decrease the computing power and make the issue easier.

If the background constructed at first dose not adapted as time proceeding, the system will be failed. The light change, moving background objects or camera movement will cause the false result of system, which must to be prevented. The simple way to avoid the above problem is to update the background once in a while. We use Eq. 2-1 to update the background after a period of time T_u .

$$P_B^n = \begin{cases} \alpha P_B^{n-1} + (1-\alpha)P_C, & I_M = 0 \\ P_B^{n-1}, & I_M = 1 \end{cases} \quad (2-1)$$

Where P_B^n represents each pixel of current background and P_B^{n-1} is pervious one. I_M represents each pixel of active part between previous frame and current frame. α (0 , 1) will be the update factor of background updating. Every time we update the background, the current frame would be saved to memory. Next time when we need to update background we do the temporal difference between the saved frame and current frame. We set I_M to 1 when the gray value of temporal difference higher than a threshold, and set I_M to 0 if this pixel doesn't change a lot.

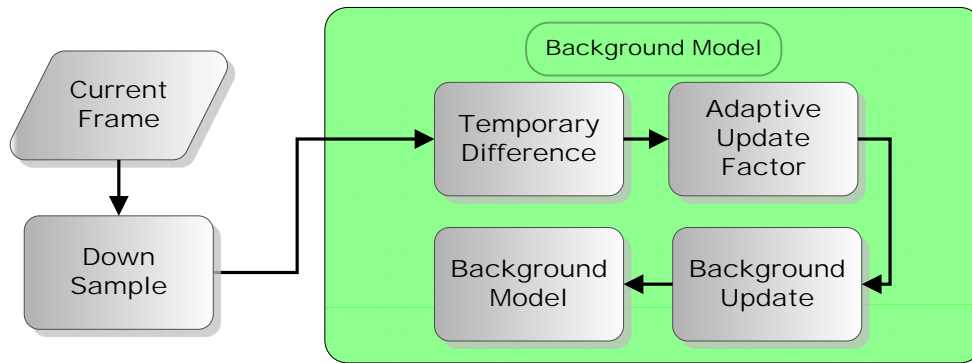


Figure 2-3 : The flow chart of background model.

Figure 2-3 is the flow chart of background model. The adaptive update factor α is the key to control background update rate. There are two cases which may cause α increase or decrease. First, the number of pixels with $I_M = 1$ is less than a number for a while and the result of background subtraction shows there are many moving point, which is caused by camera movement or heavy change of light. The α should increase to speed up the update rate for getting background stable. Secondly, if standard deviation of current frame is much large then standard deviation of background, α should increase too. This situation may be caused by local light change or shake of branches.

2.2 Adaptive Foreground Extraction

2.2.1 Optimal Threshold Finding Function

The background subtraction system is used to provide foreground object through taking threshold of difference image between the current image and reference image.

If the reference image is the previous frame, this method is called temporal difference. The temporal difference is very adaptive to dynamic environment, but generally does a poor job of extracting all relevant feature pixels. The improved methods such as mixture of Gaussian and Nonparametric Kernel can behave better performance, but they need extra expensive computation and more memories. For real time system, especially integrated with real time DSP system, we don't think their cost is worthy, because the CPU usage and memories usage are very significant for system stability.

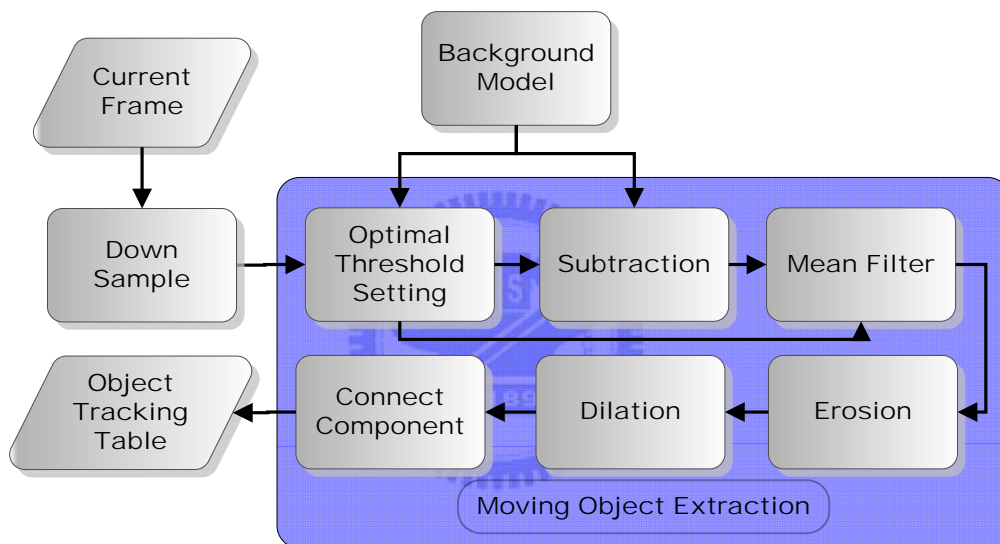


Figure 2-4 : The flow chart of moving object extraction.

If we want to use background subtraction for extracting moving objects. The selection of threshold (TH) is the key for successful moving object extraction. If TH is too low, some background are labeled into foreground. If TH is too high, some foreground are detected as background.

$$TH = TH_0 + \beta \quad (2-2)$$

The traditional way is to select a based threshold and according to the variation of difference value between background and foreground to do some modification of threshold. But this is based on the assumption that illumination gradually changes. However when light suddenly changes, such as suddenly turning off the light and camera activating the infrared ray lighting, this assumption will be violated. Instead of the traditional way, we have to automatically select TH according to the level of light change. Equation 2-2 is the concept of our adaptive threshold selecting function. We select a based threshold TH_0 and mix with a variable shifted β based on standard deviation. The definition of TH_0 is followed.

$$TH_0 = \begin{cases} \gamma, & P_B < avg_B / 2 \\ (P_B - avg_B / 2) \times \frac{(th - \gamma)}{avg_B / 2} + \gamma, & avg_B / 2 \leq P_B < avg_B \\ th, & P_B \geq avg_B \end{cases} \quad (2-3)$$

Firstly, we search the average value avg_B of background model. After that, Equation 2-3 is used to compute TH_0 for each pixel. th is the based reference threshold. P_B is the gray value of each pixel in the background model. γ is the lowest value of TH_0 , and is defined by the standard deviation, STD_B .

$$\gamma = a \times STD_B \quad (2-4)$$

$$\beta = b(avg_C - avg_B) + c(STD_C - STD_B) \quad (2-5)$$

Each one of a, b, c is a constant value between 0 and 1. avg_C, avg_B are the average of all pixel of current frame and background frame. STD_C, STD_B are the standard deviation of whole current frame and background frame respectively.

In other words, our threshold is based on background gray value. The dark part in the background frame has lower threshold, on the other side, the bright part of background has higher threshold. And the change of threshold is according to the average and standard deviation of current frame and background model. To prevent the heavy computing consumption, this threshold selecting method is activated when background update process.

2.2.2 Noise Elimination Filter

Now we have a good threshold adaptation method to extract the moving foreground object, but the noise is always a problem for the system even we have adaptive threshold. Traditionally, there will be some noise elimination filters before background subtraction. However, the traditional noise elimination filters will be violated when light suddenly changes. Instead of the traditional way, we have to find some easy way to improve the effect of this kind of noise elimination filters. The key of our noise elimination process is to put off the time of using filters till finish of background subtraction. And instead of all kinds of complex filter such as median filter or Gaussian filter, we use only mean filter to get better result.

Here is the description of our method. After background subtraction, we have initial process frame. The value of each pixel in the initial process frame is set to zero if the difference of each pixel between background frame and current frame is smaller than TH . The others will be set to the value of the result of background subtraction. And now we use five by five mean filter to decrease the inference of noise.

Mean filter is a kind of smoothing linear filter. The output of a mean filter, a smoothing, linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask. The mask is shown as follow:

$$\frac{1}{25} \times \begin{array}{|c|c|c|c|c|} \hline \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \hline \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \hline \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \hline \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \hline \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \hline \end{array}$$

After remove most of noise, we are able to use the dilation and erosion operations to make the shape of moving objects more complete. Dilation, in general, causes objects to dilate or grow in size; erosion causes objects to shrink. The amount and the way that they grow or shrink depend upon the choice of the structuring element. Dilating or eroding without specifying the structural element makes no more sense than trying to low-pass filter an image without specifying the filter. The two most common structuring elements (given a Cartesian grid) are the 4-connected and 8-connected sets. They are illustrated in Figure 2-5.

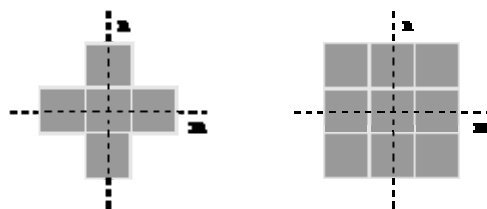


Figure 2-5 : 4-connected and 8-connected sets.

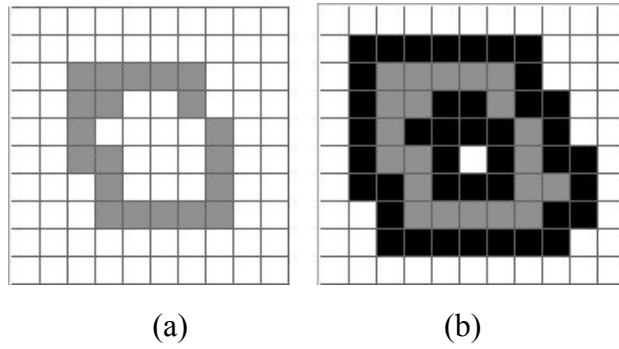


Figure 2-6 : Dilation diagram.

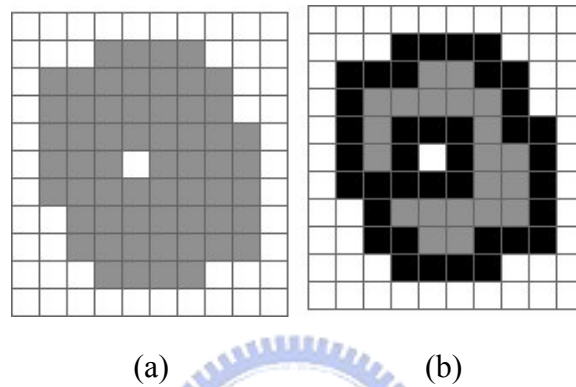


Figure 2-7 : Erosion diagram.

We can see the result of dilation operation in Figure 2-6. The left side (Figure 2-6(a)) is the original object (gray part). The result is the pixels mixed by the gray and black points as shown in Figure 2-6(b). After dilation process, we can see the gray points are surrounded by black points. It becomes bigger. In the other hand, the result after erosion process is shown in Figure 2-7. Figure 2-7(b) is the final result of erosion operation. The gray part in Figure 2-7(b) is the result after process and the black part is eroded by operation.

2.2.3 Connect Components Labeling

The dilation and erosion operation make the moving object clearly. What we need to do now is to segment the exact location and size of these objects. The connected components labeling method is what we need to do for extract the whole object from discrete points. Figure 2-8 (a) shows that without connected components labeling and all interesting points belong to 1 and others are 0. That means all points are not connected. Although we can easily distinguish these four objects, but when they don't have any connection the computer can't tell the difference. So the connected components labeling is necessary and Figure 2-8 (b) shows that connected components labeling separate four un-overlap region and paint it in different color where each color represents a single separated region.



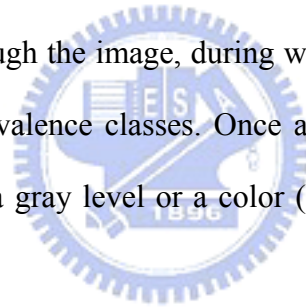
(a) Before connected components.

(b) After connected components.

Figure 2-8 : Example of connected components labeling.

The connected components algorithm is frequently used to achieve this work. Connectivity is a key parameter of this algorithm. There are 4, 8, 6, 10, 18, and 26 for connectivity. 4 and 8 are for 2D application and the others are for 3D application. We used the 8-connectivity for our implementation. The connected component algorithm

worked by scanning an image, pixel-by-pixel (from top to bottom and left to right) in order to identify connected pixel regions. The operator of connected components algorithm scanned the image by moving along a row until it came to a point (P) whose value was larger than the preset threshold of extraction. When this was true, according to the connectivity it examined P 's neighbors which had already been encountered in the scan. Based on this information, the labeling of P occurred as follows. If all the neighbors were zero, the algorithm assigned a new label to P . If only one neighbor had been labeled, the algorithm assigned its label to P and if more of the neighbors had been labeled, it assigned one of the labels to P and made a note of the equivalences. After completing the scan, the equivalent label pairs were sorted into equivalence classes and a unique label was assigned to each class. As a final step, a second scan was made through the image, during which each label was replaced by the label assigned to its equivalence classes. Once all groups had been determined, each pixel was labeled with a gray level or a color (color labeling) according to the component it was assigned to.



2.3 Moving Object Tracking

The connected components labeling process extracts all separate object and give them indexes. Then we can use these data to build a table for these objects. We call this table **Object Tracking Table**. This object tracking table contains several items, include life time, coordinates, classify record...etc. Table 1 shows all items in the object tracking table. Several items are useful for further classification which we will discuss in chapter 3, such as H/R ratio or histogram.

Table 1 : The items of object tracking table.

Item	description
Index	Labeling number
Coordinates	Top-right and bottom-left coordinates
Life Times	Life period
FindorNot	Search result for comparing previous object tracking table
FindorNot Counter	Search result buffer
NeworNot	New object or not
Classify Record	Classify Record
H/W Ratio	Height, width ratio.
Size	Size
Histogram Data	Histogram for pixel number project to width axis

To build object tracking table there are several steps we need to do. Firstly, we need to know the object existed in current frame will or not exist in next frame. Secondly we need to analysis the character of object to assist our classification process. The flow chart of how to know the object shown in this frame is still here when next frame comes in is as shown bellow.

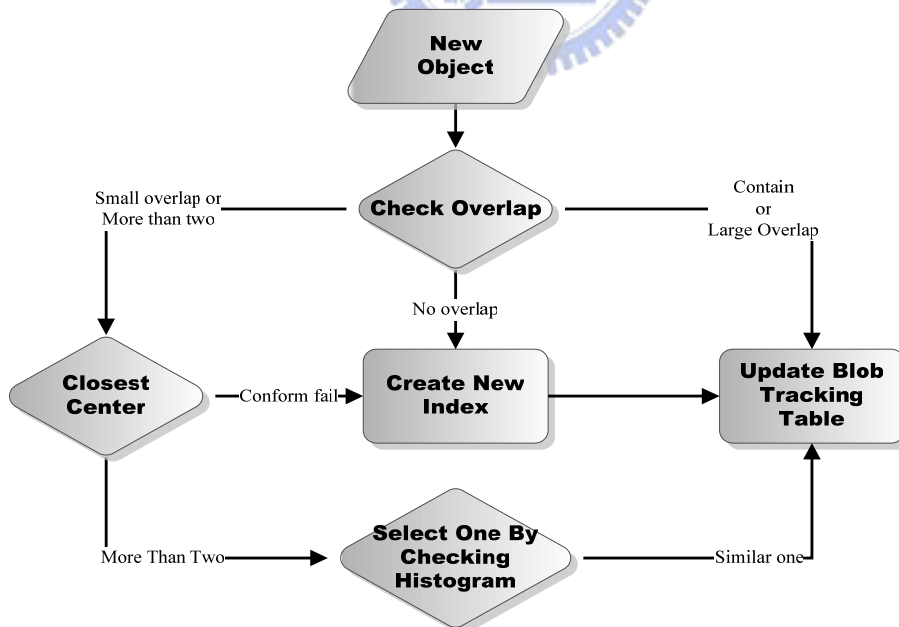


Figure 2-9 : Object tracking table update process.

We check every new object to see if the object is belonging to one of object tracking table. The first step is checking the overlapping status. There are three

condition of overlapping status: one contains the other or big overlapping area, small overlapping area and more the one overlapping object, and not overlap. When the area of one object contains the other or there is seemed to be a big overlapping area, we take these two objects are the same one and update the object tracking table. If they only have small overlapping area or the compared result shows that the new object has more the one overlapping object, we will use the center of object to tell the difference. We set a threshold for difference between two centers of object. If the difference is upper than threshold, the comparison is fail. If the difference is lower than threshold, we take these two objects the same. When more than one object is success compared, the final check method is present. We use histogram comparison to be the final gate of object tracking table updating process. When new object connect to more than two object lists in the object tracking table and they are close enough for center threshold, we use the histogram of each object to calculate the SAD (Sum of Absolute Difference). Because our system is a real-time system, when FPS is 30, using SAD to identify the similarity is practicable.

Chapter 3

Human and Non-Human Detection

In this chapter, we will explain how the human and non-human detection process works. Because of the requirement of low computing power, we choose the shape-based human model to classify human being by codebook matching, which decrease the performance of human detection from the other objects. The people walking indoor are sometimes covered by furniture such as desks or chairs. To solve this kind of problem, we provide Deformable Codebook Matching, a human detection algorithm for first half body with different height/width ratio. With Deformable Codebook Matching, when someone's bottom half body is covered, the system can still work. Further, we use Deformable Codebook Matching to implement the human detection for multiple people walking side by side. Figure 3-1 is the flow chart of human and non-human detection process.

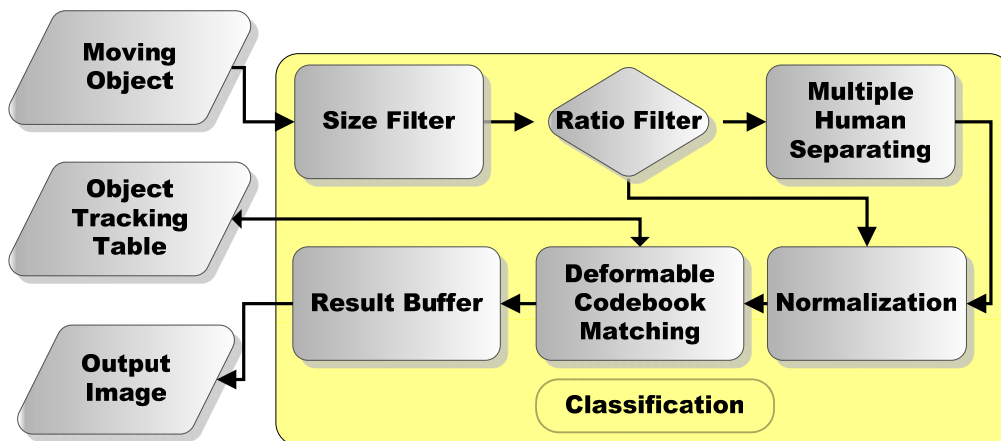


Figure 3-1 : The flow chart of human detection.

3.1 Codebook Classification

The ultimate goal of our developed system is to be able to identify people and track individuals to find out what they are doing. But the most of we can do now is to do the classification of human and non-human. The algorithm we used is presented below. For human recognition, we use the codebook to classify the human from other objects. At first, we normalize the size of human being in any attitude to 20 pixels at the horizontal by 40 pixels at the vertical, and then extract the shape and the histogram of object as the feature code to construct the codebook. Second, we match this feature vector against the code vectors in the codebook. The purpose of matching process is to find a code vector in codebook with the minimum distortion to the feature vector of object. If the minimum distortion is less than a threshold, we consider this object as human.

In order to describe how we use the codebook to classify the human from other objects, there are some variables should be defined at first. If we can extract a series of features as feature word X from every normalized image, and each of X include data of M dimensions, indicated by $X^0 \cdots X^i \cdots X^{M-1}$. There are N sets of code word V defined as $V_0 \cdots V_j \cdots V_{N-1}$ in codebook C . Each of V_j just like X has M dimensional data defined as $V_j^0 \cdots V_j^i \cdots V_j^{M-1}$. The distortion between feature word and code word is defined in Equation 3-1.

$$Dis_j = \|X - V_j\| = \sum_{i=0}^{M-1} |X^i - V_j^i| \quad (3-1)$$

$$Dis_{\min} = \min(Dis_j) \quad j = 0 \cdots N-1 \quad (3-2)$$

With the definition of these variables above, we can explain the procedure of the human detection. Every time when we get a new foreground object, we do the normalization to get a uniform size image. After normalization, we take the feature word X from this new object. And the way we extract the feature word X will be shown in the next section. The feature word X is used to compare with every V_j in the codebook (C). The compared function $DIS(X,V)$ is shown in Equation 3-1. Dis_{\min} is the minimum of comparing result in the N code words. If the value of Dis_{\min} is smaller than the threshold we defined, the object with the feature word X is considered as human; otherwise, it is not a human object. Figure 3-2 shows the demonstration of comparing X with V_j in the codebook.

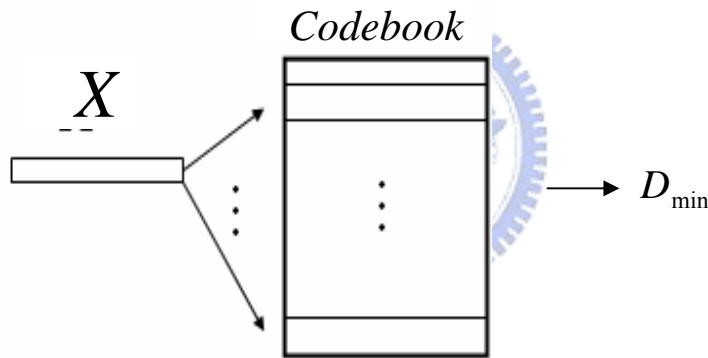


Figure 3-2 : The procedure of the comparison with the codebook.

And the way to extract the feature word X is described as follows. After normalizing the object image to 20 pixels by 40 pixels, we use a vector with twenty elements to describe the shape of the foreground object, and a histogram vector of ten elements by the projection of the object image on X axis is also used to increase the accuracy. To extract the shape features of foreground object, we draw a horizontal line with fixed coordinate at Y axis on the normalized image. Both the leftest and most right intersections of the horizontal line and the boundary of the object are recorded to represent the shape information. The features of feature word are obtained by drawing

ten horizontal lines. The twenty coordinates at X axis of the twenty intersections forms the feature word to represent the shape information. Figure 3-3 shows the feature word extraction in the image (white points).

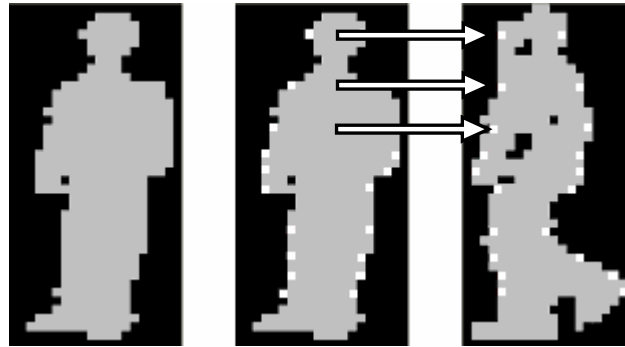


Figure 3-3 : Feature word extraction: shape information.

There are something we need to notice. The top two and bottom two row of pixels are not suitable for the feature word because these pixels are changeable. The way we find ten fixed Y axle values is to calculate the standard deviation in each fixed Y axle for total four thousand training samples, and then chooses ten lowest values each side to determine the coordinate Y axis of these ten horizontal lines.

After the shape information extraction mentioned above, we get feature word with twenty elements. The way to find the histogram of projection information is described below. Figure 3-4 (b) shows the final ten dimension of the feature word. We project the mask image to the X axis and calculate the pixel value to build a histogram of the projection on X axis. We take ten values of histogram for the feature word. If we only use discrete shape information, some of hollow object may not be detected correctly. The histogram of projection information can eliminate lots of non-human object and prevent the false alarm. The diagram to extract the histogram of project information is shown in Figure 3-4 (b)

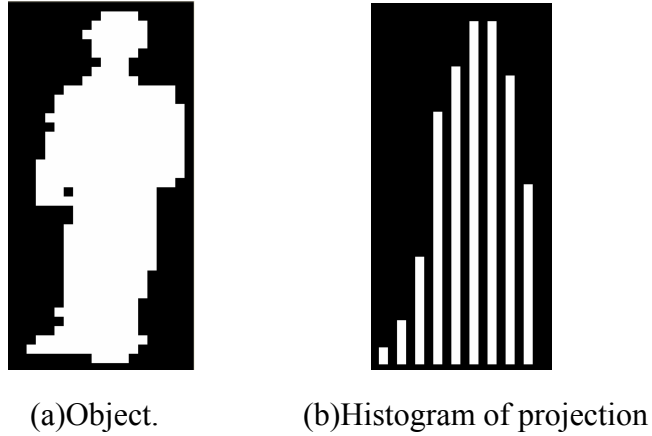
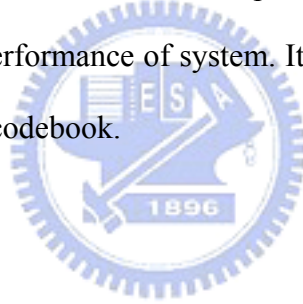


Figure 3-4 : Feature word extraction: histogram of projection information.

After describing our procedure of human detection, we will illustrate how to build a codebook for the distortion measurement in the next section. When we build a codebook for the classification, the further step is to overcome some problems encountered to improve the performance of system. It will be discussed in the section after the establishment of the codebook.



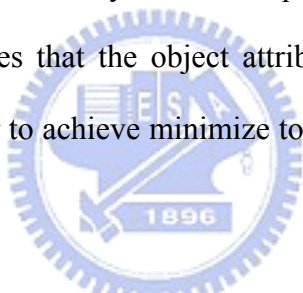
3.2 Training Algorithm

The design of the codebook is critical for the classification. The well-known partial distortion theorem for codebook design is that each partition region makes an equal contribution to the distortion for an optimal quantizer with sufficiently large N [16] and [17]. Based on this theorem, we use the distortion sensitive competitive learning (DSCL) algorithm to design the codebook. In order to describe this algorithm, we define $V = \{V_j; j = 1, 2, \dots, N\}$ as the codebook and V_j is the j^{th} code vector. X_i is the i^{th} train vector and L is the number of train vector. D_i is the partial distortion of region R_i , and D is the average distortion of codebook. The DSCL algorithm is described as follows.

3.2.1 Pre-classify (K-Means)

The first step of the training algorithm is to initiate a set of code words in the codebook for the initial sets, and we select the K-Means to do this job. We use K-Means algorithm to build N prototypes from M_i training sample for the first step of training algorithm. The simple description of K-Means algorithm is presented below.

The main purpose of K-Means algorithm is to cluster the whole samples based on attributes into k partitions. It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data. It assumes that the object attributes form a vector space. The objective is to make the center to achieve minimize total intra-cluster variance, or, the squared error function.


$$Var = \sum_{i=1}^k \sum_{x_j \in S_i} |x_j - \mu_i|^2, \quad (3-3)$$

where there are k clusters S_i , $i = 1, 2, \dots, k$. μ_i is the center or mean point of S_i . We use the Lloyd's algorithm which is the most common form of the algorithm using an iterative refinement heuristic. Lloyd's algorithm consists of two steps. At first, the input points are partitioned into k initial sets. It may use either at random or using some heuristic data. Secondly, it calculates the mean point, or center, of each set. The second step will constructs a new partition by associating each point with the closest center. Then take the new clusters into first step, and algorithm repeated by alternately

applying of these two steps until convergence, which is obtained when the points no longer switch clusters (or alternatively centroids are no longer changed). Lloyd's algorithm and k-means are often used synonymously. In reality, Lloyd's algorithm is a heuristic for solving the k-means problem. However, with certain combinations of starting points and centroids, Lloyd's algorithm can in fact converge to the wrong answer (A different and optimal answer to the minimization function above exists.)

3.2.2 Training Algorithm

We refer the DSCL algorithm in the [17] to build the codebook of codebook matching algorithm, and the steps of training algorithm is list below.

- **Step 1 :**

Initialization I

Set $V(0) = \{V_j(0); j = 1, 2, \dots, N\}$ with K-means algorithm, and

$$D(0) = \infty, D_j(0) = 1, r = 0.$$

- **Step 2 :**

Initialization II

Set $t = 0$

- **Step 3 :**

Compute the distortion of each code word.

$$Dis_j = \|X_t - W_j(t)\|$$

- **Step 4 :**

Select the winner: the k^{th} code word

$$Dis_k = \min(D_j(t)Dis_j) \quad j = 1, 2, \dots, N$$

- **Step 5 :**

Adjust the code word for winner.

$$W_k(t+1) = W_k(t) + \varepsilon_k(t)(X_t - W_k(t))$$

- **Step 6 :**

Adjust D_k for winner.

$$\Delta D_k = \frac{N_k}{t+1} \|W_k(t) - W_k(t+1)\| + \frac{1}{t} Dis_k$$

$$D_k(t+1) = D_k(t) + \Delta D_k$$

Where N_k is the number of train vectors belonging to region R_k .

- **Step 7 :**

If $t < L$ then $t = t + 1$, and go to step3.

Others go to step8.

- **Step 8 :**

Compute $D(r+1)$.

$$D(r+1) = \frac{1}{L} \sum \|X_i - W\|$$

If $\frac{D(r+1) - D(r)}{D(r)} < \varepsilon$ stop, else $r = r + 1$, then go to step2.

3.3 Deformable Codebook Matching

There are several problems for human detection in indoor environment such as light change or hidden foreground object. When people walk in the indoor environment, it is a common situation that the bottom half body of human is covered by background object. Some examples of this kind of situation are shown in Figure 3-5. The camera is setting on the ceiling and captures the video sequence with an angle of depression. On this angle, we can see the bottom half body of human on the field is blocked by the table or board.



Figure 3-5 : Indoor video.



Figure 3-6 : Outdoor video.

When it comes to outdoor environment, the situations are also the same which is shown in Figure 3-6. When it comes to these situations, most of human recognition system which use the full body features will be failed. Because of this kind of situations, we propose “Deformable Codebook Matching (DCBM)” algorithm to attack the problem of half body of human occurring at first as shown in Figure 3-5 and Figure 3-6. Further more, we use the deformable codebook matching algorithm to do the multiple-occlusive human detection. This part will be also present after we finish describing the DCBM algorithm.

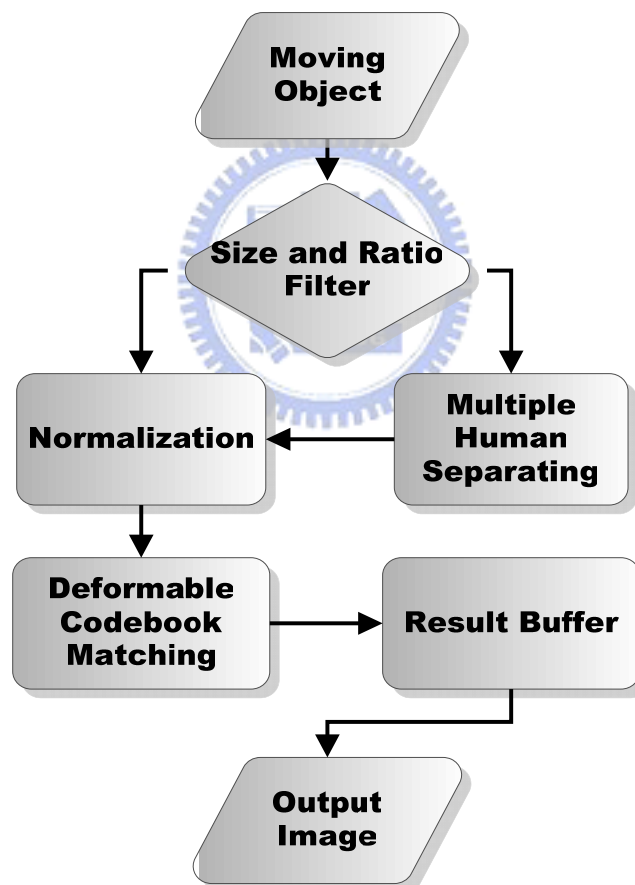


Figure 3-7 : The flow chart of deformable codebook matching.

Figure 3-7 is the flow chart of deformable codebook matching algorithm. The ratio filter tells us which detecting algorithm we should execute, full body matching

or first half body matching or multiple human matching. The result buffer can receive the result from object tracking table to correct the temporary result.

3.3.1 Full Body Matching

As we discussed in the previous section, we use the height and the width of the object to decide which matching algorithm we should use. The table 2 shows this process, where $R_{H/W}$ is the ratio between the height and the width of the object.

Table 2 : H/W ratio selecting table.

Full Body Matching	$1.5 \leq R_{H/W} < 2.5$
Half Body Matching I	$1.2 \leq R_{H/W} < 1.5$
Half Body Matching II	$0.9 \leq R_{H/W} < 1.2$
Multiple People detection	$0.65 \leq R_{H/W} < 0.9$

The full body matching is the default matching algorithm. We use all information of the feature code word to distinguish human from the other objects. The distortion Dis between feature word and code word in the codebook is defined in Equation 3-1. And Dis_{min} defined in Eq 3-2 is the minimum of the distortion with N code words. If the value of Dis_{min} is smaller than the threshold we set, the object is human, otherwise, it is not human. This part we already explain in the section 3.1.

The threshold of this matching algorithm is defined by the testing result. After training procedure, we use another two hundred testing samples to test the codebook we built and find the suitable threshold for the codebook. The testing result is shown in Figure 3-8. We take the value in the intersection as the value of threshold. And there is one thing we should notice that the feature word contain two part, shape

information and histogram information. Just as the feature word, there are two thresholds for shape feature word and for histogram feature word.

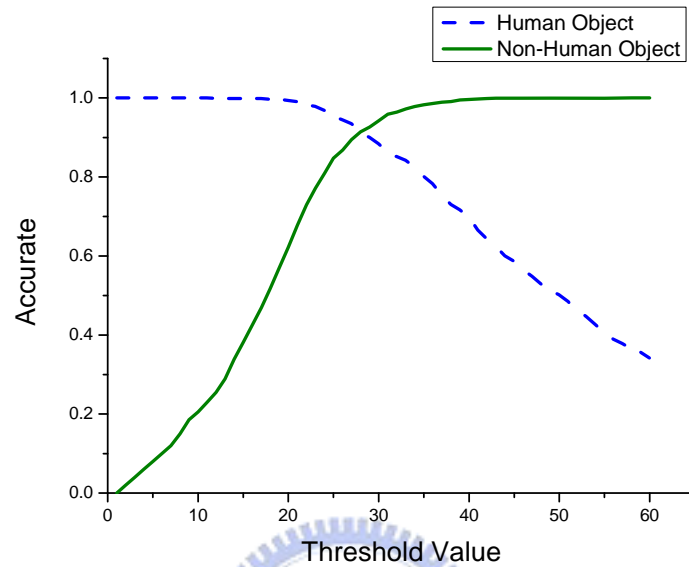


Figure 3-8 : Threshold finding result.

3.3.2 First Half Body Matching

The first half body matching is the most important part of the deformable codebook matching. This matching algorithm is proposed for solving the foreground object covered by background object such as the cases in Figure 3-5 and Figure 3-6. It uses the limited information to classify the moving object. The main idea is very simple. We use the H/W ratio $R_{H/W}$ to decide how many dimensions of data we should use for classification. Table 2 shows the way we select matching algorithm by looking up $R_{H/W}$. We can see there are two level of first half body matching. Actually, we should use the linear function to calculate the percentage of the data we

need, but in order to reduce the computing power and simplify the system, we only use two stages to simulate the overall situations.



(a) People with only half body.



(b) People with full body.

Figure 3-9 : Comparison with full and half body.

When people show in the video with only first half body, we can see in Figure 3-9, it contains a little less information of the human body than full body human object. If we want to remain the performance of detecting rate and not to modify the codebook we already build. The best and the simplest way is to use the information which is not lost when covered or the information which can not be affected by this situation.

The information which will not be lost when the human object is covered by something is the upper half body shape-based features. With this definition, they are the first ten or eighteen shape features of the shape-based part in the feature word. We can see the Table 3. In the ordinary full body codebook matching algorithm, the shape-based information is twenty dimensional data. With decreasing of $R_{H/W}$, we also decrease the number of shape-based feature. And the size of object after normalized needs to be change for the matching procedure too.

Table 3 : Half body matching table.

Level	$R_{H/W}$ Range	Data Type	
		Shape	Histogram
Full Body Matching	$1.5 \leq R_{H/W} < 2.5$	20	10
Half Body Matching I	$1.2 \leq R_{H/W} < 1.5$	16	10
Half Body Matching II	$0.9 \leq R_{H/W} < 1.2$	10	10

There is one thing we should notice, the feature word contain two part, information from shape and information from histogram. As we mention before, the information we need for the half body matching is the feature which can not be affected when covered situation occurs. When we get the feature word from histogram, we also normalize it in to a fixed range. So even if the bottom part of human is covered, the histogram of the human is not change a lot. The threshold for this part is also been modified according the requirement. So the information from histogram can be taken as the invariable feature with covered situation occurring.

After recombination of these two parts of feature word, the only thing we need to do is to compare the feature word with the codebook to obtain the distortion. And this part has been discussed in the section 3.1. The only thing we need to do is to do some little modification in Equation 3-1.

$$Dis_j = \|X - V_j\| = \sum_{i \in S} |X^i - V_j^i| + \sum_{i \in H} |X^i - V_j^i| \quad (3-4)$$

Equation 3-4 is the result after some modification. S is the set of the shape-based feature which we need in the different level of half body matching algorithm. H is the set of all the feature from histogram.

3.3.3 Multiple Occlusive Human Detection

DCBM can also be used to solve some more complex problems, as well as the situation of detecting the human whose bottom half body been covered. For example, we can utilize DCBM to realize the human detection in the situation of multiple-occlusive human. We find out that these two problems have lots of common points. First, the major problem of multiple-human detection is that there are some occlusive part can not be seeing when people walk side by side as shown in Figure 3-10. Second, when we take a close look at non-occlusive part of this two human, we can see the upper half body of these two people are not covered. According to these two common points we consider that we can use the deformable codebook matching algorithm to realize the human detection in detecting the multiple-occlusive human.



Figure 3-10 : Two persons walk side by side.

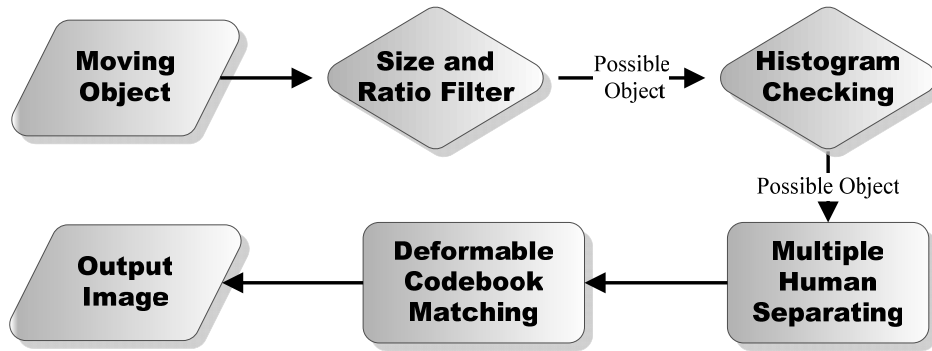


Figure 3-11 : Flow chart of multiple-occlusive human detection.

Figure 3-11 is the flow chart of this procedure. The first step is to select the object which probably is the occlusive object of two or more human. This step contains two moves, one is the size and ration filter and the other is the information from histogram. We use size and ration filter to take appropriate size and H/W ratio which is between 0.65 and 0.9, just as the value in the Table 2. And we use information of histogram to ensure if the moving object contains two or more occlusive blobs. Finally, we use the information of histogram to separate these blobs and then feed into the DCBM algorithm to recognition what these objects are.

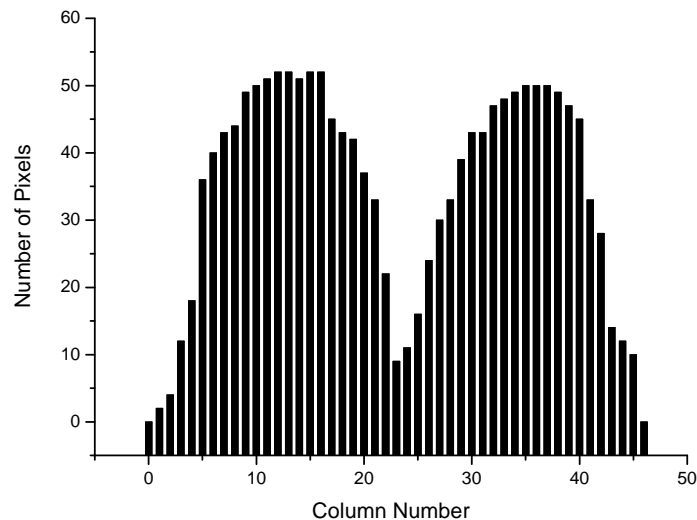


Figure 3-12 : Projection histogram of Figure 3-10

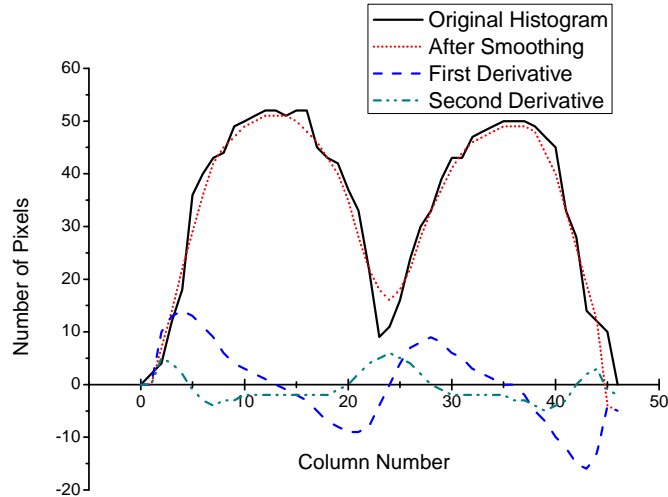


Figure 3-13 : Projection histogram checking and separating.

Figure 3-12 is the projection histogram of moving object in Figure 3-10, we can easily see there are two major blobs joined together, and the height/width ratio is competent. But how does our system know there are two blobs joined together? We use the first order differentiation and second order differentiation of the histogram to ensure the shape of histogram is like a camel's hump. According to Fermat's theorem, let $f : (a, b) \rightarrow R$ be a continuous function and suppose that $x_0 \in (a, b)$ is a local extremum of f . If f is differentiable in x_0 and $f'(x_0) = 0$. So we can find there are three extremums for the histogram except the beginning and ending point in Figure 3-13. And the “second derivative test” tells us if the function is twice differentiable in a neighborhood of a stationary point, then the sign of second derivative can tell us the open side of the camel's hump. By using first and second derivative of histogram we can easily to figure out whether the shape of histogram is right or not. We can use the above procedure to tell the difference between Figure 3-13 and Figure 3-14 which is a histogram of tree waved shadow. And by using the location of stationary points, we can separate the connected blobs by cutting through the location of local minimum.

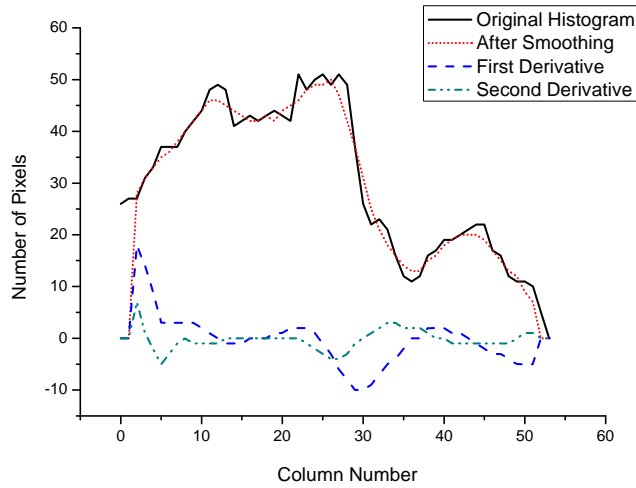


Figure 3-14 : Histogram analysis of a tree waved shadow.

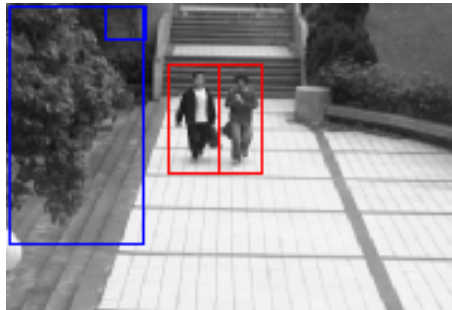


Figure 3-15 : Result of multiple-occlusive human detection.

The vertical red line in the red block in Figure 3-15 is the result of multiple-human separating algorithm. As we can see, the vertical red line makes these two blobs become individual ones. Finally we cut the blob we separated and take the information of first half body. Then we use DCBM algorithm to classify the blob. And this is how the Multiple Occlusive Human Detection algorithm works.

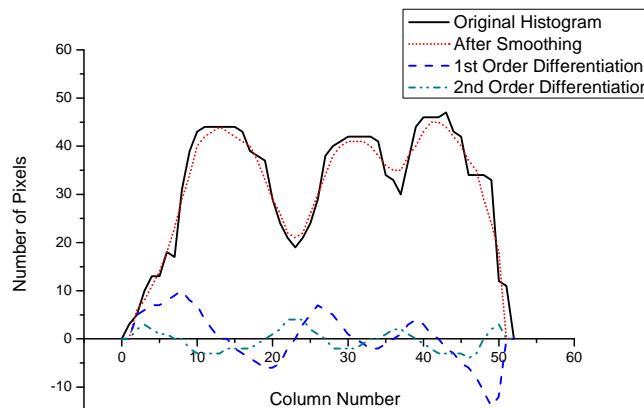
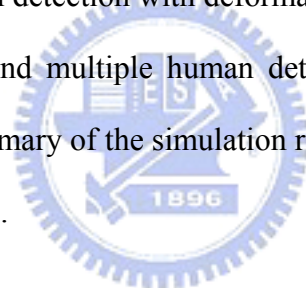


Figure 3-16 : Histogram analysis of three occlusive human

Chapter 4

Experimental Result

In this chapter, we will show the experimental results of our human detection system. First, we discuss the importance of the threshold when doing temporal difference and binarization in the chapter 2 and present the function to determine the optimize threshold. Section 4.1 will show the simulation result of this method. And section 4.2 will present the result of how the noise elimination filter works, and carry out some comparison with the ordinary situation. Section 4.3 presents the most main simulation result of the human detection with deformable codebook matching, DCBM algorithm. Including single and multiple human detection and half body situation. Section 4.4 and 4.5 is the summary of the simulation results. And finally we will make some discussion in section 4.5.



4.1 Simulation Result of Optimize Threshold Finding Algorithm

Recall the function we discussed in Chapter 2, our threshold is based on background gray value. The dark part in the background frame has lower threshold, on the other hand the bright part of background has higher threshold. And the adjustment of threshold is according to the average and standard deviation of current frame and background model. To prevent the heavy computing consumption, this threshold selecting method is activated during background update process. Figure 4-1

is the example of plot of the threshold adjustment. The middle line is TH_0 , the basic part of our threshold, the upper and under line is the threshold modified by the argument β . The two turning points of Figure 4-1 are decided by the stander deviation and the average value of current frame and background model.

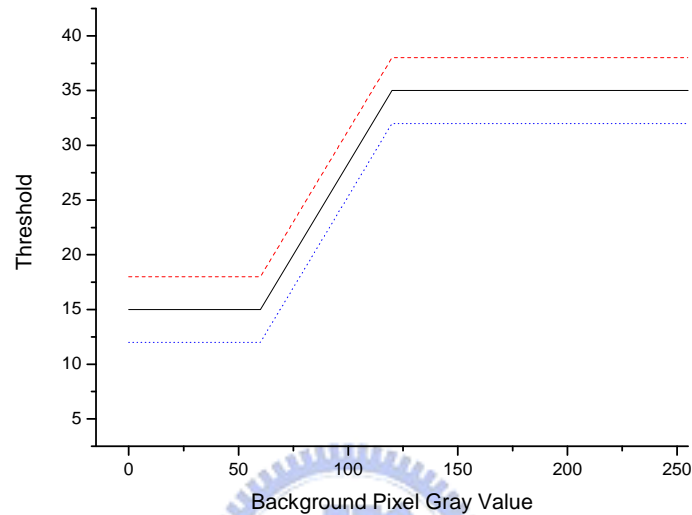
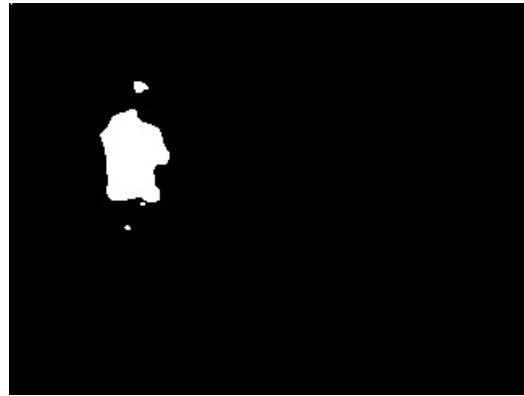


Figure 4-1 : The plot of threshold adjustment.

Figure 4-2(a) is the testing video of our system, the scene of this video is our laboratory but the light is off at all. The only light source is the infrared rays from camera itself. This frame shows the poor luminance and bed situation in the indoor environment. Figure 4-2(b) is the result with utilizing the fixed threshold. We can see that the people's bottom half body in Figure 4-2(b) can not be detected. Figure 4-2(c) is the result with utilizing the regulation 3σ of threshold which is used in several papers with good performance. We can see that it is a little batter than Figure 4-2(b), but it is not as well as our threshold yet. If use our threshold adjustment function, we can see in Figure 4-2(d) not only bottom half body but also the shoulder and head can be seen more clearly. It shows that the higher sensitivity part of our threshold adjustment function works.



(a) Current frame.



(b) With fixed threshold.



(c) With regular 3σ threshold.



(d) With our threshold.

Figure 4-2 : Example of utilization of the threshold adjustment function.



(a) Frame n



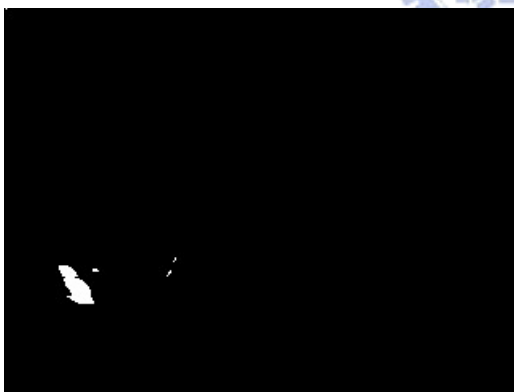
(b) Frame n+1

Figure 4-3 : Scene of Figure4-4.

Figure 4-3 is another testing video of our system, the scene of this video is in an office environment. But the camera is against the light. If the object is near to camera, the diaphragm of camera will change quickly. It causes lots of noise when carrying out background subtraction and binarization. We can see Figure 4-4(a), when we utilize the fixed threshold to carry out the difference, lots of noise will be detected. If we utilize the regulation of threshold to carry out the difference, because the value of difference with larger σ the foreground object can not be extracted.



(a) With fixed threshold.



(b) With regular 3σ threshold.



(c) With our threshold.

Figure 4-4 : Example of utilizing three kinds of threshold.

Then the results using our threshold adjustment is shown in Figure 4-4(c), we eliminate most of noise in the result of image difference. After this procedure, we can use the erosion operator or noise elimination filter to remove the noise remain.

When using the threshold we adjusted, we can segment much clearer object than using the regulation of threshold, and with more robust when encountering high noise.

4.2 Simulation Noise Elimination Filter

When we talked about our noise elimination filter, we mention that the key of our noise elimination process is to put off the timing of using filters till finish of background subtraction. And instead of all kinds of complex filter such as median filter or Gaussian filter, we use only mean filter and get better result. Now we will demonstrate the result of using the filter in the opportune moment.

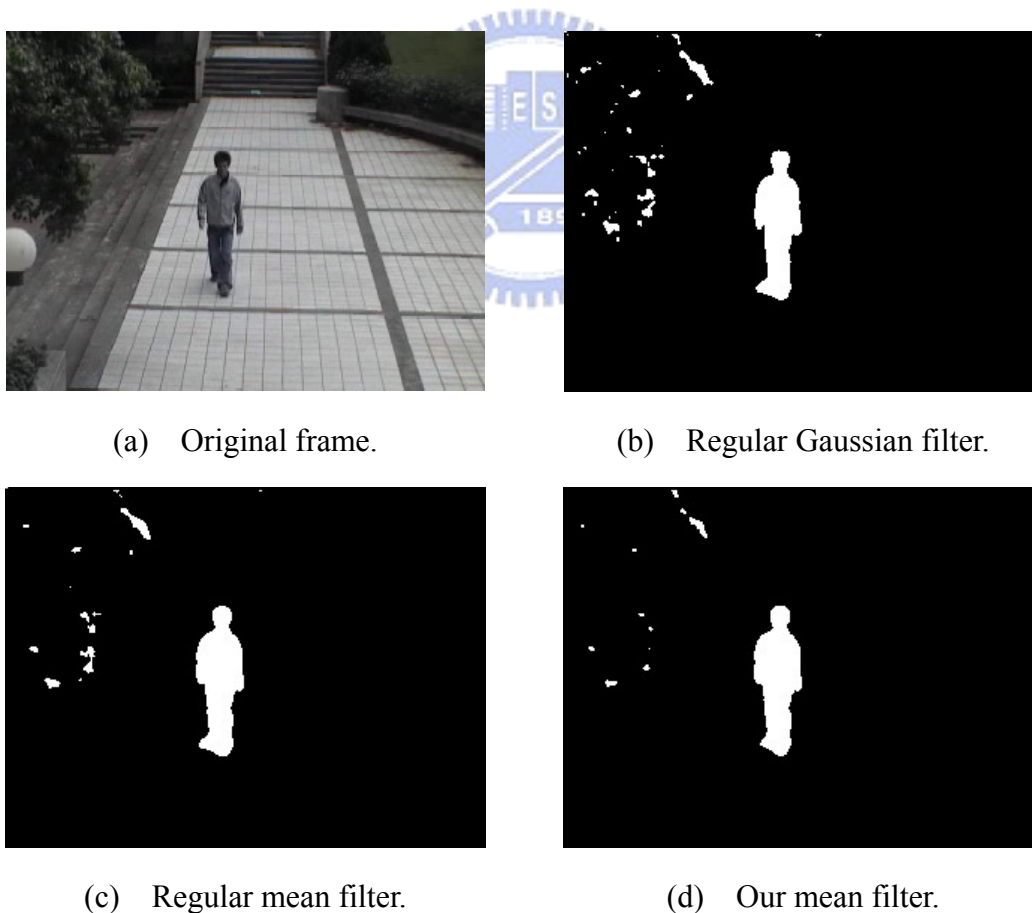


Figure 4-5 : Examples of noise elimination filter.

Figure 4-5 is the comparison of noise elimination filter in common use. Figure 4-5(a) is current frame of video sequence. It is a pavement outside the building with tree waving next the path. Figure 4-5(b) is the result using the five by five Gaussian filter with $\sigma = 1$ before background subtraction, in regular way. Figure 4-5(c) is the result using the five by five mean filter before background subtraction, in regular way too. And finally, Figure 4-5(d) is the result using the five by five mean filter, but after background subtraction. In Figure 4-5, we can easily see that the effect of the algorithm we present has the most effect of eliminating the tree waving. And as we can see, the human object is not affected a lot by the filter. The human object remains its original shape.

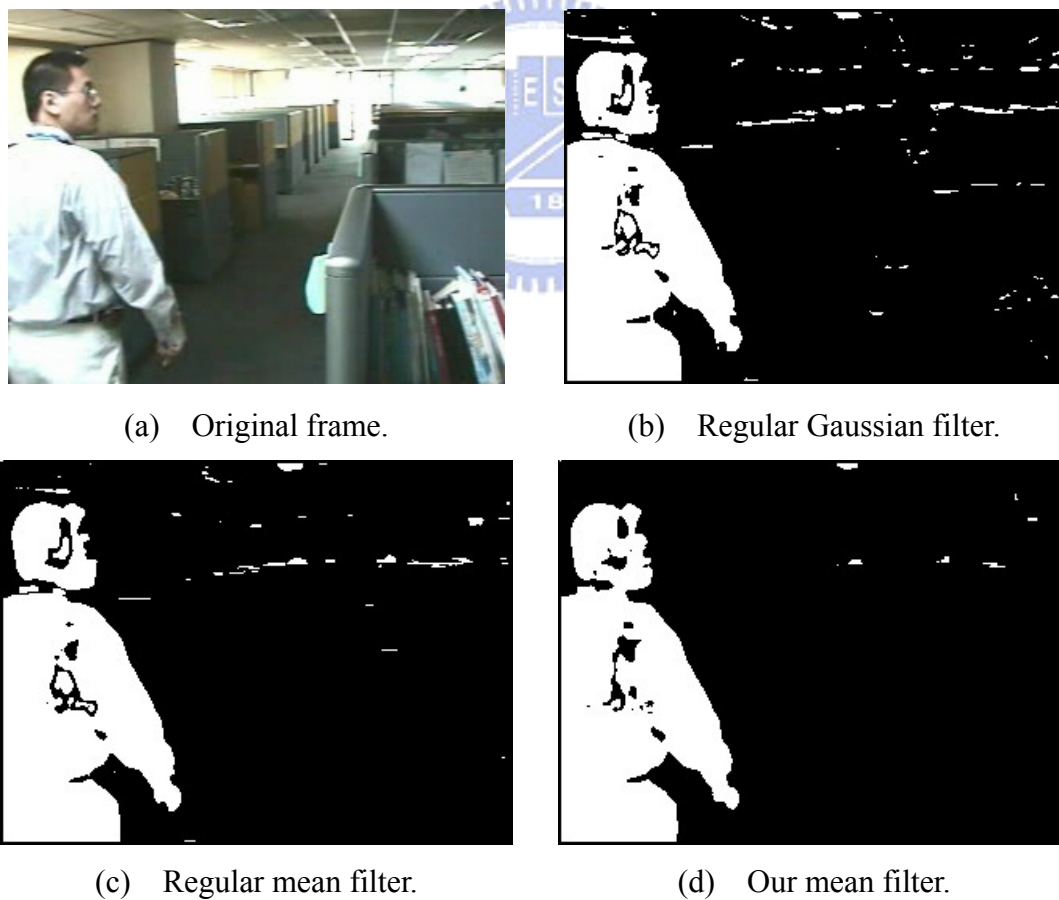


Figure 4-6 : Other examples of noise elimination filter.

Figure 4-6 is the comparison of noise elimination filter in common use too. Figure 4-6(a) is current frame of video sequence. It is a indoor office with strong light change. Figure 4-6(b) is the result using the five by five Gaussian filter with $\sigma = 1$ before background subtraction, in regular way. Figure 4-6(c) is the result using the five by five mean filter before background subtraction, in regular way too. Figure 4-6(d) is the result using the five by five mean filter, but after background subtraction. In Figure 4-5, and Figure 4-6 we can easily see that the algorithm we present works on not only the background effect object effect but also the light change.

4.3 Simulation Result of Deformable Codebook

Matching

In this section we will show the result of our human and non-human detection system. First of all, we explain the tableau we will see in our system.



Figure 4-7 : Snap shot of the output of our System.

Figure 4-7 is the snap shot of the output of our DSP system, and we can see

there are two cubic blocks in this frame. The upper right red block represents the output of our system. The little block in the frame represents the location of the human detected by the system. The upper right block is the statistics of the result after a span. Because of the display of our DSP system can not show too many information in the current frame, some result will be shown by the PC. Figure 4-8 shows this kind of result, the blue block shows the non-human object in the current frame, and red block indicate the human object we detected.

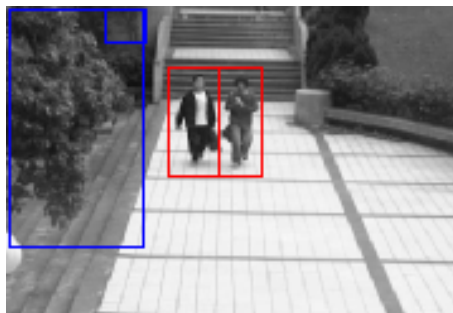


Figure 4-8 : Snap shot of the output of our system on PC.

4.3.1 Human and Non-Human Detection

First we show the result in indoor environment. Figure 4-9 is the result of front and lateral human object detection shown in the scene in the normal indoor environment.



(a) Front side of human detection



(b) Lateral side of human detection.

Figure 4-9 : Results of normal indoor environment.

We also show the result of front and lateral human detection by our system in the normal outdoor environment in Figure 4-10. Figure 4-10(a) shows the front-side of human and Figure 4-10(b) shows the lateral -side of human detected by our system. We can see trees waving in the boundary of the video sequence in Figure 4-10, and they can't affect our system.



(a) Front Side of Human Detection (b) Lateral Side of Human Detection

Figure 4-10 : Result of normal outdoor environment.

Figure 4-11 and Figure 4-12 will show the non-human object shown in the scene including indoor and outdoor situation.



(a) Result with non-human object (motorcycle).



(b) Result with non-human object (car and motorcycle).



(c) Result with non-human object (dog).

Figure 4-11 : Results of non-human object shown in the outdoor environment.



(a) Background frame.



(b) Background frame.



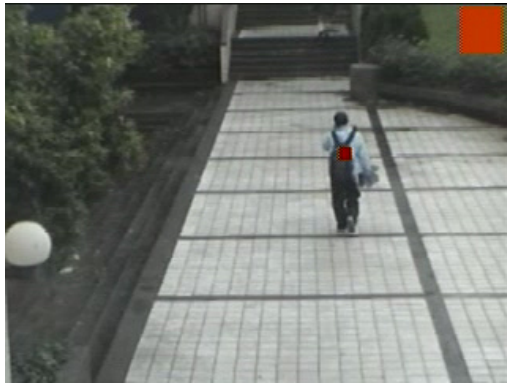
(c) Moving chairs.



(d) Automatic valve.

Figure 4-12 : Results of non-human object shown in the indoor environment.

Figure 4-13(a) shows the human with bag or carries something with hand. Figure 4-13(b) shows the human running through the path. Figure 4-14 shows the multiple human detection including indoor and outdoor situation. Figure 4-15 shows the moving objects with human and non-human objects at the same time.



(a) People carries something.



(b) People running.

Figure 4-13 : Results of complex human detection.



(a) Multiple-human detection outdoor.

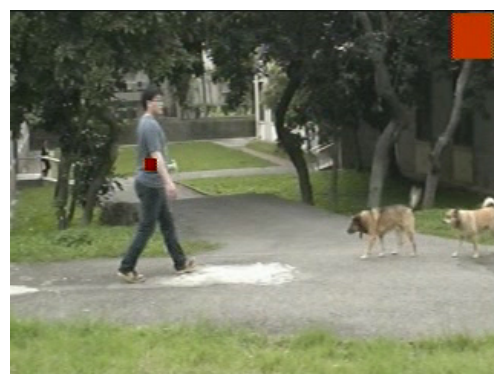


(b) Multiple-human detection indoor.

Figure 4-14 : Results of multiple human detection.



(a) Human and car in one frame.



(b) Human and animal in one frame.

Figure 4-15 : Moving objects with human and non-human objects.

Figure 4-16 is the scene with the lightless environment. In Figure 4-16(b) we can see the trees waving in the boundary.



(a) Infrared rays light source.



(b) Dark and windy situation.

Figure 4-16 : Results of complex human detection.

4.3.2 First Half-Body



(a) Front.



(b) Lateral.

Figure 4-17 : Results with only half-body (indoor).



(a) Covered by car.



(b) Covered by car.



(c) Covered by background object.



(d) Covered by background object.

Figure 4-18 : Results with only half-body (outdoor)

The figures above shows that how the DCBM algorithm works. Figure 4-17 simulates lag and bottom-body covered in the indoor environment, and we still can detect the human pass through. Figure 4-18 shows the result in the outdoor environment. And of course, the human in the video is cut from waist.

4.3.3 Object Tracking Table

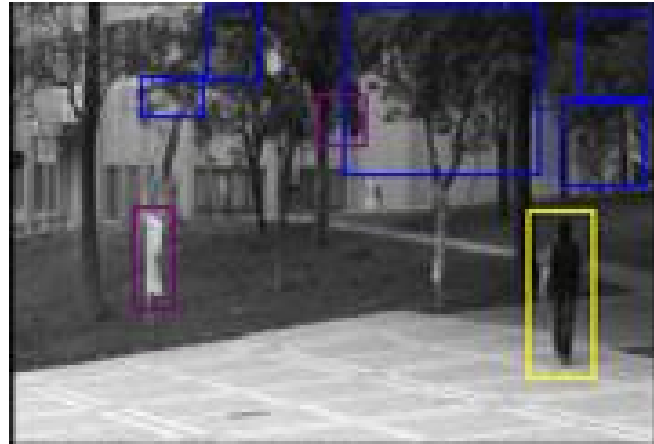
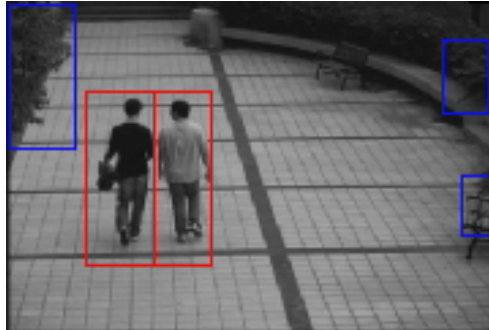


Figure 4-19 : Result of object tracking table.

The main purpose of object tracking table is to prevent the false alarm of codebook classification. The major target is the background object. Sometimes the tree or flag waving just like the human body, and they probably make the system do some wrong decisions. The object tracking can prevent this kind of false alarm. Here we use the PC snap shot of the output, because it is easy to explain how it work. The blue block in Figure 4-19 is the non-human object. The yellow one represent the human object after statistics, it means that the object is always detected as human in a period of time. On the other hand, the purple block is the object which does not move too much, and detected as non-human most of time, then we drive it out to prevent the false alarm.

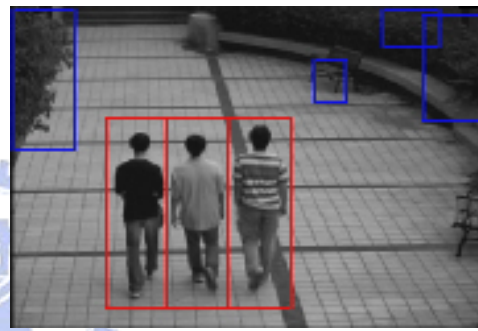
4.3.4 Multiple Occlusive Human Detection



(a) One of results.



(b) One of results.



(c) One of results.

Figure 4-20 : Results of multiple-occlusive man detection.

We already explain the way we detect two or three people walk side by side. The results are shown in Figure 4-20. As we explain before, the blue block shows the non-human object, and the red block indicate the human object. The vertical line in the red block is creative by multiple-human detection algorithm. As we see in Figure 4-20, when people walk shoulder by shoulder, the algorithm we proposed can separate them and running DCBM algorithm for human detection.

4.4 Testing Environment

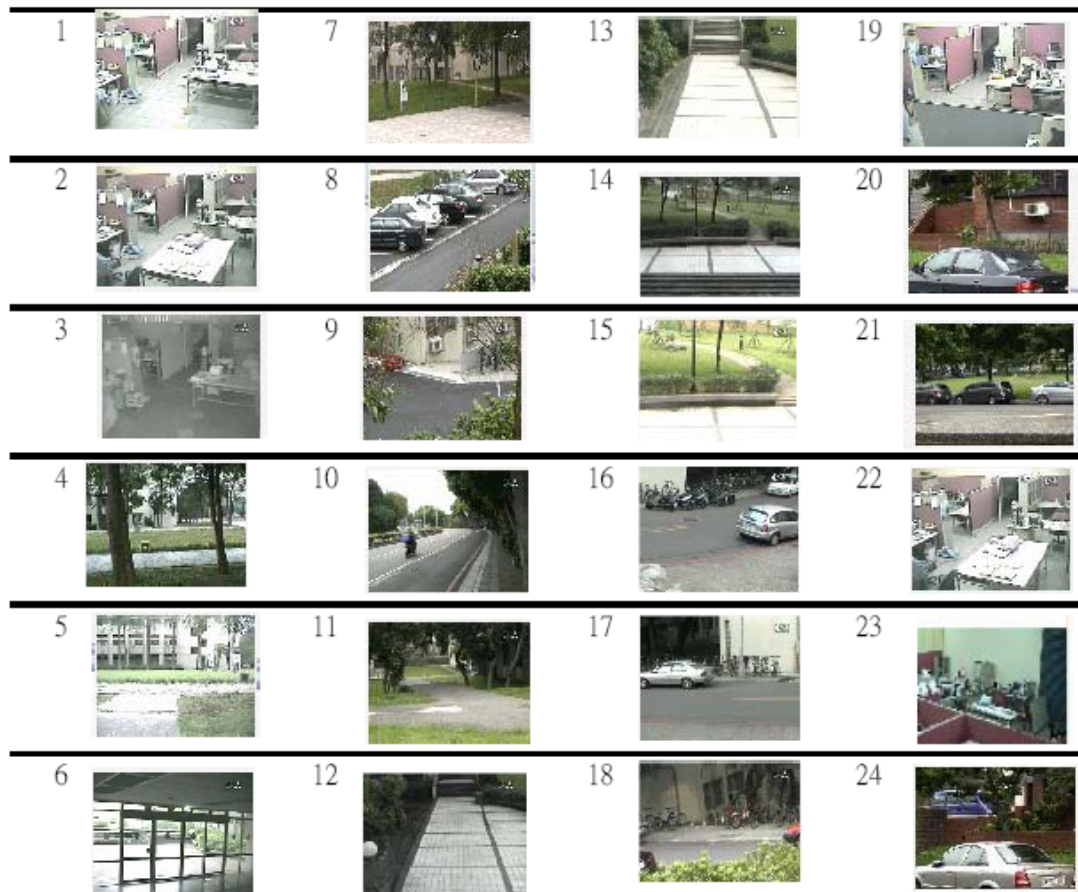






Figure 4-21 : List of testing environment.



We select twenty-four scenes for testing, and more than thirty testing videos. The list of testing environments is shown in Figure 4-21. We have indoor, outdoor and night scenes. The result of these scenes is recorded below.






4.5 Accuracy of DCBM algorithm

Table 4 is the testing result of the twenty-four scenes list above. The column marked as human is the recognition result of human object. The row marked as positive means that we recognize the human object as human and the row marked as negative means that we recognize the human object as non-human one. The column marked as non-human is the recognition result of non-human object. The row marked as “CBM test” is the testing result of the algorithm with only normal codebook matching, and the “DCBM Test” row is use our deformable codebook matching algorithm to recognition the foreground objects appeared in the scene.




Table 4 : The accuracy of our system.




	Scene 1	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	21	0	100 %	5	0	100 %
DCBM	21	0	100 %	4	1	80 %	
	Scene 2	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	3	0	100 %	4	1	80 %
DCBM	3	0	100 %	3	2	60 %	
	Scene 3	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	11	0	100 %	1	0	100 %
DCBM	11	0	100 %	1	0	100 %	
	Scene 4	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	1	2	33 %	3	0	100 %
DCBM	1	2	33 %	3	0	100 %	

	Scene 5	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	9	2	82 %	2	0	%
	DCBM	8	3	73 %	2	0	%
	Scene 6	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	4	0	100 %	5	1	83 %
	DCBM	4	0	100 %	6	0	100 %
	Scene 7	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	2	0	100 %	5	0	100%
	DCBM	2	0	100 %	5	0	100 %
	Scene 8	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	3	0	100 %	2	1	67 %
	DCBM	3	0	100 %	2	1	67 %
	Scene 9	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	4	0	100 %	10	0	100 %
	DCBM	4	0	100 %	9	1	90 %
	Scene 10	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	3	0	100 %	50	5	91 %
	DCBM	3	0	100 %	48	5	91 %
	Scene 11	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	4	0	100 %	9	0	100 %
	DCBM	4	0	100 %	9	0	100 %
	Scene 12	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	15	0	100 %	9	0	100 %
	DCBM	15	0	100 %	10	1	91 %
	Scene 13	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	36	1	97 %	28	2	93 %
	DCBM	32	5	86 %	26	3	90 %

	Scene 14	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	11	2	85 %	8	0	100 %
	DCBM	13	0	100 %	8	0	100 %
	Scene 15	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	18	0	100 %	12	0	100 %
	DCBM	18	0	100 %	12	0	100 %
	Scene 16	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	7	2	78 %	5	0	100 %
	DCBM	5	4	56 %	3	4	43 %
	Scene 17	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	9	0	100 %	2	0	100 %
	DCBM	7	1	88 %	1	1	50 %
	Scene 18	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	6	0	100 %	6	0	100 %
	DCBM	6	0	100 %	5	1	83 %

(a) Result of video with only full body human object.

	Scene 19	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	7	5	58 %	1	1	50 %
	DCBM	12	0	100 %	2	0	100 %
	Scene 20	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	8	5	62 %	5	0	100 %
	DCBM	12	1	92 %	4	1	80 %
	Scene 21	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	10	6	63 %	4	2	67 %
	DCBM	15	1	94 %	6	0	100 %

	Scene 22	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	6	4	60 %	2	0	100 %
	DCBM	10	0	100 %	2	0	100 %
	Scene 23	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	7	4	64 %	5	0	100 %
	DCBM	10	1	91 %	4	0	100 %
	Scene 24	Human		Accuracy	Non-Human		Accuracy
		Positive	Negative		Positive	Negative	
	CBM	6	6	50 %	4	2	67 %
	DCBM	11	1	92 %	6	0	100 %

(b) Result of video with half body human object.

Table 5: The statistic of the accuracy (Video with only full body human).

Summary	Human		Accuracy	Non-Human		Accuracy
	Positive	Negative		Positive	Negative	
CBM Test	167	9	94.8 %	166	10	94.3 %
DCBM Test	160	15	91.4 %	157	20	88.7 %

Table 6: The statistic of the accuracy (Video with lots half body human).

Summary	Human		Accuracy	Non-Human		Accuracy
	Positive	Negative		Positive	Negative	
CBM Test	44	30	59.4 %	24	5	80.7 %
DCBM Test	70	4	94.5 %	24	1	96.0 %

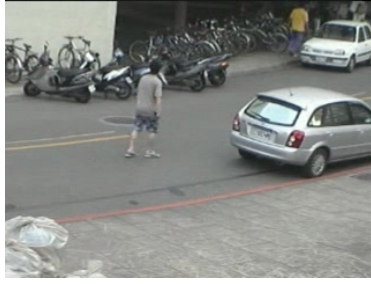
Table 7: The average of accuracy above.

	Average Accuracy (Human)	Average Accuracy (Non-Human)
CBM Test	77.2 %	87.5 %
DCBM Test	93.0 %	92.3 %

4.6 Discussion

First, Table 5 to Table 7 shows that the accuracy of our system is more than ninety percent, and we think it is enough for a warning system. There are something need to be explained about Table 5, Table 6 and Table 7. The first eighteen scenes are the normal testing videos which mean they don't have human which only have first half body shown in the video. The other six scenes are including lots of testing sample which only human's first half body can be seen. In this way, we can see the advantage of our DCBM algorithm. Table 5 is the result of full body matching algorithm. In Table 5 although the accurate of DCBM algorithm is a little lower than normal codebook algorithm, but when there are some human are covered in the video, the accuracy of normal codebook algorithm is become not acceptable. We can find this situation in Table 6. But our DCBM algorithm obviously can take this test. In Table 7 we can see the accuracy of our DCBM algorithm keeps the accuracy when the general situation.

Of course, there still some situation may cause the system fail. Figure 4-22 is the example of system fail. Sometimes the color of people dressing is too close to background, it may cause the background subtraction failed and cut the object by half. It is shown in Figure 4-22(a). There is a fix size of the object after normalization, if the object in the video is much smaller than this size, the feature is no longer valid for system. In this situation, we may take this object as non-human object. Example is Figure 4-22(b).



(a)



(b)

Figure 4-22 : Example of system fail #1.



(a)



(b)

Figure 4-23 : Example of system fail. #2

Because of the codebook classification is shape-based classification the shape of object is the major feature for classification. The shape of motorcycle passing through is just like the shape when people walking. There are some objects of people riding motorcycle is classified to human object, the example is shown in Figure 4-23(a). Sometimes, the shape of tree waving object is not predictable, sometimes it will be took to be the human, the example is shown in Figure 4-23(b). Although we have multiple-human detection algorithm, but when people walk as a group like Figure 4-24, it can not be detected by our system.



Figure 4-24 : Example of system fail. #3

Chapter 5 Conclusion

In this thesis, a fast real-time human detection system with low computing power is proposed. The first part of our human detection system is to segment the moving object from the scenes. We use the background subtraction here to segment the moving blob. We provide a simple and fast function to calculate the binarization threshold for the varying environments and videos taken by different cameras. In second part of our system, we use simple trajectory tracking and condition judgment to provide some data for human detection algorithm and to decrease the false-alarm rate. The final part is human detection. Because of the requirement of low computing power, we choose the shape-based method, and the codebook by training to classify human being from the other objects. The people walking indoor are sometimes covered by furniture such as desks or chairs. To solve this kind of problem, we provide Deformable Codebook Matching, a human detection algorithm for first half body with different height/width ratio. With Deformable Codebook Matching, when someone's bottom half body is covered, the system can still work. Further, we use Deformable Codebook Matching to implement the human detection for multiple people walking side by side.

The contribution of our thesis is listed below:

1. We realize the human detection on the DSP platform. It is easily to build up and the cost is cheaper.
2. The optimized threshold adjustment algorithm and novel usage of noise elimination filter make the system work in the scene with only infrared ray as light source, or in the scene with changeful luminance.

3. DCBM algorithm can provide the higher accurate in the scene that foreground object may be covered by some background object. And DCBM algorithm also makes the multiple-human detection passable.

Although the result of our system is fine, but there still some shortcomings need to be solved. The background subtraction method is not a good solution when light source is changeful, or the camera is against the light source. Although the optimized threshold adjustment algorithm and novel usage of noise elimination filter we present work, the nature shortcoming of the background subtraction method still exist. When the situation is over the capability of the system, it's failed either. People in the training sample walk as front or lateral side. When people crawl or carry something which can cover him at all, this kind of situation can not be handle by the system. Although we can add this kind of training samples into our system, the accuracy will be affected by lots of similar things, such as tree waving or dogs. In the multiple-occlusive human detection, the situation we can handle is when people walking shoulder by shoulder, but there are still lots of different situations for multiple-occlusive human walk into the scene. If we want to solve these questions, we must develop more complex algorithm to separate them.

References

- [1] P. J. Burt, J. R. Bergen, R. Hingorani, R. Kolczynski, W. A. Lee, A. Leung, J. Lubin, and H. Shvayster, "Object tracking with a moving camera," in *Proc. of the IEEE Workshop on Visual Motion*, March 1989, pp. 2-12.
- [2] P. H. Batavia, D. E. Pomerleau, and C. E. Thorpe, "Overtaking vehicle detection using implicit optical flow," *IEEE Conference on Intelligent Transportation System*, Nov.1997, pp. 729-734.
- [3] W. J. Gillner, "Motion based vehicle detection on motorways," in *Proc. of the Intelligent Vehicles '95 Symposium*, Sept. 1995, pp.483-487.
- [4] L. Zhao and C. E. Thorpe, "Stereo- and neural network-based pedestrian detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 3, pp. 148-154, Sept.2000.
- [5] C. E. Smith, C. A. Richards, S. A. Brandt, and N. P. Papanikolopoulos, "Visual tracking for intelligent vehicle-highway systems," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 4, pp. 744-759, Nov. 1996.
- [6] R. Polana and R. Nelson, "Detecting activities," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1993, pp. 2-7.
- [7] C. Curio, J. Edelbrunner, T. Kalinke, C. Tzomakas, and W. von Seelen, "Walking pedestrian recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 3, pp.155-163, Sept. 2000.
- [8] C. Orrite-Uruñuela, J. Martínez del Rincón, J. Elías Herrero-Jaraba, G. Rogez, "2D Silhouette and 3D Skeletal Models for Human Detection and Tracking," *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, 2004.
- [9] Z. L. Jlang, S. F. Li, D. F. Gao, "A Time Saving Method for Human Detection in Wide Angle Camera Images," *Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian*, 13-16 August 2006.
- [10] Y. L. Tian and A. Hampapur, "Robust Salient Motion Detection with Complex Background for Real-time Video Surveillance," *Proceedings of the IEEE Workshop on Motion and Video Computing (WACV/MOTION'05)*, August 2005.
- [11] A. Elgammal, R. Duraiswami, D. Harwood, And L. S. Davis, "Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance," *Proceedings Of The IEEE*, Vol. 90, No. 7, July 2002.

- [12] K. Kim, T. H. Chalidabhongse, D. Hanuood, L. Davis, "Background Modeling And Subtraction By Codebook Construction," *2004 International Conference on Image Processing (ICIP)*, March 2004.
- [13] L. Zhao and C. Thorpe, "Stereo- and Neural Network-Based Pedestrian Detection," *Proceedings Of The IEEE*, 1998.
- [14] S. M. Yoon and H. Kim, "Real-time multiple people detection using skin color, motion and appearance information, " *Proceedings of the 2004 IEEE International Workshop on Robot and Human Interactive Communication Kurashiki, Okayama Japan*, September 20-22,2004.
- [15] D. Tan, K. Huang, S. Yu, T. Tan, "Efficient Night Gait Recognition Based on Template Matching," *The 18th International Conference on Pattern Recognition (ICPR'06)*, 2006.
- [16] A. Gersho, "Asymptotically Optimal Block Quantization," *IEEE Transactions On Information Theory*, Vol.It-25, No.4, July 1979.
- [17] J. Zhou and J. Hoang, "Real Time Robust Human Detection and Tracking System," *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* 2005.
- [18] C. Curio, J. Edelbrunner, T. Kalinke, C. Tzomakas, and W. von Seelen, "Walking pedestrian recognition," *IEEE Trans. Intelligent Transportation Systems*, vol. 1, issue 3, pp.155-163, Sept. 2000.
- [19] H. Roh, S. Kang and S-W Lee, "Multiple People Tracking Using an Appearance Model Based on Temporal Color," in *15th International Conference on Pattern Recognition (ICPR'00) - Volume 4*, 2000.
- [20] D. Zhang and G. Lu, "A Comparative Study of Fourier Descriptors for Shape Representation and Retrieval," *the 5th Asian Conference on Computer Vision*, Jan. 2002.
- [21] I. Haritaoglu, D. Harwood and S. Davis, "W⁴: Real-Time Surveillance of People and Their Activities," *IEEE Trans. On Pattern and Machine Intelligence*, Vol.22, No.8, Aug. 2000.
- [22] I. Haritaoglu, D. Harwood and S. Davis, "Backpack: Detection of People Carrying Objects Using Silhouettes," *IEEE International Conference on Computer Vision*, pp. 102-107, 1999.
- [23] I. Haritaoglu, D. Harwood and S. Davis, "Hydra: Multiple People Detection and Tracking Using Silhouettes," In *Second Workshop of Visual Surveillance at CVPR*, pages 6-13, 1999.
- [24] T. Zhao and R. Nevatia, "Tracking Multiple Human in Complex Situations," *IEEE Trans. On Pattern and Machine Intelligence*, vol. 26, No. 9, Sep, 2004.

- [25] S. J. McKenna, S. Jabri, Z. Duric, H. Wechsler and A. Rosenfeld, "Tracking Groups of People," *Comput. Vision Image Understanding*, no. 80, pp. 42--56, 2000.
- [26] Y. Kuno, T. Watanabe, Y. Shimosakoda and S. Nakagawa, "Automated detection of Human for Visual Surveillance System," in *IEEE Proceedings of ICPR*, 1996.
- [27] M. Riesenhuber and T. Poggio, "Models of Object Recognition," in *Nature Neuroscience Supplement*, vol. 3, Nov, 2000.
- [28] I. Sekita, T. Kurita and N. Otsu, "Complex Autoregressive Model for Shape Recognition," in *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 14, No4, Apr, 1992.
- [29] H. Kauppinen, T. Seppanen and M. Pietikainen, "An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification," in *IEEE Tran. On Pattern Analysis and Machine Intelligence*, Vol. 17, No. 2, Feb, 1995.
- [30] L. Wang, T. Tan, W. Hu and H. Ning, "Automatic Gait Recognition Based on Statistical Shape Analysis," in *IEEE Trans. On Image Processing*, vol. 12, No. 9, Sep, 2003.

