國立交通大學

電機與控制工程學系

碩士論文

基於改良式獨立成分分析之人物偵測與追蹤

Human Detection and Tracking Based on Modified Independent Component Analysis

研 究 生：鍾采蓉

指導教授：林進燈 博士

張志永 博士

中華民國 九十七 年 七 月

基於改良式獨立成分分析之人物偵測與追蹤

Human Detection and Tracking Based on Modified Independent Component Analysis

研 究 生：鍾采蓉　　Student：Tsia-Jung Chung

指導教授：林進燈 博士　　Advisor：Dr. Chin-Teng Lin

張志永 博士　　Dr. Jyh-Yeong Chang

國立交通大學

電機與控制工程學系

碩士論文

A Thesis

Submitted to Department of Electrical and Control Engineering

College of Electrical Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master

in

Electrical and Control Engineering

June 2008

Hsinchu, Taiwan, Republic of China

中華民國 九十七 年 七 月

基於改良式獨立成分分析之人物偵測與追蹤

學生：鍾采蓉　　　　　　　　指導教授：林進燈 博士

張志永 博士

國立交通大學電機與控制工程研究所

中文摘要

近幾年來，人物偵測及追蹤在電腦視覺中是一項常被深入探討的領域，且其可被廣泛應用在居家照護、保全及病人監控等系統。本論文提出一改良式獨立成分分析技術的人形自動偵測系統。我們用獨立成入分析法抽取辨識特徵，且以條件熵來做爲特徵選擇的依據，以此得到具有良好辨識能力且具有代表性的特徵。強建的支持向量機則爲我們系統中主要的數據分類法。我們的實驗環境包含室內及室外，而監視畫面中的移動物體則有行人、動物及車子等等。

我們使用背景相減法取出畫面中的移動物體。爲了處理複雜背景的情況，使用高斯混合模型來建構背景。針對移動物體被部分遮避的情形，我們提出金字塔型橢圓形頭部偵測法來分離它們。此外，利用簡單的色彩資訊及卡爾曼濾波器進行移動物體的追蹤及動向預測。

Human Detection and Tracking Based On Modified Independent Component Analysis

Student: Tsia-Jung Chung　　Advisor: Dr. Chin-Teng Lin
Dr. Jyh-Yeong Chang

Department of Electrical and Control Engineering
National Chiao Tung University

Abstract

In recent years, video based human detection and tracking are a popular research area, and it has been used in widely applications such as homecare, security, patient monitoring and so on. This paper introduces a human detection system using modified Independent Components Analysis (ICA). The ICA features are selected by conditional entropy and classified by Support Vector Machine (SVM). The proposed system monitors the movement of human, animals or vehicles which across a secured area, and it works well in indoor or outdoor environment.

The background subtraction is used to extract moving objects. In order to handle situations where the background of the scene is cluttered and not completely static but contains small motion, we models the background based on Gaussian mixture model (GMM). In complex situation, the moving object may disappear totally and partially due to occlusion by other objects. A fitting ellipse function based modification pyramid method is used for separating some multi-person occlusion. Our system combines with Kalman filter to estimate motion information and use the information in predicting the appearance of targets in succeeding frames.

致　　謝

兩年的研究生生活即將結束，感謝我的指導教授林進燈博士及張志永博士，在這兩年來悉心指導我，讓我學習到許多寶貴的知識及經驗，使我受益良多。另外也感謝李錫堅、廖弘源等口試委員們的建議與指教，使得本論文得以更加完善。再次感謝老師們對我的一切幫助與指導。

其次，感謝超視覺實驗室的大家長鶴章、剛維及德正學長，博士班的Linda、肇廷、建霆、子貴及勝智等學長姐給予我的建議及指導，尤其是Linda學姐，在最後的分析整合上給了我許多的建議，讓我獲益良多，本論文能順利完成多虧了學長姐們的幫助。也要感謝同學東昱、怡珊、清泉及宏揚，在你們的陪伴下我們一起成長，一起努力。

感謝父母對我的教育，沒有他們就沒有今日的我。感謝家人們默默地支持著、關心著我，讓我有勇氣去追求我所想要的，讓我無後顧之憂。

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

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Contribution

In recent years, video based human detection and tracking is a popular research area, and it has been widely applied in various applications, such as home care, security, patient monitoring and so on. Due to the increase of crime, automatic visual surveillance with computer vision plays an important role in security. The ability to distinguish people from other moving objects such as animals or vehicles, and track them to analysis their behaviors are the important issue.

Human detection and tracking surveillance system is one major issue of surveillance. The main idea is to find whether there are humans in the secured area or not. A outperform human detection system can reveal the number of people in a single image.

Human detection system is generally consists of two parts: the extraction and the human recognition part. When a moving object get into the range of secured area, we need to locate its position, size and even analyze its moving trajectory, this process is called foreground segmentation. However, it is not easy to extract moving object from video perfectly. The problems may be caused by camera shake, rain and structural change by sunlight and shadow effects. Furthermore, groups of people which move together or interact with each other are difficult to be separated and recognized. In this thesis, we resolve above problem by using a fitting ellipse function which depend on the pyramid method, and a tracking system based on Kaman filter to resolve occlusion problem.

The second part is to recognize whether a moving object is a human or not. At this part human detection system extracts human features and uses an advance algorithm of mathematic equation such as neural network to achieve the goal. There are many techniques to find the human features, but some are too complex and take very long time to calculate, and some are not strong enough to exactly distinguish people. In this thesis, a modified independent component analysis (ICA) is proposed to improve the performance of human recognition. ICA is a statistical method that transforming an observed multidimensional random vector into components that are statistically independent. However, the drawback of ICA algorithm is that the class discriminability of independent component is not sorted by the creating sort, and it is not depend on binary classified capability. Therefore, a feature selection method based on conditional entropy is proposed to modify to resolve the drawback of ICA.

1.2 Related Work

Recently 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, 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 in [22], [11] was used to detect vehicle. Zhao et al [12] 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 require at least two cameras and can be used only for short and middle distance detection. Carlos Orrite-Urunuela [24] uses 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, this algorithm following by a point distribution model (PDM) approach using a Principal Component Analysis (PCA) to establish the shape of human. But the system need to use 3D skeletal structure, it needs multiple cameras, and only successful extract whole shape in simple and clean environment. The size of human must be large enough for their algorithm, which is a disadvantage for video surveillance system. The frame differencing is a simple method of moving extraction. Smith et al [13] 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-Line Jiang [25] also used background subtraction method to segment isolate human. To avoid shadow they use the homogenous of shadow and 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 [14]. They firstly use the background subtraction to locate the motion area and 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 $N \sim (\mu,\sigma)$, 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, C. Stauffer and W.E.L. Grimson construct a mixture of Gaussian model. This paper discusses modeling each pixel as a mixture of Gaussians and using an online approximation to update the background [20].

To distinguish human with other object, 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. J. Zhou and J. Hoang construct a codebook by shape information, and use it to classify human being from other objects [21]. But if the boundary of human body is not obvious, for example, partially occlusion, or the human is carrying something with hand, the result may be wrong. Motion Histograms of Oriented Gradients (HOG) extract features from shape information [27]. This algorithm based on approach use Fast Fourier Transform and its periodicity against time [26]. Some system integrates multiple features to recognize human such as shape pattern, motion pattern, skin color, etc. Curio et al [32] 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. In [12], the Back-Propagation Neural 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. Sang Min Yoon [16] used robust skin color, background subtraction and human upper body appearance information. They extracted the human candidate regions using color transform and background subtraction. 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.

1.3 System Overview

Our proposed system is illustrated in Fig. 1-1. The surveillance video data is captured by a static camera. Frame by frame of video data is processes at Gaussian mixture model (GMM) to model a dynamic background, and at the same time the foreground segmentation based background subtraction will extract the moving objects from background. In the complex environment, the moving objects may disappear or partially occluded. We apply fitting ellipse method to separate the moving objects that partially occluded. The Klaman filter used to estimate motion information and use the information in predicting the appearance of targets in succeeding frames. At last In dependent Component Analysis (ICA) is applied to extract the feature of moving object, and classify to human or other object by Support Vector Machine (SVM).

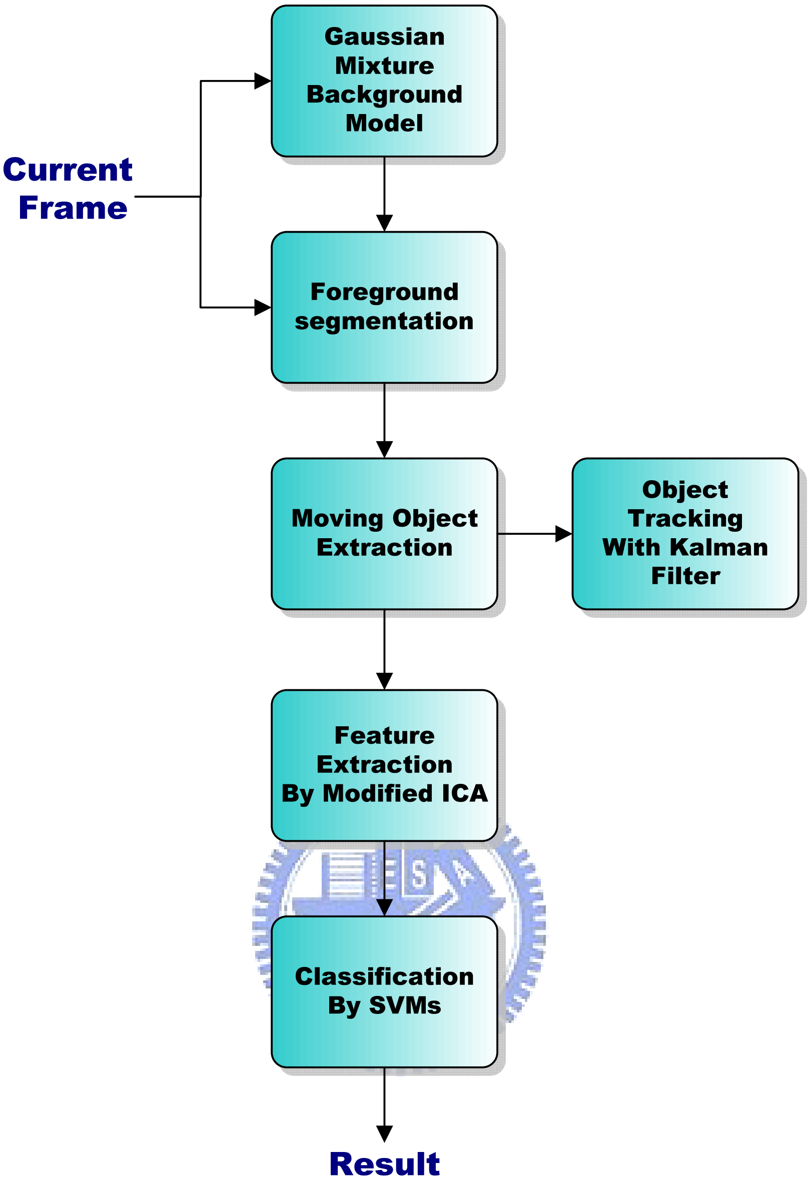

Gaussian Mixture Background Model
Current Frame
Foreground segmentation
Moving Object Extraction
Object Tracking With Kalman Filter
Feature Extraction By Modified ICA
Classification By SVMs
Result

Fig. 1-1 : System architecture overview

1.4 Organization

The remainder of this thesis is organized as follows. Chapter 2 describes the foreground segmentation, moving objects extraction and tracking. Chapter 3 introduces detection system including modified feature extracting and classification ability. Chapter 4 shows the experimental results of our system. Chapter 5 makes the conclusions of this thesis and the future works.

Chapter 2

Object Extraction

In this chapter, we will explain how to extract the moving object from video. The structure of proposed system is consists of three subsystems: foreground segmentation, moving objects extraction and moving object tracking. In section 2.1, we present the framework of moving object segmentation, which included background modeling, shadow removing, and multiple human separating. In section 2.2, we present the framework of objects tracking.

2.1 Moving Object Extraction

2.1.1 Gaussian Mixture Model for Background Subtraction

In the surveillance video, the position of the camera is fixed therefore the background image is stationary. The simplest way to segment moving object from background is by using background subtraction method. If the background model is robust, then the moving object can be extracted completely from background.

We need to build a background which can appropriately update the small motion that changing in time, for example the change of intensity or the motion of tree leaves, etc. If the background model is fixed, the moving object will contain the background regions, it will increase the false detection rate in our next processes. Thus a robust background model is necessary.

For above reasons, a robust Gaussian mixture model (GMM) [20] is chosen to construct a dynamic background model. The process is shown in Fig. 2-1. The temporal difference is used to extract the possible background regions, and the background model is constructed by GMM algorithm. In the GMM algorithm, the background can be modeled as Gaussian distribution $N \sim (\mu, \sigma)$. This basic Gaussian model can adapt to gradual light change by recursively updating the model using an adaptive filter.

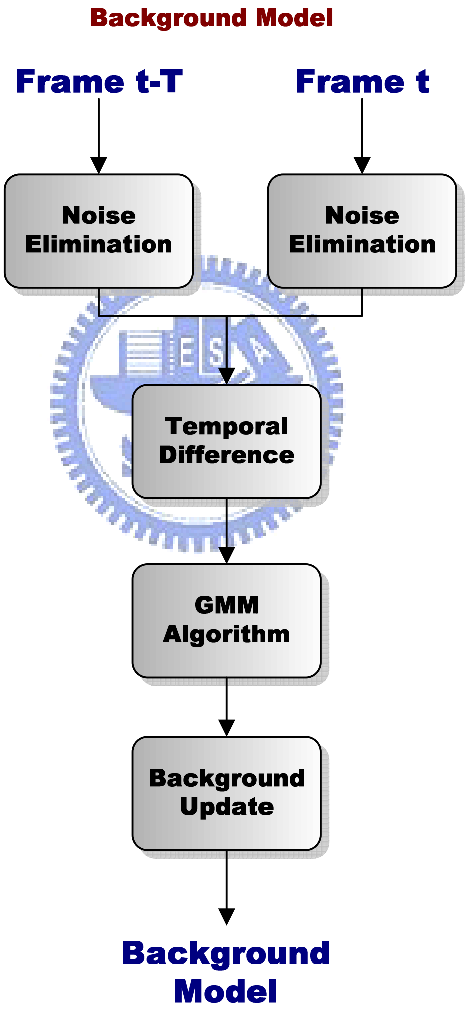


Fig. 2-1 : Background model construction

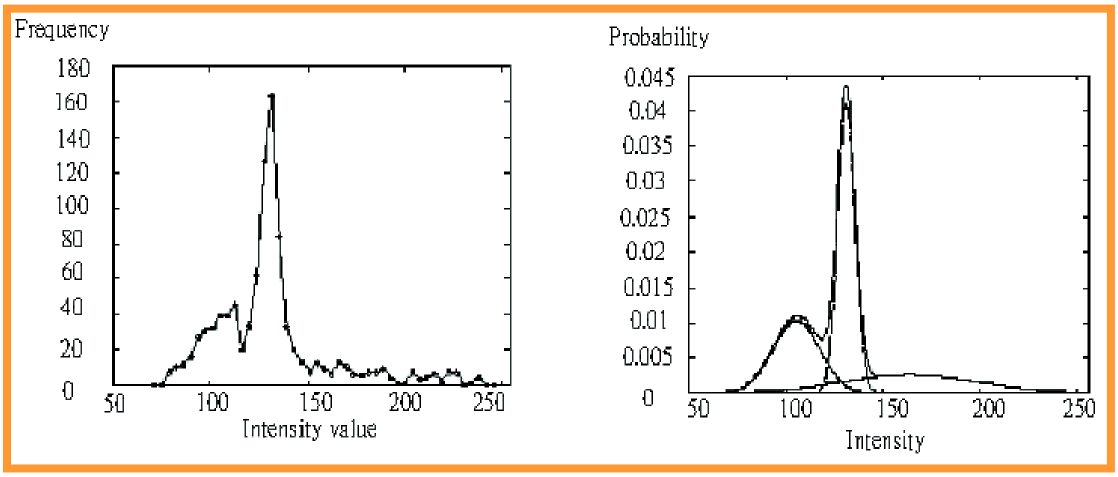


Fig. 2-2 : Mixture of Gaussian

An example of mixture of Gaussian is depicted in Fig. 2-2, there are three different Gaussian distributions.

We build a GMM for each pixel and update them over time, if the update time is take a long time, then the background model will be more stable. In our experimental, we use three Gaussians to construct the background model, but for the complex environment, it may need more than three number of Gaussians.

When one frame input, we determine where pixel in current frame is in a background region or not. Because the color of moving objects has a larger variance than the background, the determining threshold is set by the variance of each Gaussian background model.

After we obtain the background model, we can extract the moving object from background by using background subtraction. In this process, we only use the luminance. It will decrease the computing power and make the issue easier than using colorful image. The whole system of foreground segmentation is shown as Fig. 2-3.

There are some noises or error information is obtained from segmentation process. We classify this noise into two major types, one is the noise like the photosensitivity camera noise or due to light suddenly change. The other type is shadow effect. In order to obtain accurate object region, we must reduce these noise.

To reduce the noises, a low-pass filter can be implemented in the preprocess step, in other hand it was at the first step before the images process at the GMM and foreground segmentation. In our experiment, we compare the two of spatial low-pass filter: mean filter and Gaussian filter. Our purposed is to reduce noises and do not want to lose the boundary or shape information. We choose the Gaussian smoothing operator in our system. Because the mean filter will eliminate some of original edge information. The results of two filters are shown in Fig. 2-4. We can see that the foreground extracted by Gaussian filter is completer than extracted by mean filter. And a 3X3 filter is enough for our experimental environment.

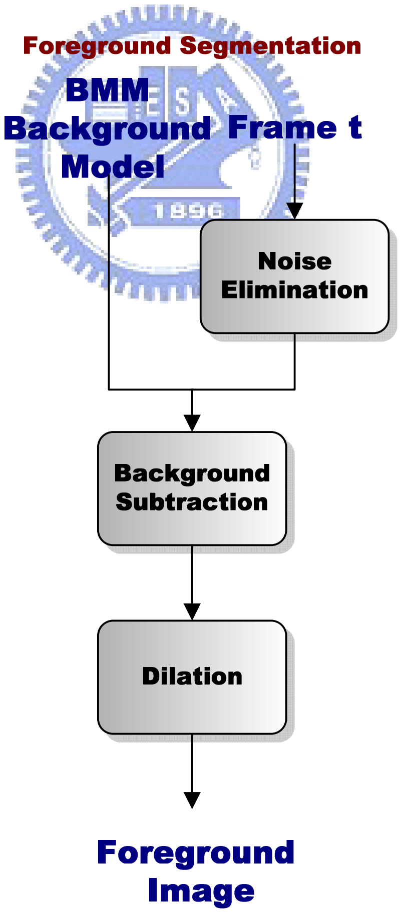
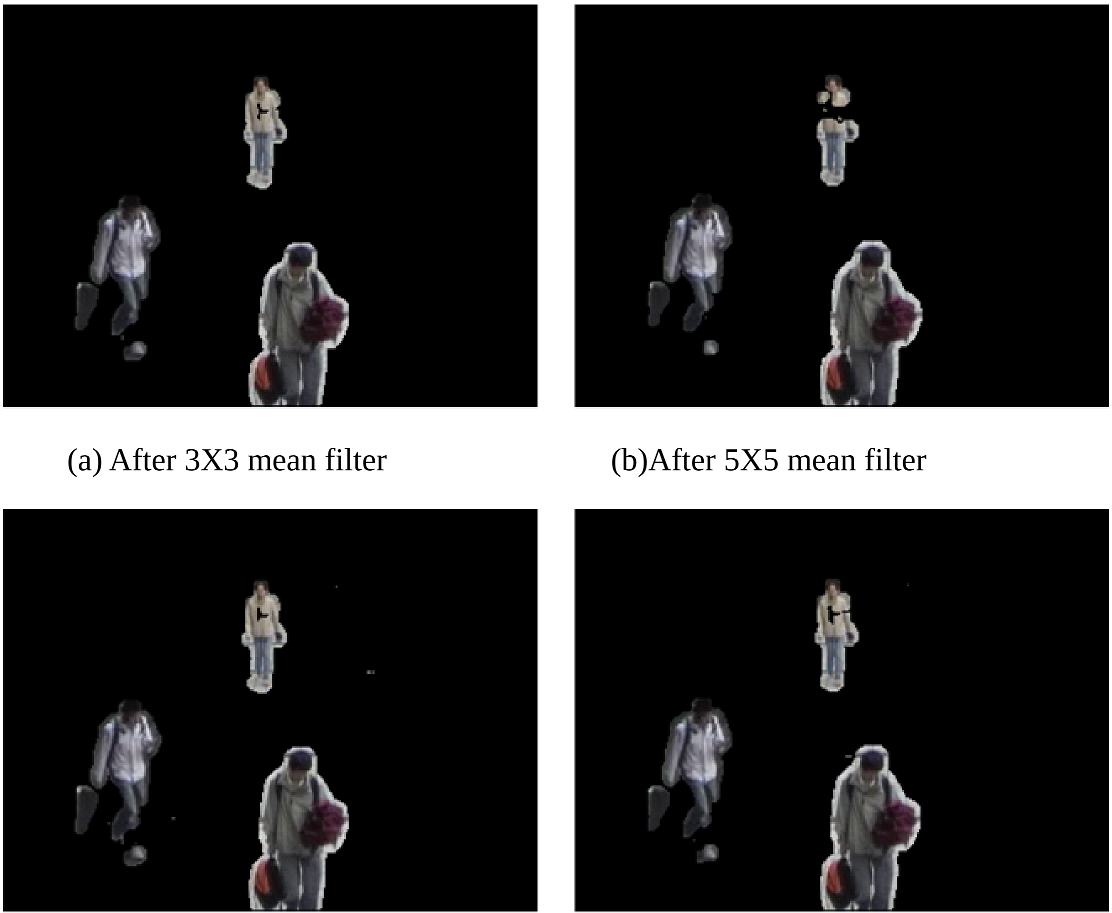


Fig. 2-3 : Foreground segmentation processing diagram



(a) After 3X3 mean filter

(b)After 5X5 mean filter

(c) After 3X3 Gaussian filter

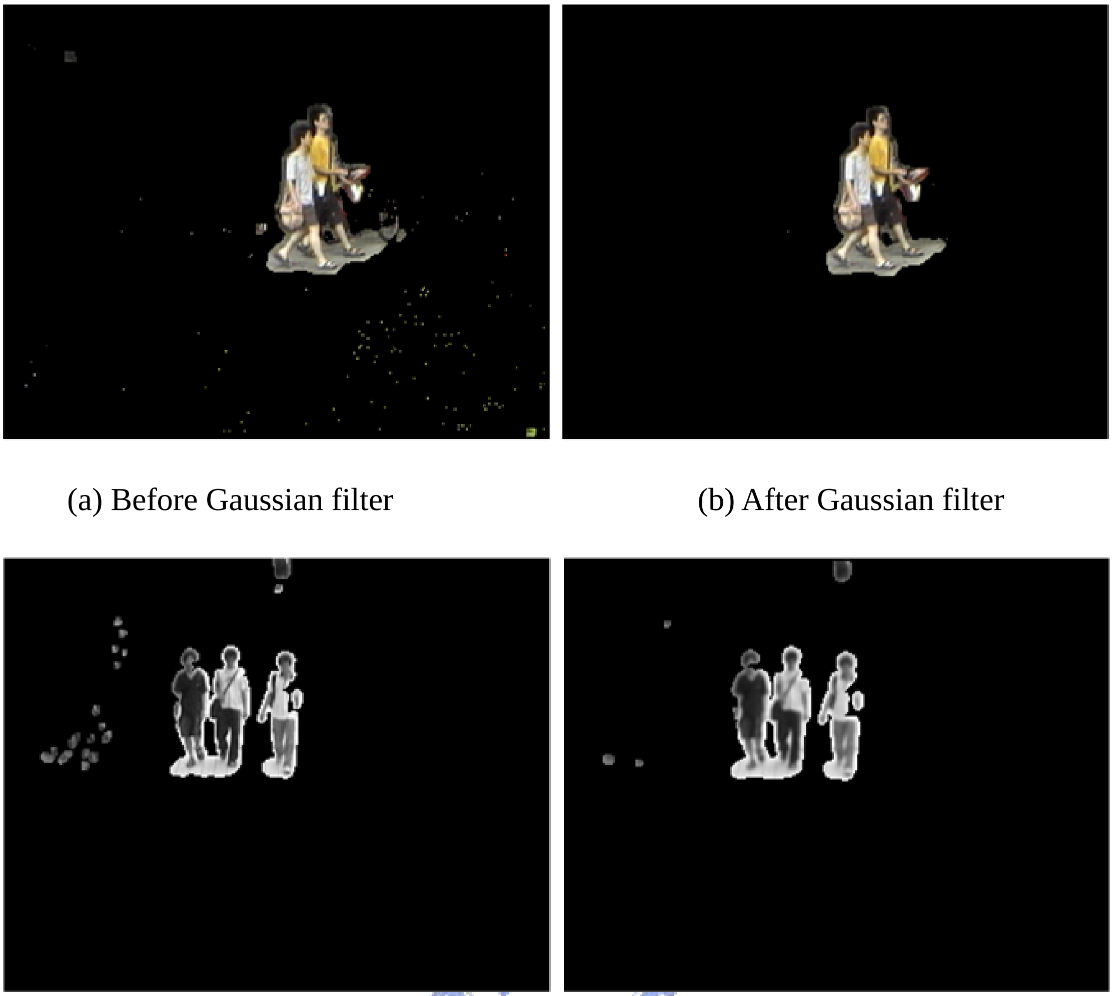
(d) After 5X5 Gaussian filter

Fig. 2-4 : Examples of smoothing filters

The mask of Gaussian filter which we used in our system and the results of noise elimination are shown below.

1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

Fig. 2-5 : Suitable 3X3 mask of Gaussian filter



(a) Before Gaussian filter (b) After Gaussian filter

(a) Before Gaussian filter (b) After Gaussian filter

Fig. 2-6 : The results of noise elimination.

In order to get foreground image, we first compute the difference image (DI) between current frame and background model. Because the human eyes are more sensitive to luminance than to chrominance, thus we only take difference value on the luminance channel. For a pixel (x,y), the difference is calculated by $DI(x,y)=\left|I_c(x,y)-I_b(x,y)\right|$, where I_c denotes as the luminance of current image, and I_b is the luminance of background image. By using the standard deviation of Gaussian background model as the threshold, we can get a possible foreground image PFI . At the same time, the DI is used to update the background model.

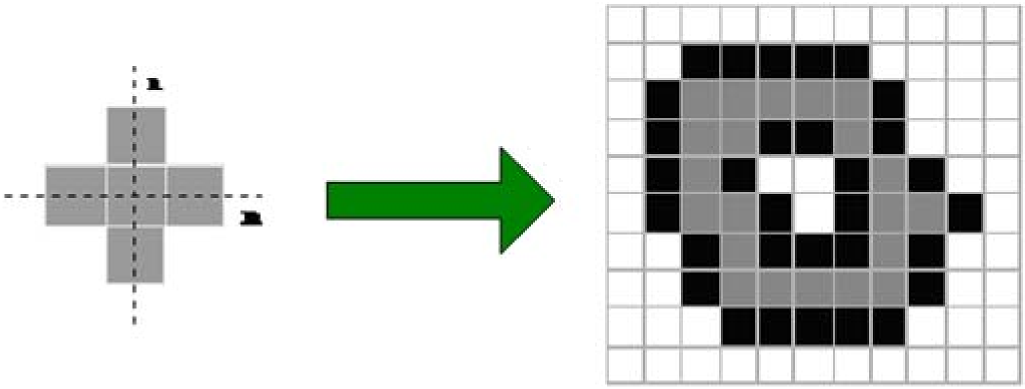
$$PFI(x,y)=\begin{cases}1 & if \quad DI(x,y)\geq 3\sigma(x,y)\\ 0 & if \quad DI(x,y)<3\sigma(x,y)\end{cases} \tag{2-1}$$

2.1.2 Preprocessing for Segmentation

After remove most of noises, some of moving object regions have been broken, so we uses twice dilation operation to fulfill the holes inside the object regions. The structuring element is 4-connected boundary points of region as shown in Fig. 2-7(a).

Each object in the foreground image must be extracted one by one. Connected components label and group each pixel based on pixel connectivity. We rid some objects which have small size, the other reason is it may be a noise. Then apply shadow elimination to remainder object.

One of moving object region may contain multiple people, especially when two or more people move together or partially occluded each other. In this case, we can not accurately recognize the object is a human or not. So, in this case, the human splitting is an important step. The moving object extraction system is illustrated as Fig. 2-8, we will explain them in following sections.



(a) Dilation mask

(b) Dilation process

Fig. 2-7 : Dilation diagram

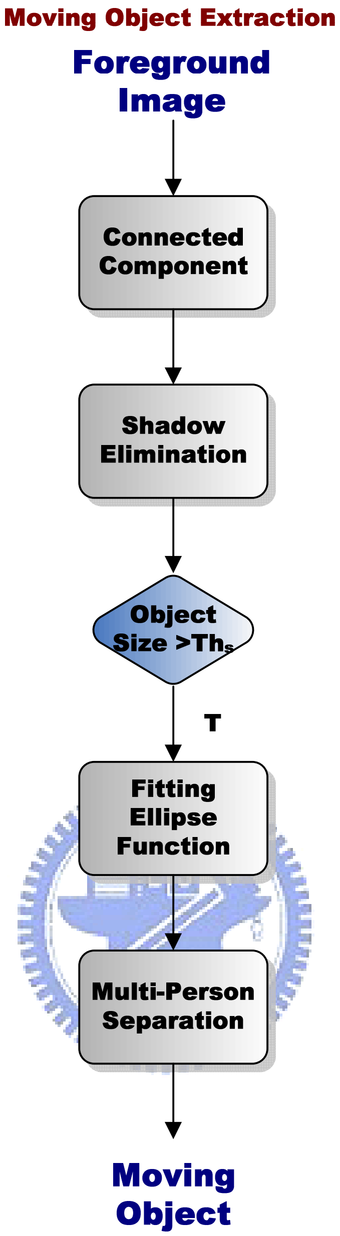


Fig. 2-8 : Moving object extraction processing diagram

2.1.3 Color Based Shadow Elimination

Color information is useful for suppressing shadows from foreground image. Given three color variables, R, G and B, the chromaticity coordinates are $r = R/(R+G+B)$, $g = G/(R+G+B)$ and $b = B/(R+G+B)$, where

$r+g+b=1$. We use < *r, g, I*> to detect shadow region, where I denotes as the luminance of the pixel. The chromaticity coordinates are used because it less sensitive to small changes in illumination [1] [5].

There are some observation about the features of shadow that can be considered [2] [17].

Observation 1: The luminance values of the shadow pixels are lower than those of the corresponding pixels in the background image.

Observation 2: The texture values of the shadow such as edge are change a little from those of the corresponding pixels in the background image.

Observation 3: The chromaticity value of the shadow pixels are change a little from those of the corresponding pixels in the background image.

Fig. 2-9 depicts the processing of shadow removing depend on above observation.

Suppose I_{PO} is the luminance of the possible-object image POI, and I_B is the background image. According to above observations, if pixel (x, y) is in a shadow region, we have following relationship:

$$I_{PF}(x,y) < I_B(x,y) \tag{2-2}$$

Between-pixel invariants:

$$\frac{I_{PF}(x,y)}{I_{PF}(x+1,y)} = \frac{I_B(x,y)}{I_B(x+1,y)} \tag{2-3}$$

Within-pixel invariants:

$$\begin{aligned} r_{PF}(x,y) &= r_B(x,y) \\ g_{PF}(x,y) &= g_B(x,y) \end{aligned} \tag{2-4}$$

Eq. (2-5) – (2-9) are used to calculate the shadow similarity value.

$$\begin{cases} d_h(x,y) = \dfrac{I(x,y)}{I(x+1,y)} \\ d_v(x,y) = \dfrac{I(x,y)}{I(x,y+1)} \end{cases} \tag{2-5}$$

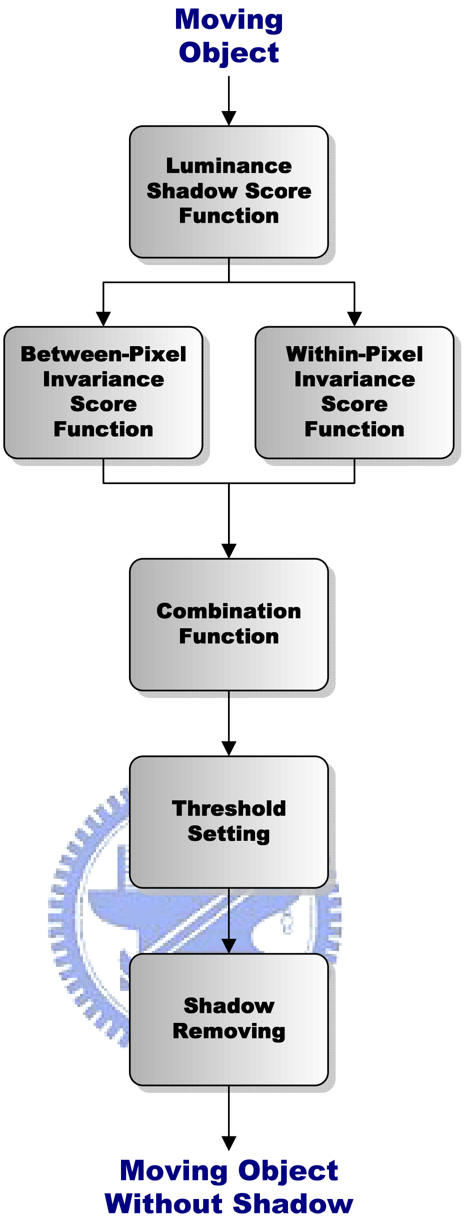

Moving Object
Luminance Shadow Score Function
Between-Pixel Invariance Score Function
Within-Pixel Invariance Score Function
Combination Function
Threshold Setting
Shadow Removing
Moving Object Without Shadow

Fig. 2-9 : Diagram of shadow elimination

$$\begin{cases} r(x,y)=\dfrac{R(x,y)}{R(x,y)+G(x,y)+B(x,y)} \\ g(x,y)=\dfrac{G(x,y)}{R(x,y)+G(x,y)+B(x,y)} \end{cases} \tag{2-6}$$

$$\Psi(x,y)=\sum_{(i,j)\in W}\left|d_{PF,h}(i,j)-d_{B,h}(i,j)\right|+\left|d_{PF,v}(i,j)-d_{B,v}(i,j)\right| \tag{2-7}$$

$$\Theta(x,y)=\left|r_{PF}(x,y)-r_{B}(x,y)\right|+\left|g_{PF}(x,y)-g_{B}(x,y)\right| \tag{2-8}$$

$$\Omega(x,y)=\alpha\cdot j(x,y)+(1-\alpha)\cdot\Theta(x,y) \tag{2-9}$$

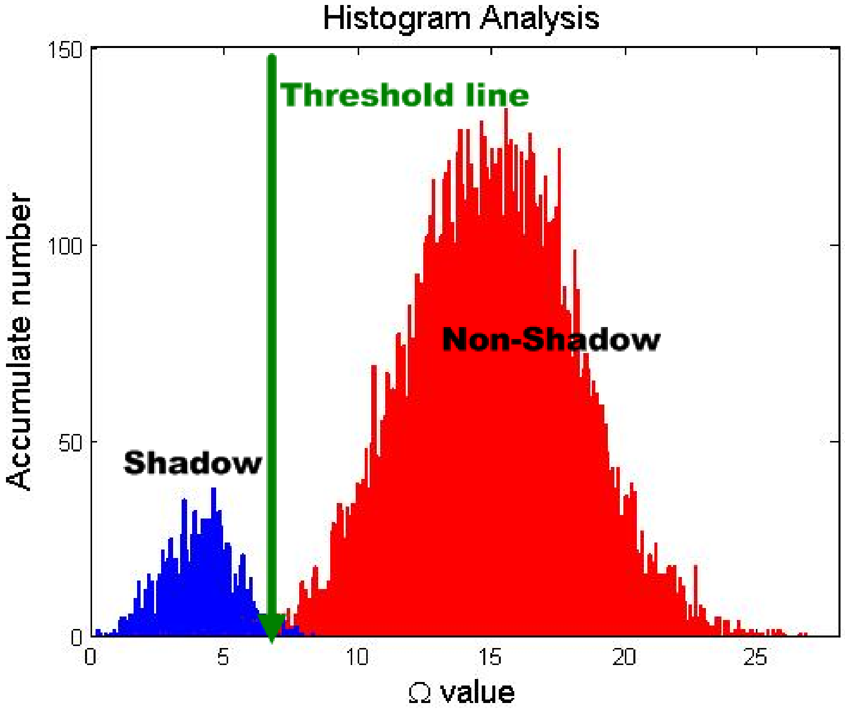
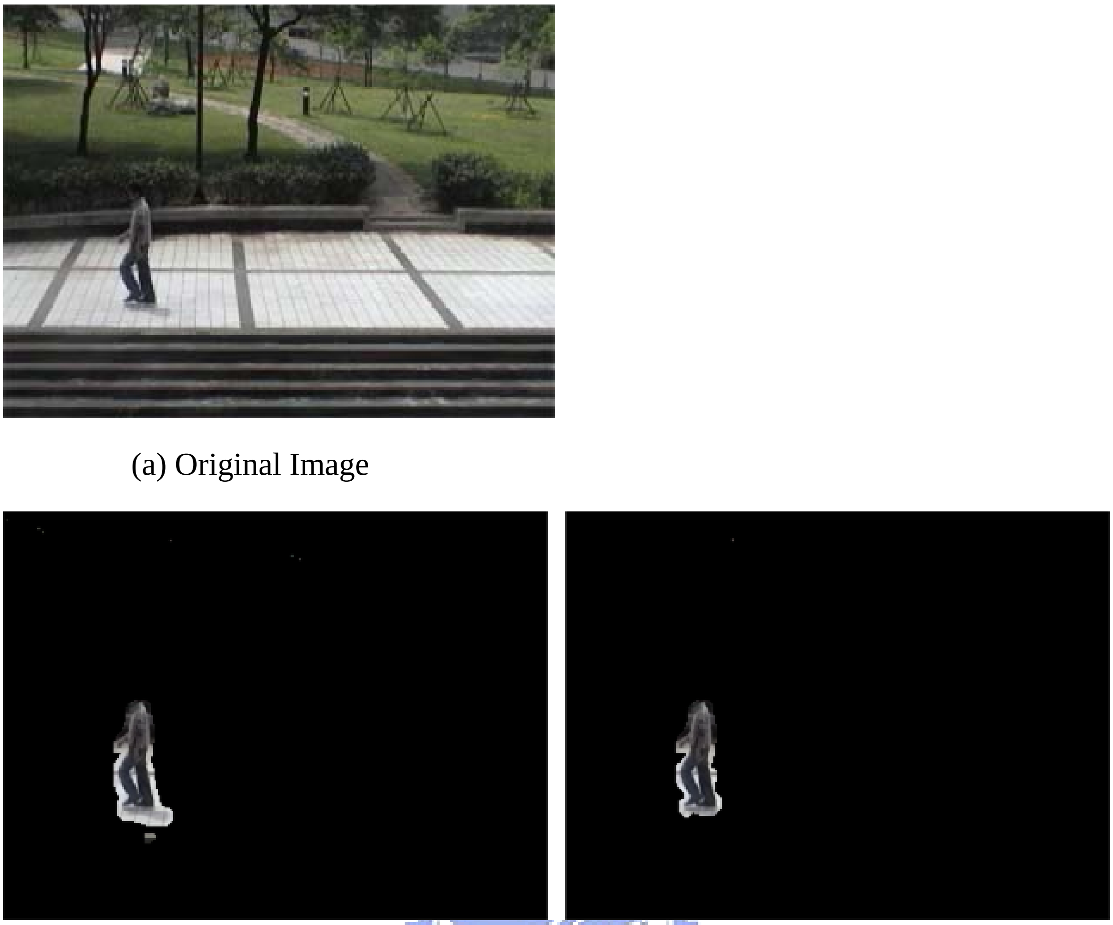


Fig. 2-10 : Distribution model of shadow and moving object

The combination of Eq. (2-9) with α weighting parameter is used to define the similarity of a moving object region to shadow. We define a dynamic threshold value to decide a pixel (x, y) is in a shadow pixel or not. In order to define this threshold value we make an assumption that the shadow region is less than the real moving object region. By using Eq. (2-9) the distribution of the pixels of a moving object is shown at Fig. 2-10. We set the threshold Th_s using Eq. (2-10) and detect a shadow image SI using Eq. (2-11), where μ_{PO} denotes the mean of POI, σ_{PO} is standard deviation of POI, and β is a weighting. The results of our shadow elimination are shown in Fig. 2-11.

$$Th_s = \mu_{PO} - \beta \cdot \sigma_{PO} \tag{2-10}$$

$$SI(x,y) = \begin{cases} 1 & if\ \ I_{PO}(x,y) > I_B(x,y)\ and\ \Omega(x,y) < \mu_{PO} - \beta \cdot \sigma_{PO} \\ 0 & otherwise \end{cases} \tag{2-11}$$



(a) Original Image

(b) Before shadow elimination

(c) After shadow elimination

Fig. 2-11 : The result of shadow Removing

2.1.4 Fitting Ellipse Function for Multiple Object Separation

Another problem of human detection is occur when a small group of people moving together or partially occluded, In this case, we can not recognize the human without separating each other perfectly. In most situations, although the group of people partially occluded, the heads are usually less occluded than body part. The head shape is so special even a person rotates his head around different phase, we can also detected the head easily. So we use the head information to overcome the occluded problem. There are many methods of head detection. In this thesis, we use ellipse model to fit the shape of the head.

Our proposed ellipse model is shown as Fig. 2-12, where the dot "●" represents the head position, the star "*" represents the background position, and the point (0, 0) is the center of ellipse model.

The pyramid down sample method is used to fit an ellipse model with different size of moving object. Fig. 2-13 illustrate the pyramid down sample process.

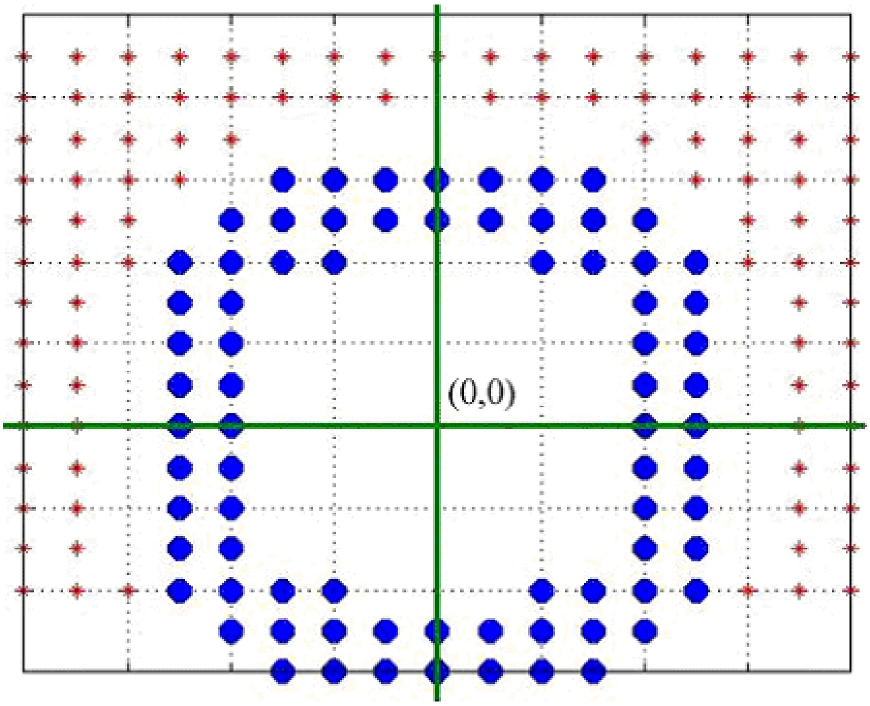


Fig. 2-12 : Fitting Ellipse model

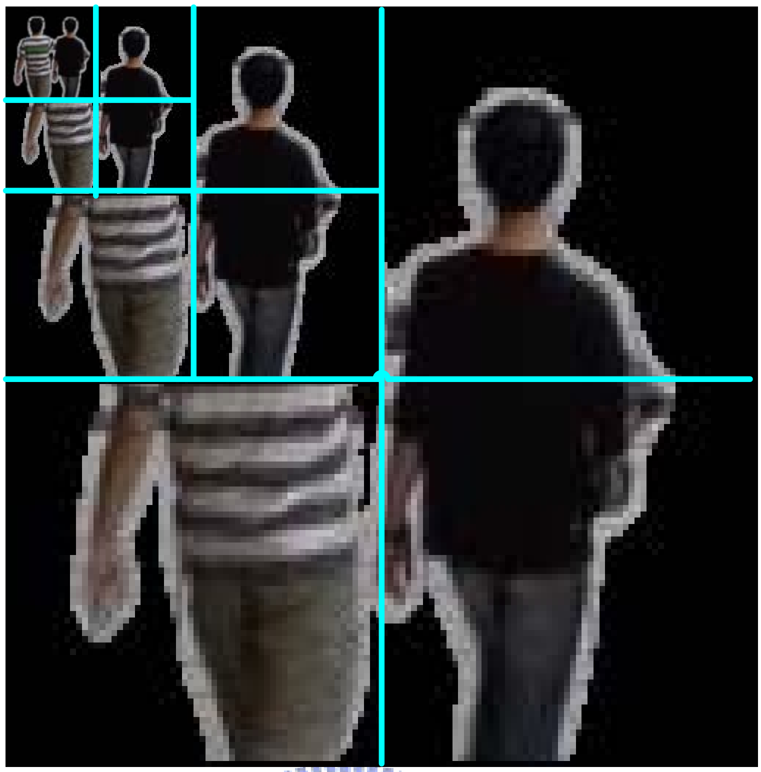


Fig. 2-13 : The pyramid down sample process

We fit the ellipse model in a searching window that containing human head. In the same time we calculate the similarity between the object and ellipse mask. By setting a threshold of similarity, we can decide which point is the possible position of the center of the head. The results of head detection is shown in Fig. 2-14, where the small square blocks are the possible positions of the head, and the other large blocks represent the object area.

At the end, there may be more than one group of points in the head position. So, what we need to do is to group these points and find the center of each group as the new center. We project each possible center point on x-axis and y-axis. The x-axis is used to distinguish each group. Then the gravity position of each group is the center of the head. Fig. 2-15 illustrate the processing, the right side, left side and bottom histograms is the number of object pixels project in y-axis and x-axis.

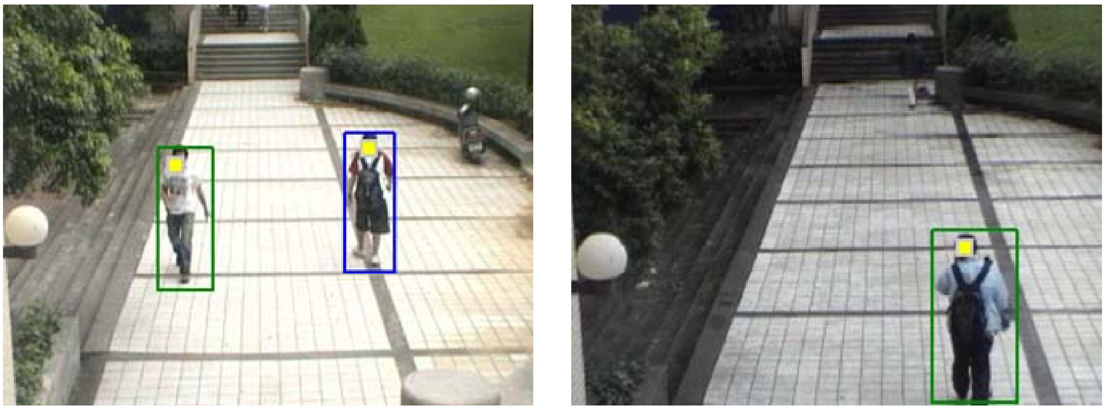


Fig. 2-14 : The results of ellipse head detection.

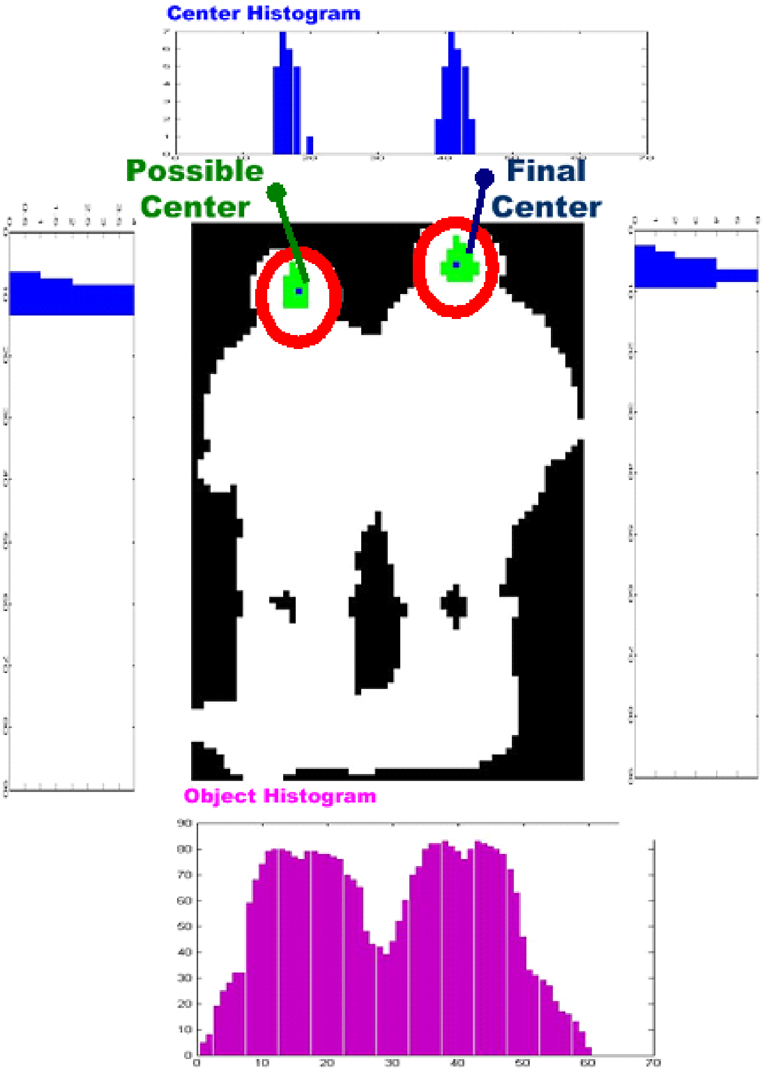
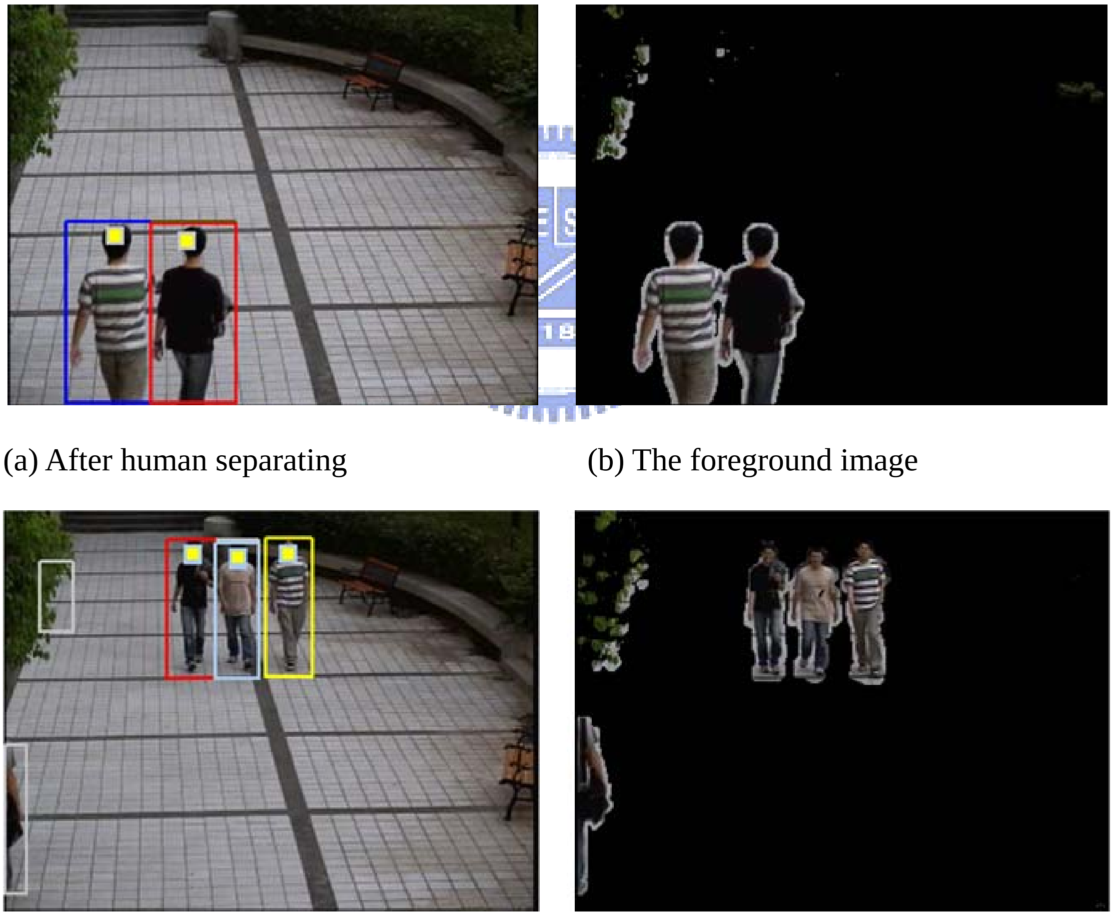


Fig. 2-15 : Separating human by fitting ellipse function

In order to eliminate some false detection such as broken foreground, raise hand, an umbrella, etc. We project the moving object on x-axis. If the center of the head is on a pick of histogram then we keep this center, otherwise we reject it.

In Fig. 2-16, the left side images (a) (c) shows the results of human separation using our method, and the left side images (b) (d) represents the corresponding foreground images. Even though the people is partially occluded as shown in the block at Fig. 2-16 (d), after fitting ellipse model each of human region are separated clearly.



(a) After human separating

(b) The foreground image

(c) After human separating

(d) The foreground image

Fig. 2-16 : The results of separating human

2.2 Moving Objects Tracking

For resolving occlusion problem and decreasing the false alarm rate, the moving object tracking is a useful and necessary processing. We use the color appearance mode as the tracking feature. Because the color distributions of items are typically quite stable under rotation, scaling and partial occluded. At the same time, the Kalman Filter calculates and predicts new the location of each object [4] - [7]. Therefore occlusion and the split of objects can also be handled in our tracking algorithm. Fig. 2-17 shows the tracking process diagram.

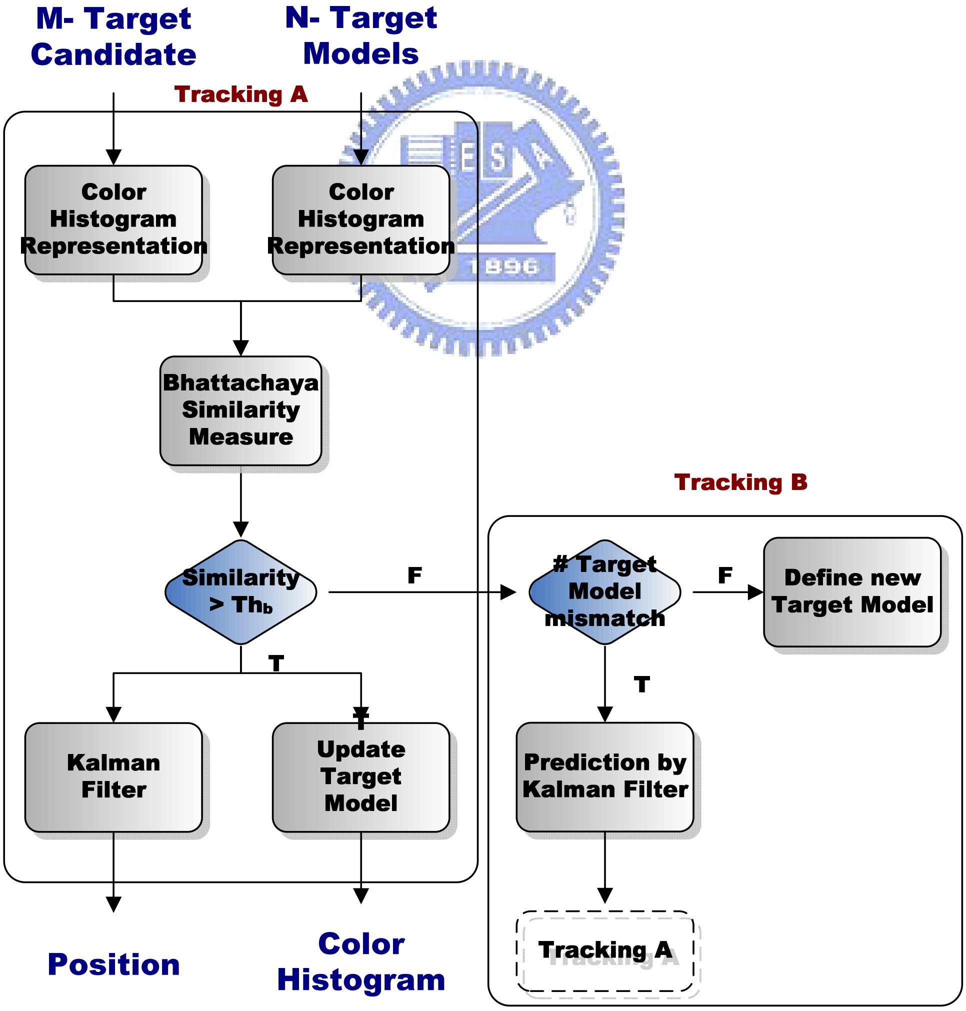


Fig. 2-17 : Moving objects tracking processing diagram

2.2.1 Tracking Based on Color Combination Model

Figure 2-18 gives some examples of the probability density function of a color histogram for an object. We can use RGB, HSV, or YUV color channel to calculate the histogram. In Fig. 2-18 the outside rectangle is the object's blob region, and the inside rectangle is the region that we used to capture it's PDF of the color histogram. The PDF is computes by Eq. (2-12), where hist(i) represents the i' th bin of the color histogram, and N is total bins. For reducing cost time and increasing the accurate of tracking object, we just collect the pixels in 3/4 part of object's block area, which centered at the center of the object.

$$p_i = \frac{hist(i)}{\sum_{i=1}^{N} hist(i)} \tag{2-12}$$

The color feature is unstable under the change of lightness, and the diaphragm of the active camera which we use will be changing by itself with the environment illumination, we have to avoid the effect of this varying factor. Consequently, we solve this problem by using the HSV color space, because it can extract the lightness information from RGB color values, so we can reduce the sensitivity of this single quantity of illumination. But using HSV color space has a problem which is when situation (S) is near to 0, hue (H) will becomes quite noisy, hence the HS histogram is used only the pixels with situation lager than a threshold 0.1. Otherwise, for gray color the hue is almost a zero, so a V (value) histogram is also needed. Thus, if S>0.1, we use HS, otherwise we use the V. We quantize these three channel color space, and its PDF will be used for our target model. The total number of histogram bines is $N_H N_S + N_V$.

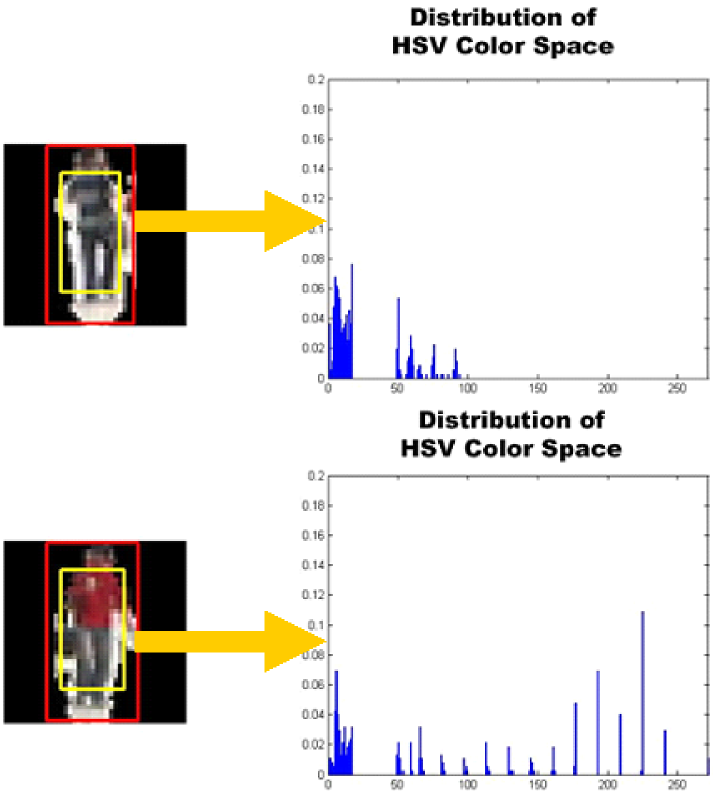


Fig. 2-18 : The PDF of color histogram

For our tracking, we calculate the Bhattacharya distance to measure the similarity of two discrete probability distributions. It is defined as Eq. (2-13), where p_i is the PDF of a candidate object, and q_i is the PDF of a target object model.

$$BC(p,q) = \sum_{i=1}^{N} \sqrt{p_i \cdot q_i} \qquad (2\text{-}13)$$

When a new object is detected in first time, we compute its PDF color histogram and take this object as target model. When the object is tracked later, we update the color model by linear combination between old color histogram model and current color histogram, as Eq. (2-14). q_i represents the i'th bin of the PDF of target object model, and p_i represents the matched object in current frame. Weighting coefficient γ is depended on Bhattacharya coefficient.

$$\begin{aligned} &\gamma = BC/4; \\ &q_i = (1-\gamma)\cdot q_i + \gamma \cdot p_i \end{aligned} \qquad (2\text{-}14)$$

2.2.2 Kalman Filter for Occlusion Handing

Kalman filter is an efficient recursive filter that estimates the state of a dynamic system from a series of incomplete and noisy measurements. We predict the position of each object by using Kalman filter, that can help use to resolve some problem about occlusion. At each frame, the position of each object is predicted by the Kalman filter, the best position is obtained as below and is used to update the filter parameters. The measurement equation relates the states and measurement at time t as follows:

Tracking Model：

$$\begin{bmatrix} y_{t+1} \\ x_{t+1} \\ y_t \\ x_t \end{bmatrix} = \begin{bmatrix} 2 & 0 & -1 & 0 \\ 0 & 2 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_t \\ x_t \\ y_{t-1} \\ x_{t-1} \end{bmatrix} \tag{2-15}$$

In Eq. (2-15), (x, y) denotes as the location of center of gravity of the object at time t. Kalman filter algorithm is shown in Table 1, where the observation of the true state is calculated by the center of the object in the foreground. At each frame, if any object candidates can not match the target model, we will use the moving object position that obtained from Kalman filter to find if there is a matched object, because the object may be occluded.

Table 1 : Kalman Filter

Kalman Filter	
$Z_t = HX_t + V_t$	*measurement state*
predict :	
$X_{t\|t-1} = AX_{t-1\|t-1} + BU_t$	*predicted state*
$P_{t\|t-1} = AP_{t-1\|t-1}A^T + Q$	*predicted estimate* cov *ariance*
update :	
$Kg_t = P_{t\|t-1}H^T / \left(HP_{t\|t-1}H^T + R\right)$	*optimal kalman gain*
$X_{t\|t} = X_{t\|t-1} + Kg_t\left(Z_t - HX_{t\|t-1}\right)$	*updated state estimate*
$P_{t\|t} = \left(1 - Kg_t\right)P_{t\|t-1}$	*updated estimate* cov *ariance*

Chapter 3

Human Detection

The human detection system structure is consisted of two sub-systems: ICA feature extraction, and SVM classifier. The process is shown in Fig. 3-1.

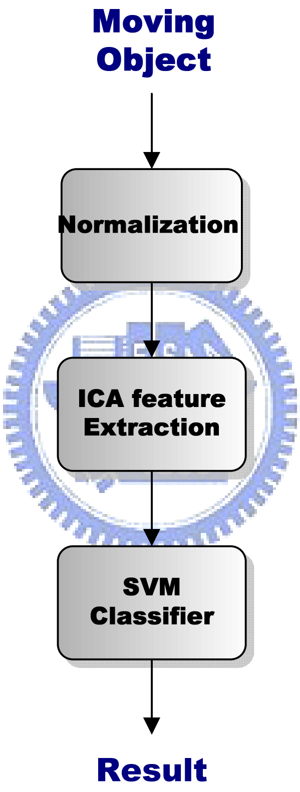


Fig. 3-1 : Human and non-human detection diagram

We normalize the training data and the testing data to 40X40 square images. A simple normalization algorithm is by comparing the width and height of moving object block. The object is centralized by shifting the object horizontally (if width > height) or vertically (if height >width). The database after normalization is shown in Fig. 3-2 and Fig. 3-3. We establish the training database from 16 different videos which consist of 1843 positive training data and 2066 negative training data. The testing database is captured from the other 18 different videos, which consist of 3178 positive testing data and 2847 negative testing data.

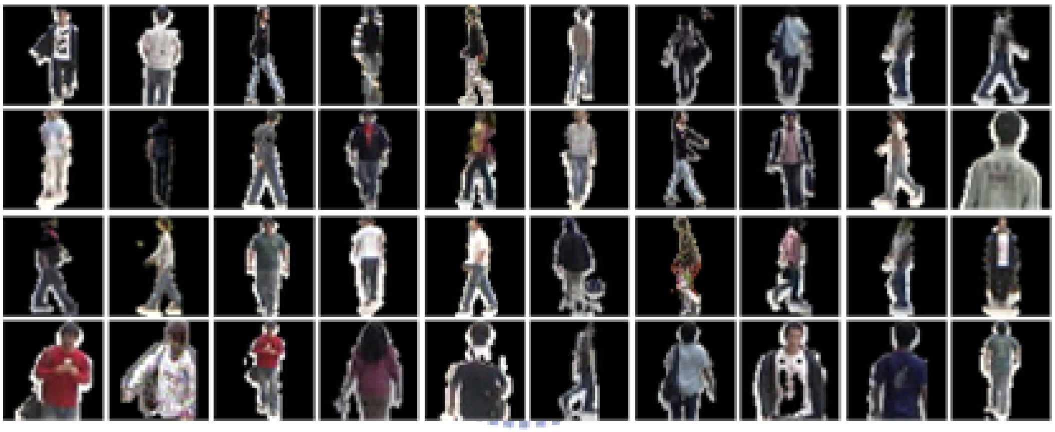


Fig. 3-2 : Positive database

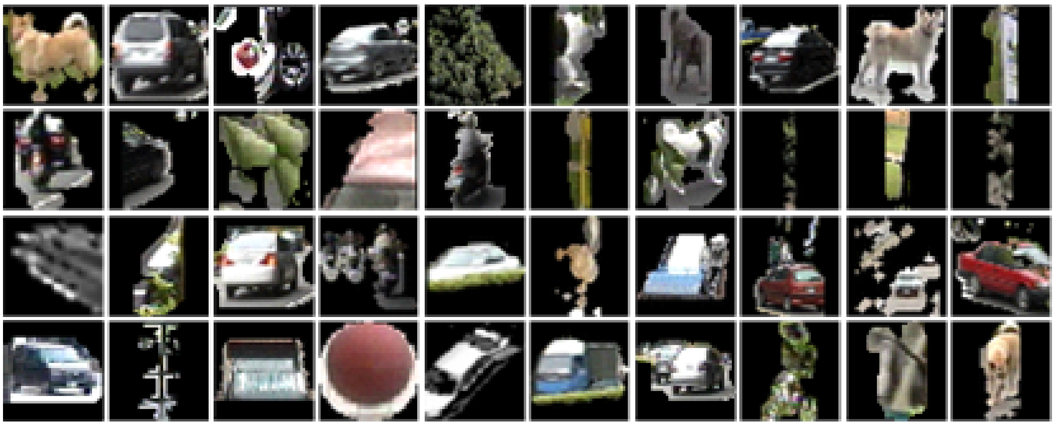


Fig. 3-3 : Negative database

3.1 Feature Extraction using Independent Component Analysis

Independent Component Analysis (ICA) is a statistical method for transforming an observed multidimensional random vector into components that are statistically independent. ICA is a generalization of principle component analysis (PCA), it is a high-order statistic approach. ICA transforms each input image to the combination of bases and its corresponding coefficients. In our system, we have a set of independent source (basis) image as our database, and choose important bases which may increase the classification ability.

But PCA is a second-order statistic approach. In our system, we get a set of independent source (basis) image as our database, and pick some better basis which may have better classified ability than others. Each input image can see as the combination of these bases, and the combination coefficients are the classified feature for our system.

In many cases, source signals are simultaneously linearly filtered and mixed. For reasons of computational and conceptual simplicity, the observation is often sought as a linear transformation of the original signals. In other words, each components of the observation is a linear combination of the original variables. The aim is to process these observations in such a way that the original source signals are extracted by the adaptive system [29] [30].

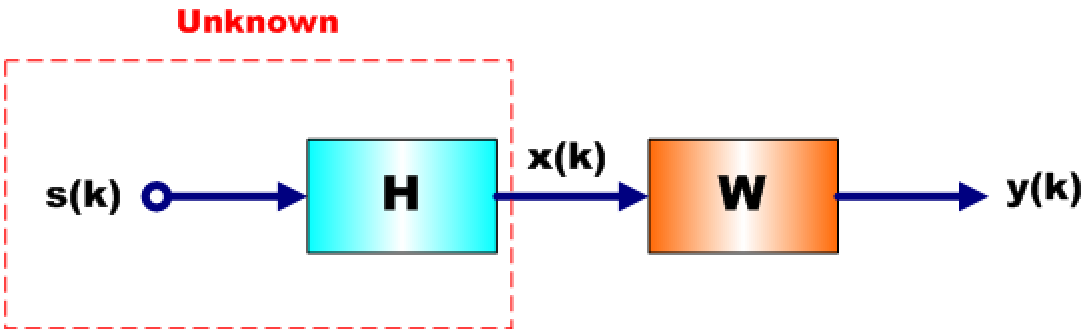


Fig. 3-4 : The block diagram of ICA model

ICA is a method to separate and estimate the original sources waveforms from the sensor array without knowing the transmission channel characteristics and the sources. The block diagram is shown in Fig. 3-4, where ***s***, ***x***, and ***y*** are defined as source signal, mixing signal and output signal, **H** and **W** matrix are defined as mixing matrix and separating matrix.

In practice, the information about the original signals and the mixing system are unknown, and the information of mixed signals is given from sensor. Assume we observe m mixed signals $\mathbf{x}_1, \mathbf{x}_{2,} \ldots, \mathbf{x}_m$ which are linear combinations of n (typically $m \geq n$) source signals $\mathbf{s}_1, \mathbf{s}_2 \ldots, \mathbf{s}_n$. The source signals are unknown mutually statistically independent and zero-mean. This can be written as following equation:

$$\mathbf{x}_j = h_{j1}\mathbf{s}_1 + h_{j2}\mathbf{s}_2 + \ldots + h_n\mathbf{s}_n \quad for\ all\ j \tag{3-1}$$

Let us denote by $\mathbf{H}$ the matrix with elements h_{ij}. It is an unknown full rank $m \times n$ mixing matrix. All vectors are understood as column vectors; thus the transpose of $\mathbf{X}$, is a row vector. And we assume $\mathbf{s}$ is an unknown vector whose elements are $\mathbf{s}_{1,}\mathbf{s}_2 \ldots, \mathbf{s}_n$. By using this vector-matrix notation, the above mixing model is written as:

$$\mathbf{X} = \mathbf{HS} \tag{3-2}$$

Without loss of generality, we can assume that both the mixture variables and the independent components have zero mean; if this is not true, then the observable variables x_j can always be centered by subtracting the sample mean, which makes the model zero-mean. It can be shown as following that we must also assume that the independent component must have non-Gaussian distributions.

To estimate the mixing matrix $\mathbf{H}$, we have to compute its inverse, which is called separating matrix $\mathbf{W}$. It is necessary to design a feed-forward or else recurrent neural network with an associated and adaptive learning algorithm that enables estimation of sources, identification of the separating matrix $\mathbf{W}$ which is a $n \times m$ full rank matrix with good tracking abilities. Then we obtain the independent component simply by Eq. (3-3), where $\mathbf{Y}$ is a vector with elements $\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_n}$..

$$\mathbf{Y} = \mathbf{WX} \tag{3-3}$$

In many applications, especially where the number of independent components is large and they have sparse (or other specific) distributions, it is more convenient to use the following equivalent form 3-4. By taking the transpose, we simply interchange the roles of the mixing matrix $\mathbf{H}$ and the ICs $\mathbf{S}$, thus the matrix $\mathbf{S}^T$ can be considered as the mixing matrix, and the vectors of the matrix $\mathbf{H}^T$ as independent components.

$$\mathbf{X}^T = \mathbf{S}^T \mathbf{H}^T \tag{3-4}$$

Equation (3-4) is obtained by estimating both the unknown matrices $\mathbf{S}$ and $\mathbf{H}$ in such a way that rows of $\mathbf{S}$ and columns of $\mathbf{H}$ be as independent as possible and both of them consist of the same or similar statistical properties.

For image application, ICA and PCA are promising approaches to Image understanding. The idea of ICA or related decomposition approaches is to decompose the image to basic independent components and to start with a large set of independent components.

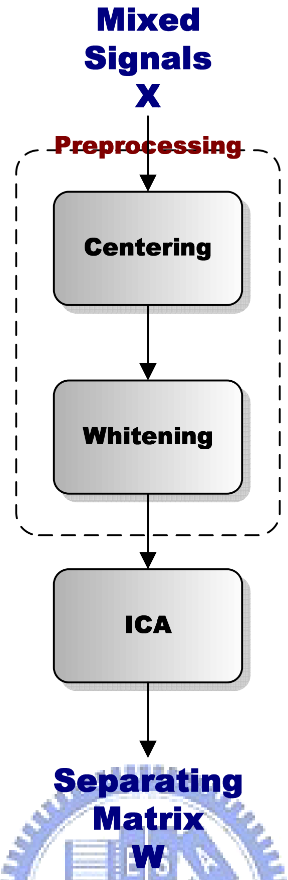


Fig. 3-5 : Off-line ICA training diagram.

The process of off-line training of ICA is illustrated in Fig. 3-5. We are going to introduce them in the following sections.

3.1.1 Preprocessing

Before applying an ICA algorithm on the data, it is worthless to do some pre-processing. In this section, we discuss some preprocessing techniques that make the estimation problem of ICA much simpler and become better. The two preprocessing techniques are centering and whitening..

Centering

Centering $\mathbf{x}'$ is the most basic and necessary preprocessing, i.e. subtract its mean vector $\mathbf{m} = E\{\mathbf{x}'\}$ thus to make a new mixtures $\mathbf{x}$ become a zero-mean variables, as shown in Fig. 3-6 (a)-(b).

$$\mathbf{x} = \mathbf{x}' - E\{\mathbf{x}'\} \tag{3-5}$$

This preprocessing does not mean it cannot be estimated. After estimating the mixing matrix $\mathbf{H}$ with centered data, we can reconstruct the original signals by adding the mean vector back to the centered estimate of independent components if necessary.

Hence, if centering is processed at beginning, both mixtures and independent components can be assumed as zero-mean signals. This processing simplifies the theory and algorithms quite a lot.

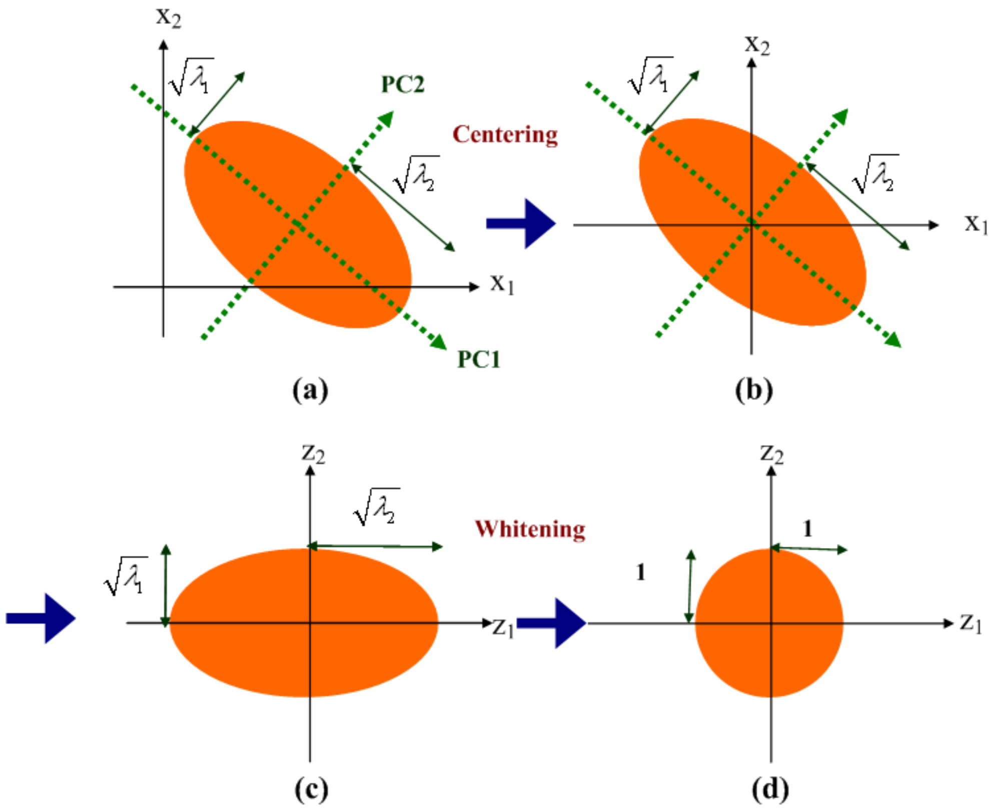


Fig. 3-6 : Pre-processing of ICA

Whitening

Whitening, sometimes called sphering is a process that let the observed vector $\mathbf{x}$'s components become uncorrelated and their variances equal unity. In other words, the covariance matrix of $\mathbf{y}$ equals the identity matrix. It is shown in Fig. 3-6 (c)-(d).

$$E\left\{\mathbf{y}\mathbf{y}^T\right\} = \mathbf{I} \tag{3-6}$$

Consequently, whitening means that we linearly transform the observed data vector $\mathbf{x}$ by linearly multiplying it with some matrixes $\mathbf{V}$, so that we obtain a new vector $\mathbf{z}$ that is white.

$$\mathbf{z} = \mathbf{V}\mathbf{x} \tag{3-7}$$

There are many linear transform methods to implement whitening; one popular method is to use the eigenvalue decomposition (EVD) of the covariance matrix, and principal component analysis (PCA) can be used here. Let's take a look at following equation

$$E\left\{\mathbf{x}\mathbf{x}^T\right\} = \mathbf{E}\mathbf{D}\mathbf{E}^T. \tag{3-8}$$

where $\mathbf{E}$ is the orthogonal matrix of eigenvectors of covariance matrix of $\mathbf{x}$, and $\mathbf{D}$ is the diagonal matrix of its eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$, followed by $\mathbf{V} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T$ is a whitening matrix. Whitening now can be done by

$$\mathbf{z} = \mathbf{V}\mathbf{x} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}\mathbf{x} \tag{3-9}$$

while $\mathbf{D}^{1/2}$ is compute by using Eq. (3-10), and the principal component is shown in Fig. 3-6(a).

$$D^{1/2} = \begin{bmatrix} \lambda_1^{-1/2} & 0 & \cdots & 0 \\ 0 & \lambda_2^{-1/2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n^{-1/2} \end{bmatrix} \tag{3-10}$$

Whiten technically transforms the mixing matrix into a new one $\tilde{\mathbf{H}}$, according to Eq. (3-2) and Eq. (3-9), we have

$$\mathbf{z} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}\mathbf{H}\mathbf{s} = \tilde{\mathbf{H}}\mathbf{s} \tag{3-11}$$

It is easy to prove that the new matrix $\tilde{\mathbf{x}}$ in Eq. (3-11) is a white matrix. This can be seen from

$$E\left\{\mathbf{z}\mathbf{z}^T\right\} = \tilde{\mathbf{H}}E\left\{\mathbf{s}\mathbf{s}^T\right\}\tilde{\mathbf{H}}^T = \tilde{\mathbf{H}}\tilde{\mathbf{H}}^T = \mathbf{I} \tag{3-12}$$

In large dimensions, an orthogonal matrix contains only about half parameters of an arbitrary matrix. Thus the whitening process can reduces the number of parameters to be estimated. It may also be quite useful to reduce the dimension of the data at the same time as we do whitening in the PCA process. PCA is a standard technique commonly used for data reduction in statistical pattern recognition and signal processing. Typically, only the N eigenvectors associated with the largest eigenvalues are used to define the subspace, where N is the desired subspace dimensionality. We can discard some eigenvalues of covariance matrix of $\mathbf{x}$ which are too small. This often has the effect of reducing noise.

3.1.2 Maximization Non-gaussianity of ICA

In this section, we show how to use kurtosis, a classic measure of non-Gaussianity, for ICA estimation.

The starting point for ICA is the very simple assumption that the components $\mathbf{s}_i$ are statistically independent. It can be proven as below that it is necessary to assume those independent components must have non-Gaussian distribution. The probability density function of each source signal must be non-Gaussian distribution, at most only one can be Gaussian distribution, if exceed one, the signals can not separate from each other.

For simplify computation, we assume that $\mathbf{y}$ has zero-mean and its variance equal to one. Moreover, there are two important measure ways: the first one is non-Gaussianity with kurtosis and the second is non-Gaussianity with negentropy which are both represented in the following sections.

Non-gaussianity with kurtosis

The classical measure of non-Gaussianity is kurtosis which is the fourth-order cumulant of a random variable, defined by:

$$kurt(y) = E\left\{y^4\right\} - 3\left(E\left\{y^2\right\}\right)^2 \tag{3-13}$$

Because of one unit of y, Eq. (3-13) can be simplified to $kurt(y) = E\left\{y^4\right\} - 3$. Then, We can see that kurtosis of y becomes a parameter which is directly related to the fourth moment of y, $E\left\{y^4\right\}$.

If y is a random variable which distribution is Gaussian, we can get $E\left\{y^4\right\} = 3\left(E\left\{y^2\right\}\right)^2$. So for any Gaussian variable, its kurtosis always equals to zero, and on the other hand, kurtosis is nonzero for most non-gaussian random variables. We call the random variables with a positive kurtosis as the super-Gaussian, and those with a negative kurtosis as the sub-Gaussian.

Because kurtosis is zero for a Gaussian variable, so the value of kurtosis is further form zero, the variable is more non-Gaussian. Thus the non-Gaussianity is measured by the absolute value of kurtosis.

However, there are some drawbacks in kurtosis; the main problem is that kurtosis is very sensitive to outliers. The value of kurtosis could be effect by only a few observations in the tails of the distribution which means kurtosis is not a robust measure of non-Gaussianity. The properties of negentropy are rather opposite to those of kurtosis, we will introduce negentropy.

Non-gaussianity with Negentropy

Negentropy is another way to measure a non-gaussianity, it is based on the information-theoretic quantity of differential entropy.

The entropy of a random variable is related to the information given by observation of the variable. If the variable is more random, that means unpredictable and unstructured, the entropy is larger. The entropy of a random vector y with probability density $p(y)$ is defined as following:

$$H(y) = -\int p(y)\log p(y)dy \qquad (3\text{-}14)$$

A Gaussian variable, which s said to have a normal distribution, is the most random or the least structured of all distributions. Thus a Gaussian variable has the largest entropy among all random variables of equal variance. Discussed above, we can use entropy as a measure of non-Gaussianity. In order to obtain a measurement of the distance to Gaussian, a slightly modified version of the definition of differential entropy, called negentropy is used. It is defined as

$$J(y) = H(y_{gauss}) - H(y) \qquad (3\text{-}15)$$

Where y_{gauss} is a Gaussian random variable which has the same covariance matrix as variable y. Negentropy is always non-negtive, and it is zero only when y has a Gaussian distribution. However, there is a drawback in using negentropy, the computational process is too difficult to use it in practice. The simpler approximations of negentropy will be introduced next.

Non-gaussianity with Approximations of Negentropy

The classical method of approximating negentropy is using higher-order moments, for example as following equation.

$$J(y) \approx \frac{1}{12}E\{y^3\}^2 + \frac{1}{48}kurt(y)^2 \tag{3-16}$$

However, the validity of such approximations may be rather limited. Therefore new approximations were developed based on the maximum-entropy principle. The approximation is shown as following(Hyvarinen,1998):

$$J(y) \approx \sum_{i=1}^{p} k_i \left[E\{G_i(y)\} - E\{G_i(v)\}\right]^2 \tag{3-17}$$

Where k_i are some positive constants, v is a Gaussian variable of zero mean and unit variance, the variable y is assumed to have zero mean and unit variance, and the functions G_i are non-quadratic functions. In this case, we use one non-quadratic function G. Then the approximation becomes:

$$J(y) \propto \left[E\{G(y)\} - E\{G(v)\}\right]^2 \tag{3-18}$$

If y is symmetric, above equation is a generalization of the moment-based approximation in Eq. (3-16). There are some useful G can be chosen:

$$\begin{aligned} G_1(u) &= \frac{1}{a_1}\log cosha_1 u \\ G_2(u) &= -\exp\left(-u^2/2\right) \\ G_3(u) &= \frac{1}{4}u^4 \end{aligned} \tag{3-19}$$

Where $1 \le a_1 \le 2, a_2 \approx 1$ are some suitable constant. They are used for FastICA algorithm which is shown as section 3.1.3.

3.1.3 The FastICA Algorithm

FastICA uses egentropy as an object function to measure the non-gaussianity of random variables. Here, non-gaussianity is measured by Eq. (3-18). In this section, we will first show the one-unit version of FastICA and extend it to the several-units version.

FastICA for one unit

The FastICA is based on a fixed-point iteration scheme for finding a maximum of the non-gaussianity of $\mathbf{w}^T\mathbf{x}$. The contrast functions G are used in Eq. (3-19), ant the derivatives of G are:

$$\begin{aligned} g_1(u) &= \tanh(a_1 u) \\ g_2(u) &= u\exp\left(-a_2u^2/2\right) \\ g_3(u) &= u^3 \end{aligned} \tag{3-20}$$

where $1 \le a_1 \le 2, a_2 \approx 1$ are some suitable constant.

The basic form of the FastICA algorithm is shown below:

1. Centering $\mathbf{x} = \mathbf{x}' - E\{\mathbf{x}'\}$

2. Whitening $\mathbf{z} = \mathbf{V}\mathbf{x},\ let\ E\{\mathbf{zz}\} = \mathbf{I}$

3. Choose an initial guess of unit norm for weight vector $\mathbf{w}$, e.g. random

4. Let $\mathbf{w}^+ \leftarrow E\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\}$ - $E\{g'(\mathbf{w}^T\mathbf{z})\}\mathbf{w}$

5. Let $\mathbf{w} \leftarrow \mathbf{w}^+ / \left\|\mathbf{w}^+\right\|$

6. If $\mathbf{w}$ is not converged, go back to step 4.

The convergence means that the old and new values of $\mathbf{w}$ point almost in the same direction, i.e. $\left|\left\langle \mathbf{w}^{k+1}, \mathbf{w}^k \right\rangle\right| \approx 1$.

FastICA for several units

To estimate several independent components, we need to run the one-unit FastICA algorithm by using several units with weight vectors $\mathbf{w}_1, \mathbf{w}_2 \ldots, \mathbf{w}_n$. We have two methods to do this, one is a deflation scheme which is based on a Gram-Schmidt-like decorrelation; the other one is using a symmetric decorrelation.

Deflation scheme means that we estimate the independent components by using one-unit FastICA algorithm one by one. Afterwards, every iteration step subtracts the projections $\mathbf{w}_{p+1}\mathbf{w}_j\mathbf{w}_j, j = 1, \ldots p$ from $\mathbf{w}_{p+1}$. Those details are listed below:

1. Centering $\mathbf{x} = \mathbf{x}' - E\{\mathbf{x}'\}$
2. Whitening $\mathbf{z} = \mathbf{V}\mathbf{x},\ let\ E\{\mathbf{zz}\} = \mathbf{I}$
3. Choose n, the number of independent components to estimate. Set counter $p \leftarrow 1$
4. Choose an initial guess of unit norm for weight vector $\mathbf{w}_p$, e.g. random
5. Let $\mathbf{w}^+ \leftarrow E\{\mathbf{z}g(\mathbf{w}^T\mathbf{z})\} - E\{g'(\mathbf{w}^T\mathbf{z})\}\mathbf{w}$
6. Do deflation decorrelation $\mathbf{w}_{p+1} \leftarrow \mathbf{w}_{p+1} - \sum_{j=1}^{p} \mathbf{w}_{p+1}\mathbf{w}_j\mathbf{w}_j$
7. Let $\mathbf{w}_{p+1} \leftarrow \mathbf{w}_{p+1} / \|\mathbf{w}_{p+1}\|$
8. If $\mathbf{w}_p$ is not converged, go back to step 5.
9. Set $p \leftarrow p+1$, If $p \le n$ go back to step 4.

In certain applications, it would be more appropriate to use a symmetric decorrelation, in which no vectors are 'privileged' over others (Karhunen et al.., 1997). Show as bellow:

1. Let $\mathbf{W} \leftarrow \left(\mathbf{W}\mathbf{W}^{T}\right)^{-1/2}\mathbf{W}$

2. Let $\mathbf{W} \leftarrow \frac{3}{2}\mathbf{W} - \frac{1}{2}\mathbf{W}\mathbf{W}^{T}\mathbf{W}$

3. Repeat step 2, until convergence

3.1.4 Optimal Independent Components Extraction

ICA can be used to create feature vectors that uniformly distribute data samples in subspace. Thus, we can separate each different class by using these subspaces. There are few papers talk about applying ICA to face recognition [9] [10] [18] [19], or different image types [8]. In our system, the main purpose is to detect whether the object is human or non-human. Therefore, we have two classes of samples need to be separated: human and non-human. In our system, we get a set of independent source (basis) image for our database, and choose better bases from databases. These bases own better classified abilities than others. We can see the combination of these bases in each input image, and the combination coefficients are the classified feature for our system.

But there exists a problem of ICA. That is, the class discriminability of in dependent component is not sorted by the creating sort, and it is not depend on binary classified capability. To solve this serious problem, we present a modified method for the feature selection; the method will be represented at the end of this section.

First, if we have m training images which include both human and non-human.

And the images size are $n_r \times n_c$, we reshape each image into a raw data. Then, the mixing data $\mathbf{X}$ is illustrated in Fig. 3-7. Here we use the luminance channel in our system and use the gray level image instead of colorful image. It is because that colorful image may decrease the computing power and make the issue easier.

We refer to use ICA to produce spatially basis images, which can separate the two classes of images. Because ICA extracts original source signals which are statistically independent from the mixture signals, shown as Eq. (3-2), (3-4). Fig. 3-8 illustrate the ICA mixing model used in image data, where each image vector with N pixels might be projected in to a subspace with only n dimensions.

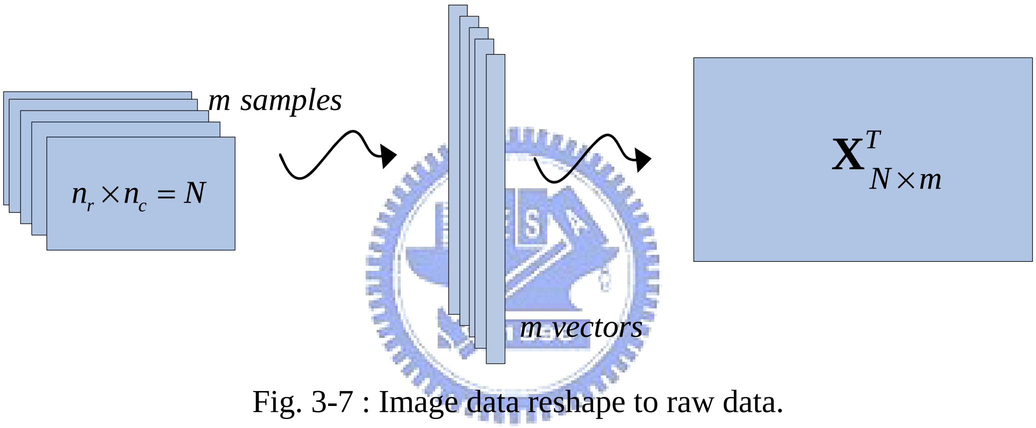
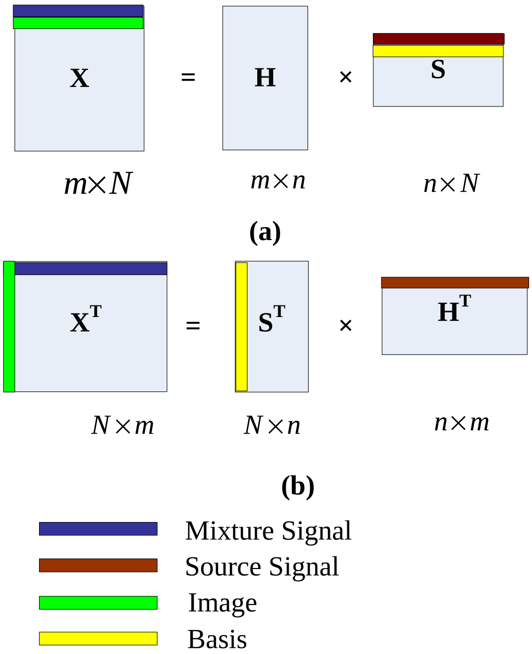


Fig. 3-7 : Image data reshape to raw data.



(a)

(b)

Fig. 3-8 : (a) ICA mixing model, (b) Equivalent form of ICA model

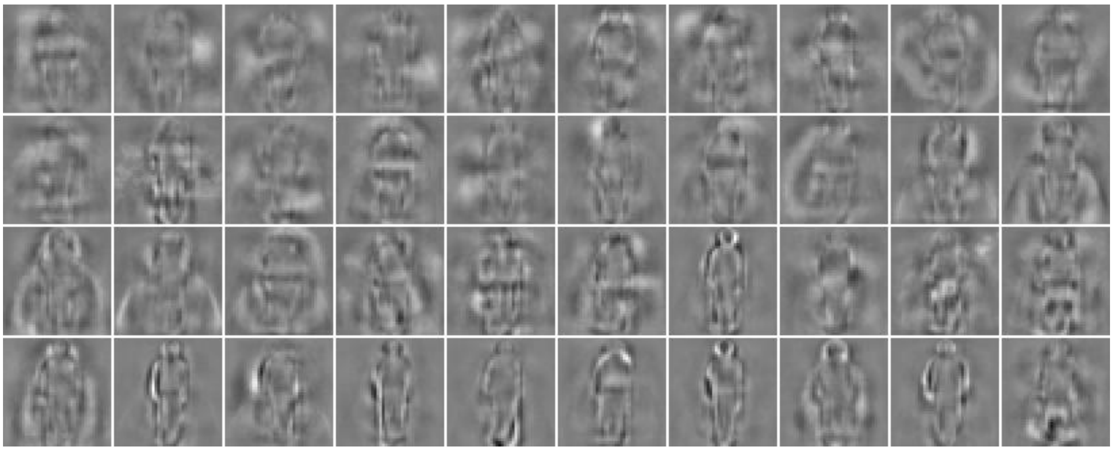


Fig. 3-9 : The basis of the image set.

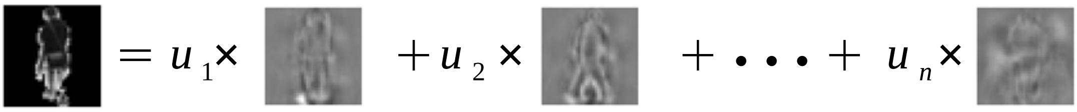


Fig. 3-10 : Image decomposition using basis from ICA.

In order to detect human and non-human, we deal each pixel in different kind of images as a mixing signal, and use the equivalent form of ICA model, the bases of mixing data $\mathbf{X}^T$ are shown in Fig. 3-9.

Any given image can be represented by a linear combination of each basis image, the linear decomposition as illustrated in Fig. 3-10. The coefficients give rare information about others, in other words, they are independent. By the way, the coefficients of the bases for reconstructing every image will be the features for our classification. If $\mathbf{W}$ is the inverse of the basis matrix $\mathbf{S}$, the coefficients matrix $\mathbf{U}$ for training matrix $\mathbf{X}^T$ will be calculated by

$$\mathbf{U} = \mathbf{W}\mathbf{X}^T \tag{3-21}$$

After done the processing of ICA, the data can be reduced to a smaller number of parameters, see as above sections. For example, image vectors with 40X40 pixels might be projected in to a subspace with only 76 dimensions.

Because we want to reduce the computing time and increase the detection rate of human detection system, feature selection is a greatly important process. The goal is that to select the best n components (basis vectors) which have a better distinguishing ability for detecting human and non-human, we should delicately choose from many components. One method is to calculate the ratio of between-class to within-class variability r for each coefficient [9], then the larger r the better distinguishing ability. Another one is to select the components according to the binary classifier capability, using Perception or Neural Network [18], Above are all depend on binary classifier capability.

If the distribution of the coefficient is like Fig. 3-11, we can distinguish human and non-human obviously by the PDF of the coefficient, where the dotted line is the threshold line. The solid line means the PDF of the positive data, and the dashed line means the PDF of the negative data. Unfortunately, the binary classifier is far from enough to select ICA features. We can take a look at Fig. 3-12, which illustrate the distribution of one coefficient on an ICA basis, the distribution can not easily distinguish two class using a threshold line. Thus, we proposed a modified ICA that is using the conditional entropy to select the optimal ICA bases. We will show the detail idea in following.

The entropy of a random variable is related to the information given by observation of the variable. If the variable is more random, that means unpredictable and unstructured, the entropy is larger. Given a discrete random variable Z which can take on possible values $\{z_1, z_2, \ldots, z_n\}$. The information entropy of Z with probability density $p(z)$ is defined as

$$H(Z) = -\sum_{i=1}^{n} p(z_i)\log p(z_i) \qquad (3\text{-}22)$$

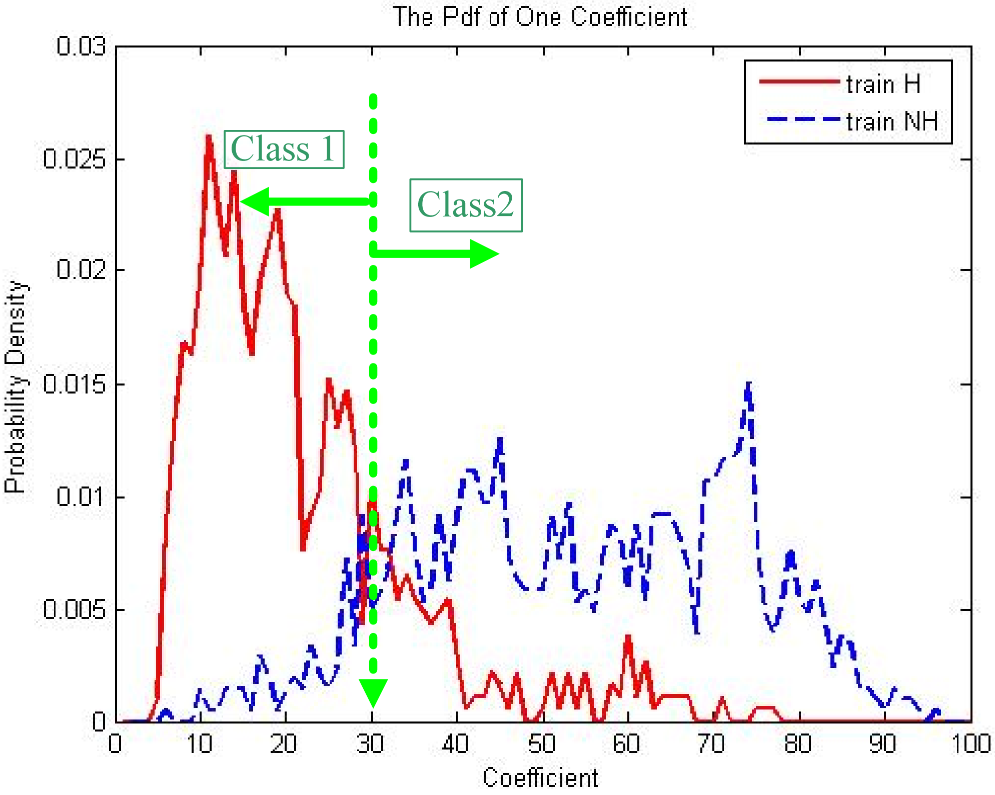


Fig. 3-11 : An ideal PDF of a coefficient

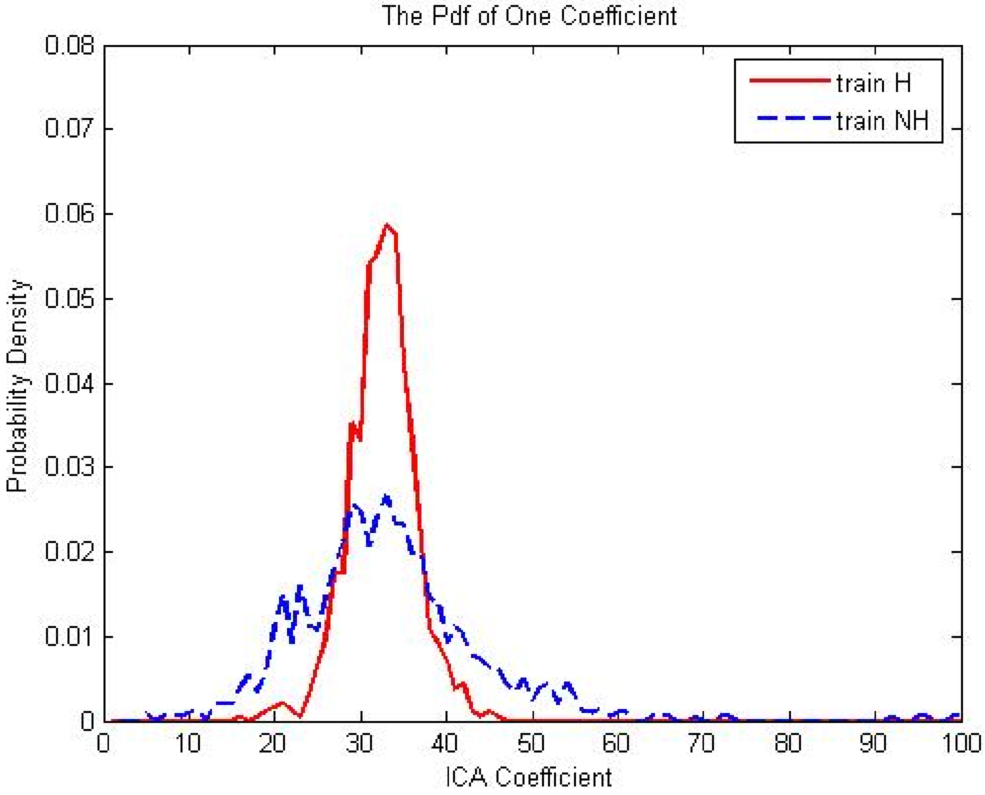
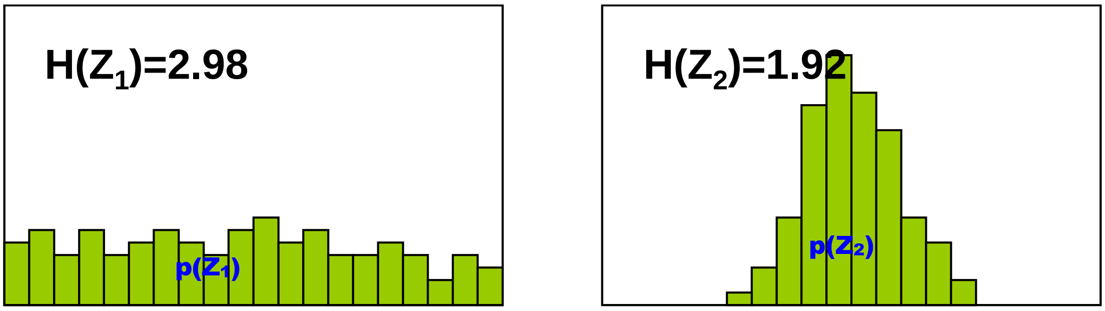


Fig. 3-12 : A real PDF of one ICA coefficient



(a) Z_1 (b) Z_2

Fig. 3-13 : The information entropy

The examples of the information entropy are shown in Fig. 3-13. Variable Z_1 is more random than Z_2, so the information entropy of Z_1 is larger than Z_2. A very sharply peaked distribution has a very low entropy, in another word, a distribution are spread out over many bins has high entropy.

An idea is that to select the best n components (basis vectors) which have a better distinguishing ability to classify. When one object input to our system, we classify it to human class or non-human class depend the coefficients.

In information theory, the conditional entropy quantifies the uncertainty of a random variable *Y* given that the value of a second random variable *Z* is known, it is defined as Eq. (3-23). For calculating, we normalize each coefficient variable to [-1, 1] and quantize it to *n* bins. Let $Y=\{-1,1\}$ be the desired class and $Z=\{z_{1,}z_2,\ldots,z_n\}$ be the ICA coefficient. We calculate the conditional entropy of each coefficient, and set it as a classifier capability.

$$H(Y|Z)=-\sum_z\sum_y p(y,z)\log p(y|z)=H(Y,Z)-H(Z) \tag{3-23}$$

where joint entropy is defined by

$$H(Z,Y)=-\sum_z\sum_y p(z,y)\log p(z,y) \tag{3-24}$$

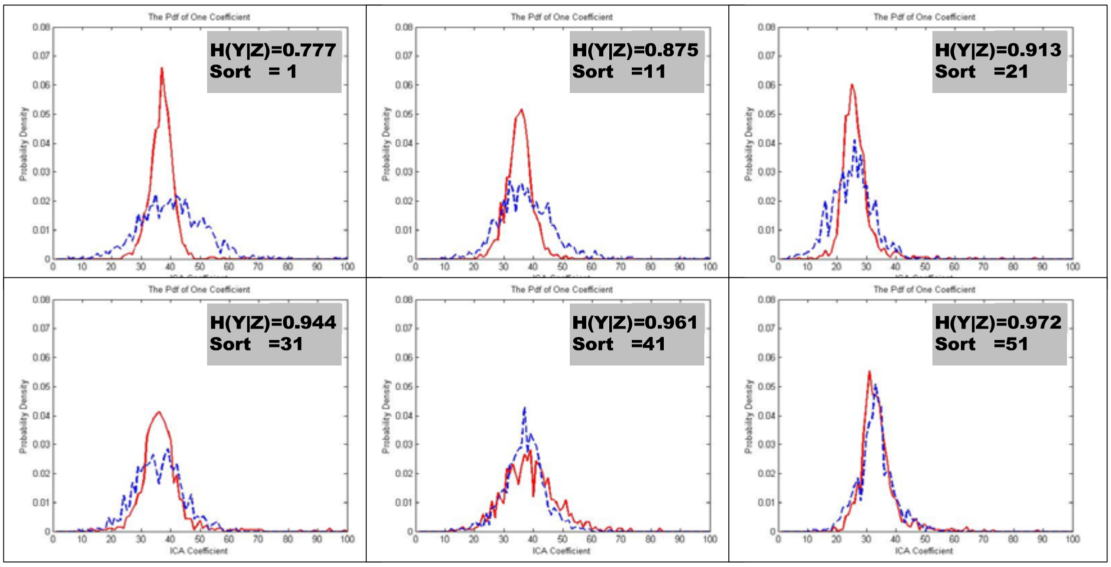
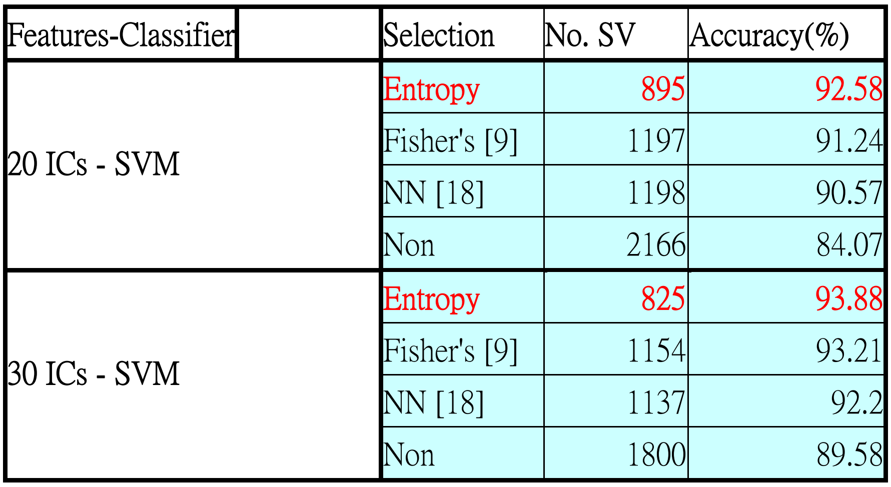


Fig. 3-14 : The Conditional Entropy of ICA coefficients

We sort these bases by the conditional entropy of its coefficient PDF, and select bases depend on the sort. The coefficients which have good classification ability are much with small conditional entropy. Fig. 3-14 illustrates some example of the PDF of ICA coefficients, shows the conditional entropies and the sort. In our experiment, the maximum dimension of ICA is 76, in another word, the maximum number of basis is 76.

Table 2 shows the classified result of selection of components from our databases, our approach is compared with a binary way, where we select two subsets of 20 and 30 independent components by using our methods and Fisher’s criterion [9], a Neural Network [18], and a result of non selection. Then the classifier compares with SVM, and we show the number of support vectors (SV) for each one. We can see that our approach has a good out come than others, not only in classified accuracy but also the number of SV. The testing data is our testing database, total number is 6025 images.

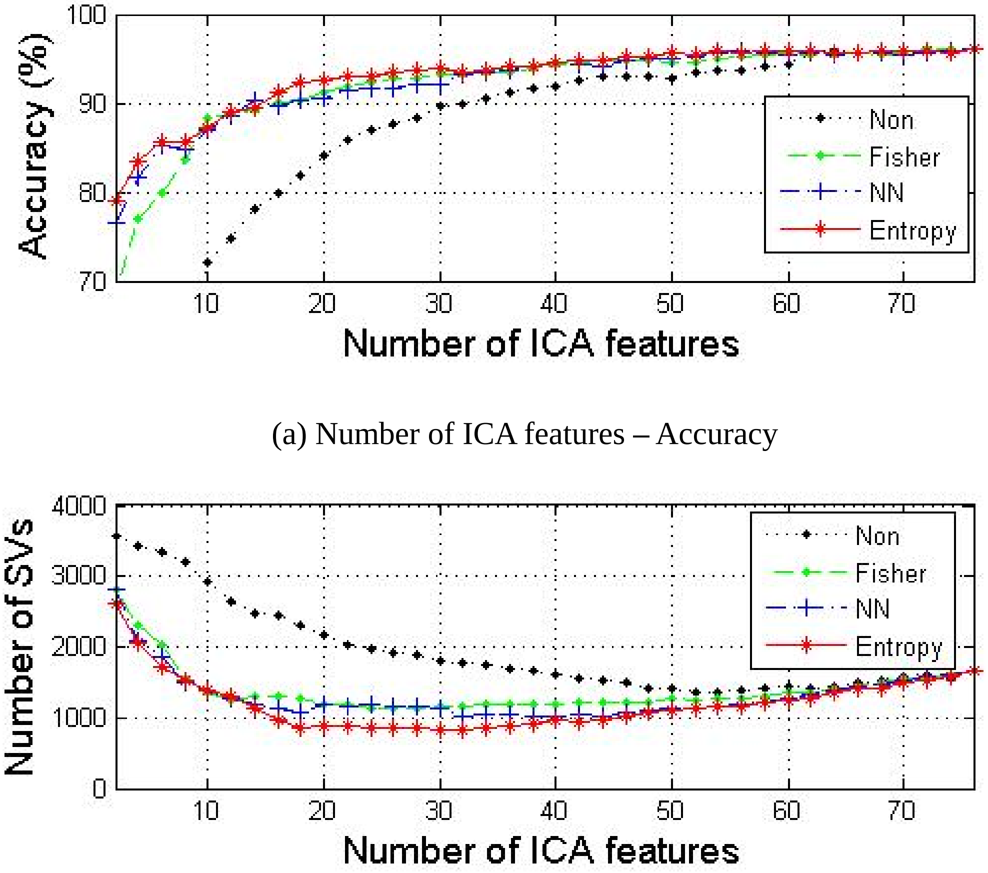
Table 2 : Results of feature selection



Features-Classifier		Selection	No. SV	Accuracy(%)
20 ICs - SVM		Entropy	895	92.58
		Fisher's [9]	1197	91.24
		NN [18]	1198	90.57
		Non	2166	84.07
30 ICs - SVM		Entropy	825	93.88
		Fisher's [9]	1154	93.21
		NN [18]	1137	92.2
		Non	1800	89.58

Because the drawback of ICA algorithm is that the class discriminability of independent component is not sorted by the creating sort, and it is not depend on binary classified capability. The main contribution of our thesis is that providing a resolution of the drawback of ICA. Fig. 3-15 illustrate the results of our proposed method and other methods, horizontal axis means the selected number of ICA features, above vertical axis means accuracy of testing data, and under vertical axis means the corresponded number of SVs, where the point-dotted line represents the result before feature selection, the point-dashed line represents the result of Fisher's criterion [9], the plus-dashdot line represents the result of NN [18], and the star-solid line represents the result of our proposed method. The total features after the process of ICA is 76, we select *n* features from them.

We can observe some thing from Fig. 3-15, first one is that whatever the feature selection method is when the number of ICA features increase from zero the accuracy will increase, and the corresponded number of SVs will decrease. However, when the number of ICs increase to the maximum number, in order to maintain the accuracy we need to use more and more SVs, that also represents the unstable of detection system. So how to select a suitable number of the features let the system have an optimal classified capability is a very important work.



(a) Number of ICA features – Accuracy

(b) Number of ICA features – Number of SVs

Fig. 3-15: Analysis of feature selection

3.2 Classification Using Support Vector Machines

Support Vector Machines (SVMs) are developed to solve the classification and regression problems. SVM has similar roots with neural networks, it demonstrates the well-known ability of being universal approximates of any multivariable function to any desired degree of accuracy, it is produced by Vapnik et. al. by using some statistical learning theory [31] - [32].

3.2.1 Introduction

Hard-Margin Support Vector Machines

SVM is a way which starts with a linear separable problem. First, we discuss hard-margin SVMs, in which training data are linearly separable in the input space. Then we extend it to the case where training data cannot be linearly separable.

For classification, the goal of SVM is to separate the two classes by a function which is induced from available example. Consider the example in Fig. 3-16, there are two classes of data and many possible linear classifiers that can separate these data, but only one of them is the best classifier which can maximize the distance between two classes - margin, this linear classifier is called optimal separating hyperplane.

Given a set of training data $\{\mathbf{x}_i, y_i\}, i=1,\ldots,m$, where $\mathbf{x}_i \in R^p$, $y_i \in \{+1,-1\}$, where the associate labels are $y_i = 1$ for class1 and -1 for class2. If this data are linearly separable, we can determine the decision function:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} - b \tag{3-25}$$

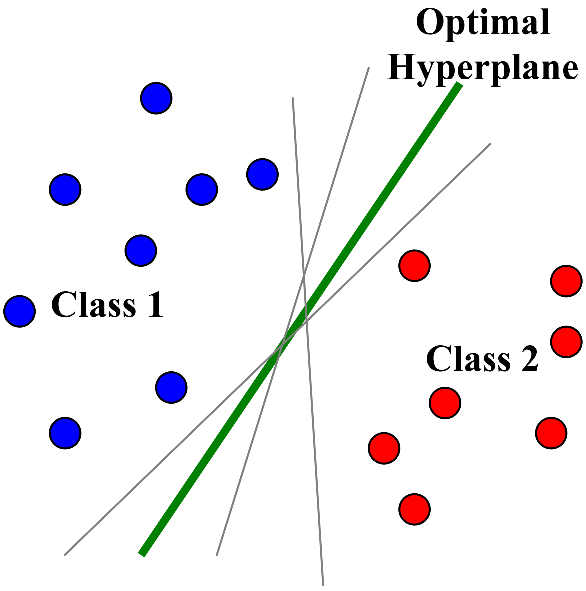


Fig. 3-16 : Optimal separating hyperplane

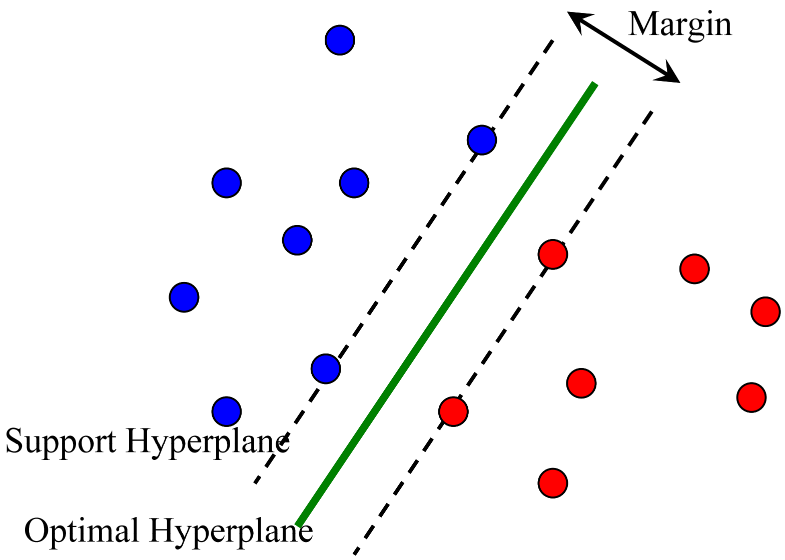


Fig. 3-17 : Illustrate the idea of SVM

Let

$$\begin{aligned}&\mathbf{w}^T\mathbf{x}_i - b > 0 \quad \mathit{for}\ y_i = +1\\&\mathbf{w}^T\mathbf{x}_i - b < 0 \quad \mathit{for}\ y_i = -1\end{aligned} \tag{3-26}$$

The vector $\mathbf{w}$ is a normal vector; it is perpendicular to the hyperplane. The parameter b determines the offset of the hyperplane from the origin along the normal vector $\mathbf{w}$, see as Fig. 3-17.

Because the training data are linearly separable, without error data satisfying $\mathbf{w}^T\mathbf{x} - b = 0$, we can select two hyperplanes that maximize the distance between two classes, the two hyperplanes include the closest data points which are named support vectors, and also called support hyperplanes. The problem can be described by the following equation, after scaling:

$$\begin{aligned}&\mathbf{w}^T\mathbf{x} - b \geq +1 \quad \mathit{for}\ y_i = +1\\&\mathbf{w}^T\mathbf{x} - b \leq -1 \quad \mathit{for}\ y_i = -1\end{aligned} \tag{3-27}$$

The distance between the two support hyperplane is $2/\|\mathbf{w}\|$, so we want to maximize the margin which means minimize $\|\mathbf{w}\|$. Thus, we have the following optimal problem:

$$\begin{aligned} &choose\ \mathbf{w}, b\ to\ \min imize \frac{1}{2}\|\mathbf{w}\|^2 \\ &subject\ to\ y_i\left(\mathbf{w}\mathbf{x}_i - b\right) - 1 \geq 0 \quad \forall i \end{aligned} \tag{3-28}$$

In order to solving the above primal problem of the SVM, we using the method of Lagrange multipliers (Minoux, 1986), and the function will be constructed:

$$L\left(\mathbf{w}, b, \alpha\right) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i \left[y_i \left(\mathbf{w}\mathbf{x}_i - b \right) - 1 \right] \tag{3-29}$$

α are the Lagrange multipliers. The Lagrangian has to be minimized with respect to $\mathbf{w}, b$ and maximized with respect to $\alpha \geq 0$.

$$L \arg range\ Multiplier\ Condition: \ \alpha_i \geq 0 \tag{3-30}$$

$$Momplementary\ Slackness: \alpha_i \left[y_i \left(\mathbf{w}^T \mathbf{x}_i - b \right) - 1 \right] = 0 \tag{3-31}$$

To minimum with respect to $\mathbf{w}$ and b of Largrangain L is given by:

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \ \Rightarrow \ \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{3-32}$$

$$\frac{\partial L}{\partial b} = 0 \ \Rightarrow \ \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \tag{3-33}$$

Equation (3-30)-(3-33) are called the KKT conditions (Karush- Kuhn- Tucker conditions). Some training data points which satisfied KKT conditions are the support vectors. Hence the solution to the problem is given by,

$$\alpha^* = \arg\min_{\alpha} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{k=1}^{m} \alpha_k \tag{3-34}$$

$$\mathbf{w}^* = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{3-35}$$

$$b^* = \frac{1}{|S|} \sum_{i \in S} y_i - \mathbf{w}^{*T} \mathbf{x}_i \tag{3-36}$$

S is the set of support vectors. Hence, the classifier is simply,

$$f\left(\mathbf{x}\right) = \operatorname{sgn}\left(\mathbf{w}^* \mathbf{x} + b^*\right) \tag{3-37}$$

Soft-Margin Support Vector Machines

However, in most situations, the training data are not optimal linearly separable, see as Fig. 3-18. There are some training data points in the wrong side. In order to correctly separate the data, a method of introducing an additional cost function associated with misclassification is appropriate, see as following equation, where $\xi_i \geq 0$.

$$\begin{aligned} &\mathbf{w}^T\mathbf{x} - b \geq +1 - \xi_i \quad for\ y_i = +1 \\ &\mathbf{w}^T\mathbf{x} - b \leq -1 + \xi_j \quad for\ y_i = -1 \end{aligned} \tag{3-38}$$

Of course, the residual value ξ_i is better when they are smaller, thus we need to minimize the cost.

$$Cost = C\left(\sum_i \xi_i\right)^k \tag{3-39}$$

The new problem is:

$$\begin{aligned} &choose\ \mathbf{w}, b\ to\ \min imize \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \\ &subject\ to\ y_i(\mathbf{wx}_i - b) - 1 \geq 0 \quad \forall i \\ &\qquad\qquad\quad \xi_i \geq 0 \quad \forall i \end{aligned} \tag{3-40}$$

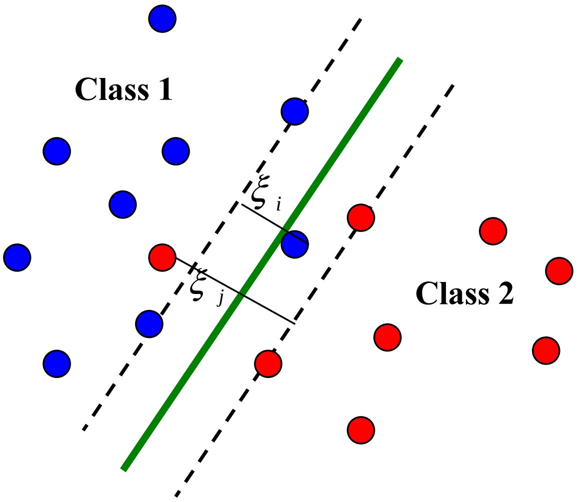


Fig. 3-18 : In separable case in a two-dimensional space

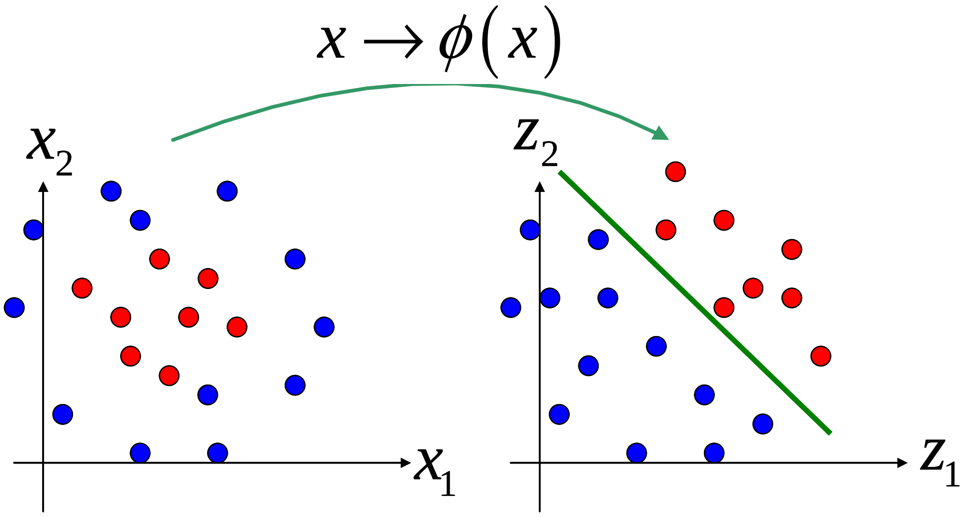


Fig. 3-19 : Feature space transforming.

Mapping to a High-Dimensional Space

If the training data are not linearly separable, we can enhance linear separability in a feature space by mapping the input space into the high-dimensional feature space. Here we show an example in Fig. 3-19.

The resulting algorithm is formally similar, except that every dot product is replaced by a non-linear kernel function k. This allows the algorithm to fit the maximum-margin hyperplane in the transformed feature space.

In the following are some of the kernels that are used in support vector machine.

Linear kernels: $k(\mathbf{x},\mathbf{x}') = \mathbf{x}^T\mathbf{x}'$

Polynomial kernels: $k(\mathbf{x},\mathbf{x}') = (\mathbf{x}^T\mathbf{x}')^d$

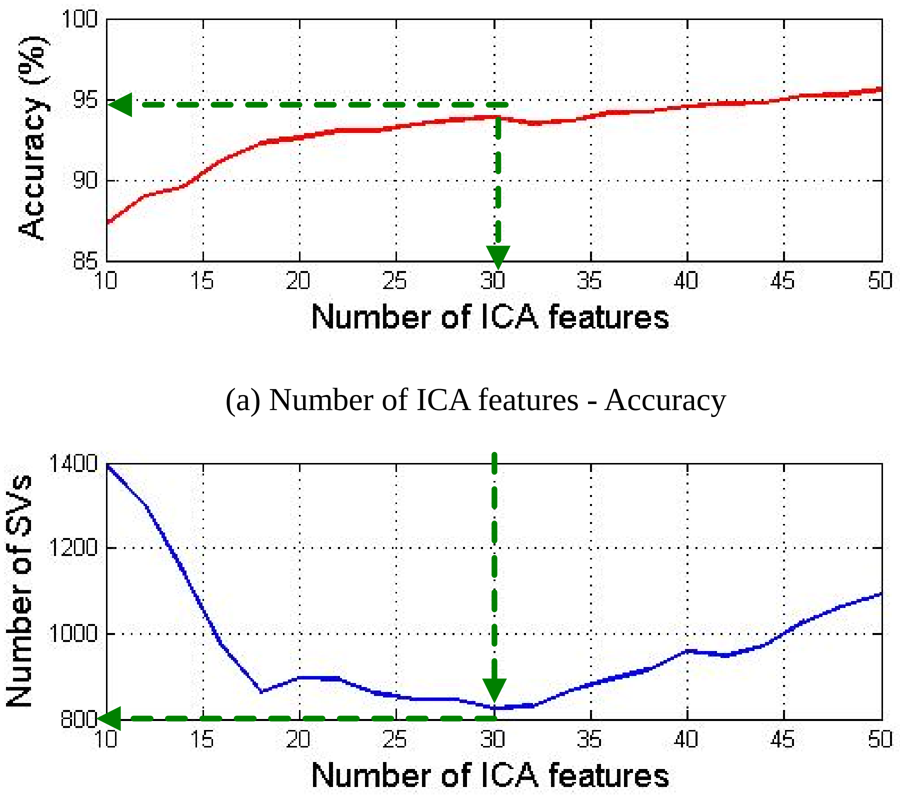
Radial basis function kernels: $k(\mathbf{x},\mathbf{x}') = \exp\left(-\gamma\|\mathbf{x}-\mathbf{x}'\|^2\right)$, *for* $\gamma > 0$

Sigmoid: $k(\mathbf{x},\mathbf{x}') = \tanh\left(k\mathbf{x}^T\mathbf{x}'+c\right)$

3.2.2 Human Classification Based on SVM

In our system, we use the LIBSVM tools [28] to train the classifier for human detection. The training data is the result coefficients form ICA. A RBF kernel function is chosen for our system. When we are using SVM training, there are two important parameters need to be set, the first one is the gamma (-g) value of kernel function, the second one is the cost (-c) value for misclassified data. The experience parameters for our image data are –g = 0.5, and -c =2.0.

Depend on the classified result of SVM, we select 30 components for our system. Figure 3-20 illustrates the selection of number of independent components (IC), horizontal axis means the number of IC which selected by conditional entropy, above vertical axis is the corresponding accurate of our testing data, and below vertical axis means the corresponding number of support vectors (SV). We can see that, when we select 30 ICs, the number of SVs is almost close to the minimum, and the accurate is better. Even thought 40 ICs have better accuracy, at the same time it need exceed approximately 200 SVs than using 30 features that may cost more calculate power for classification.



(a) Number of ICA features - Accuracy

(b) Number of ICA features – Number of SVs

Fig. 3-20 : Number of feature selection

Chapter 4

Experimental Results

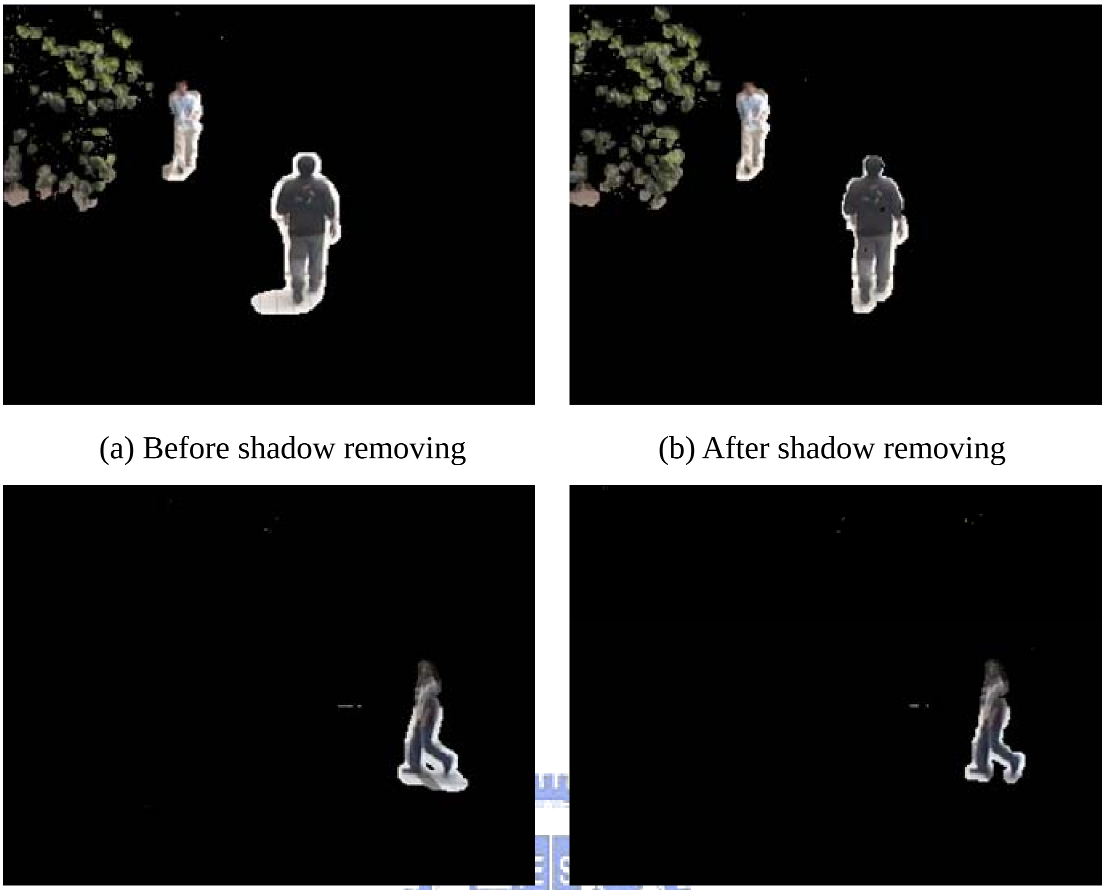
In this chapter, we will show the experimental results of our human detection system. We implemented our system with PC Intel P4 2.8H and 1G RAM. We use Borland C++ Builder 6.0 on Window XP OS. The inputs are video files (AVI uncompressed format), all the inputs are in the format of 320X240 and 30 FPS.

In section 4.1, we present the result of moving object segmentation, which included background construction, shadow elimination, and human separation. In section 4.2, the result of human detection and objects tracking system will be presented. Finally, comparing our system with other techniques, here we will make some discussion in section 4.3.

4.1 Moving Object Segmentation

We use Gaussian Mixture Model (GMM) for modeling a background image. We only update 1.6% of the background image during each frame instead of update full image, and the full image will be updated approximately 2 seconds.

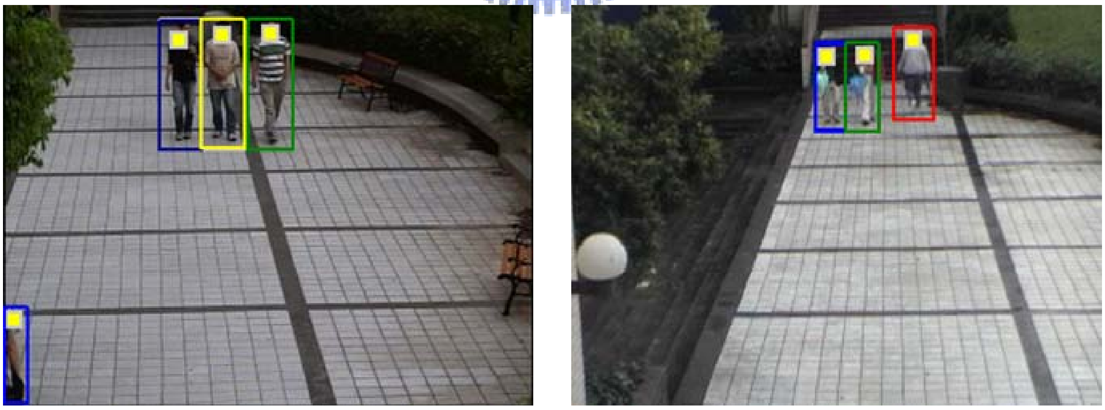
Shadow is another problem of motion segmentation. Fig. 4-1(a) (c) shows the segmentation without shadow elimination, and Fig. 4-1(b) (d) shows the segmentation after our shadow elimination.



(a) Before shadow removing (b) After shadow removing

(c) Before shadow removing (d) After shadow removing

Fig. 4-1 : The results of shadow elimination



(a) Separating three people (b) Separating two people

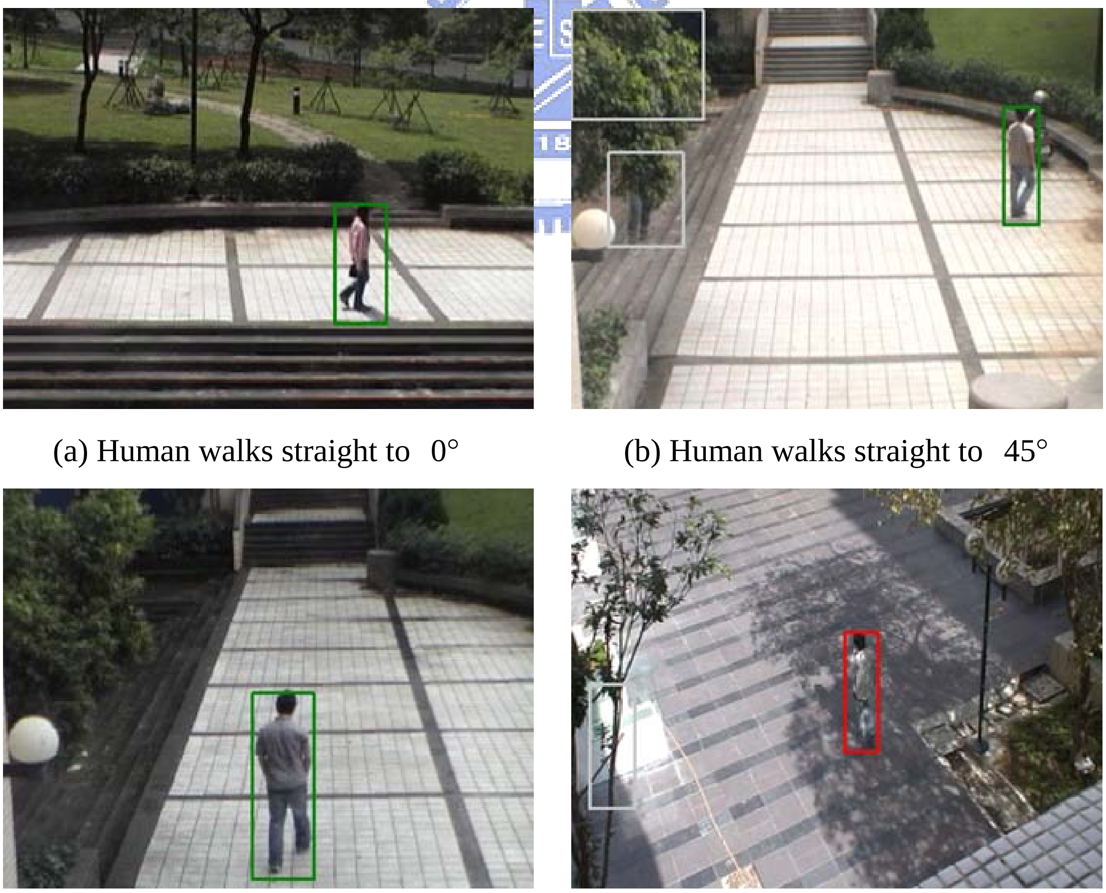
Fig. 4-2 : The results of human separation

The moving object might be partially occlusion to each other, in order to detect human accurately, we split this moving object by using ellipse head model. The results of human separating are shown in Fig., 4-2. We can separate each one from other people, where the small squares represent the positions of the head.

4.2 Human Detection and Tracking

First, we define the gray blocks represent non-human objects, and other color blocks represent human objects.

Figure 4-3 shows the detection results of multi-pose human. There are some pose of human depend on the walking direction. We can see that our system can detect human successfully even the human has vary pose.

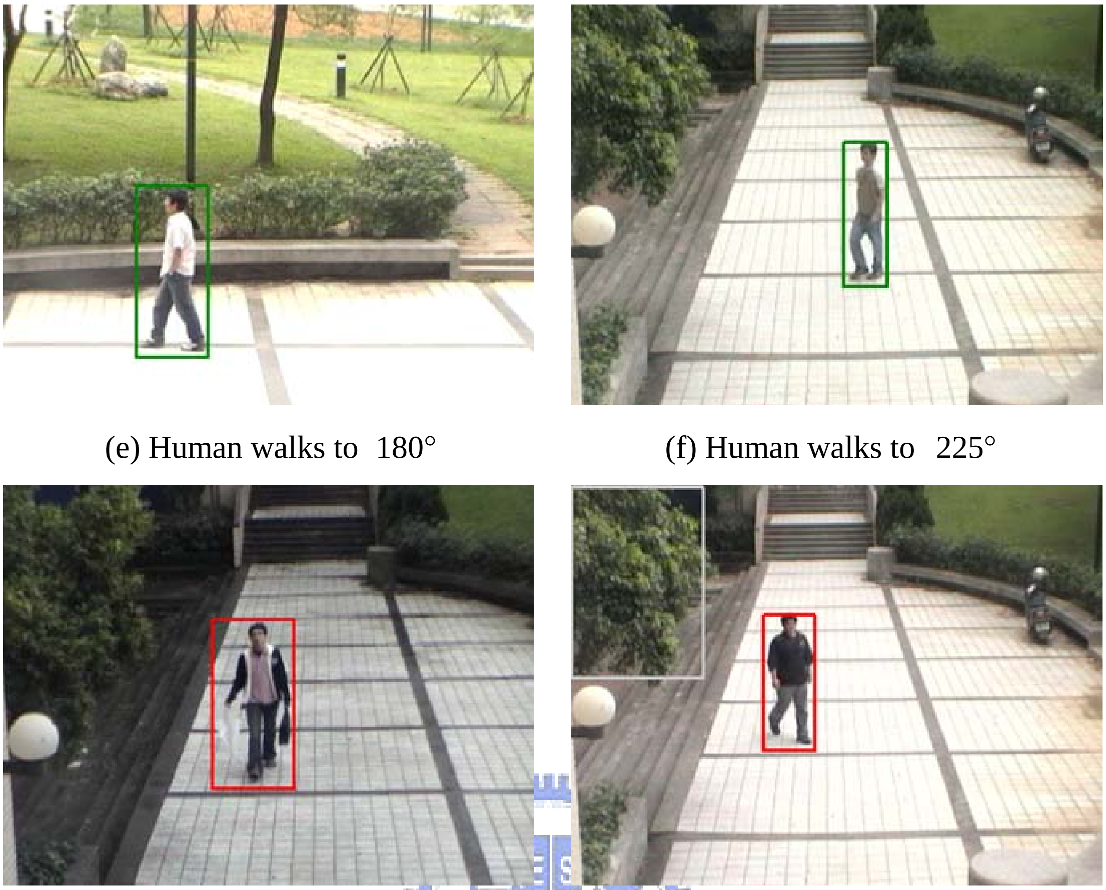


(a) Human walks straight to 0°

(b) Human walks straight to 45°

(c) Human walks to 90°

(d) Human walks to 135°

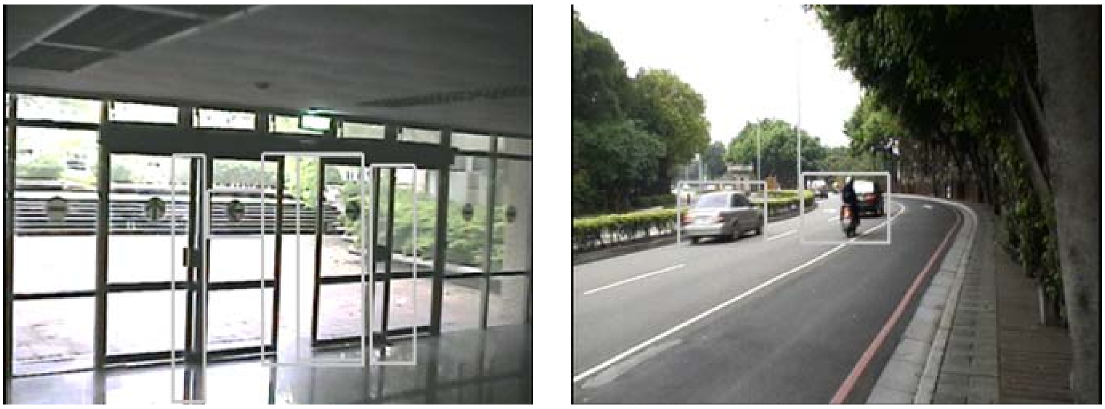


(e) Human walks to 180° (f) Human walks to 225°

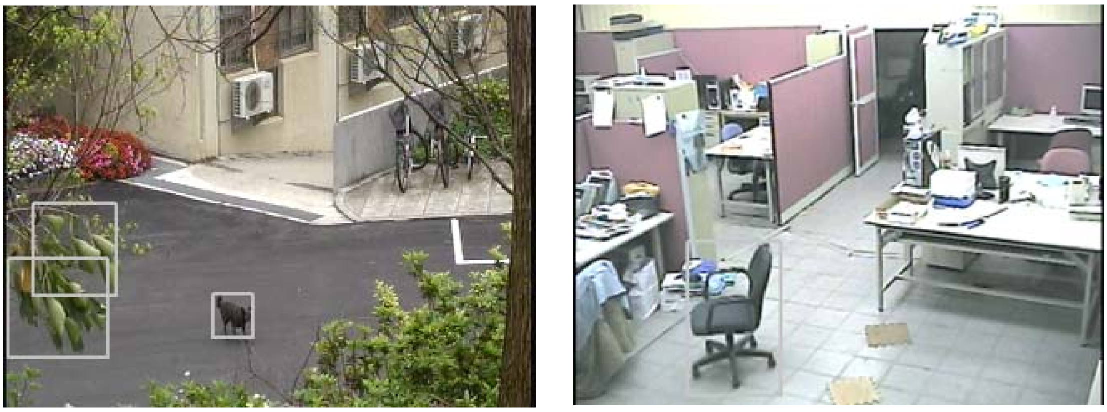
(g) Human walks to 270° (h) Human walks to 315°

Fig. 4-3 : Detection results for multi-direction of human

Following figure shows the detection system able to classify the non-hman objects, such as vehicles, animals, leaves, etc.

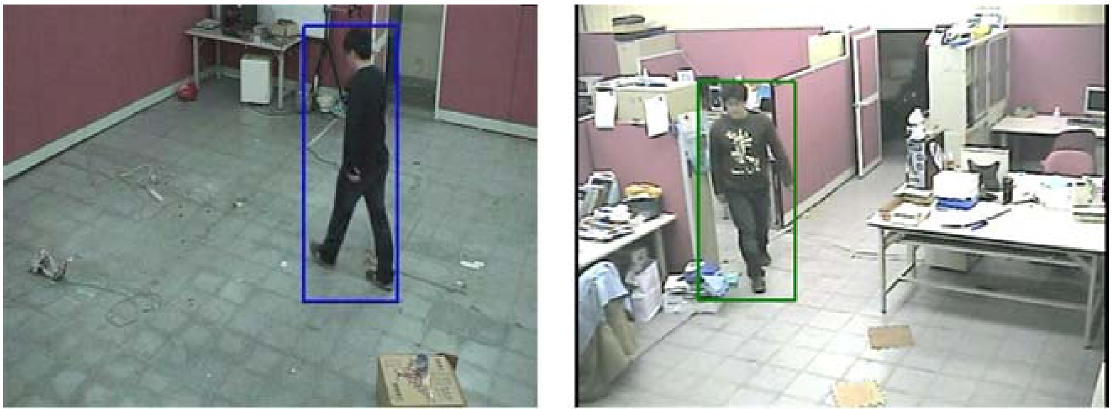


(a) Automatic valve (b) Cars



(c) A dog and leaves (d) A Chair

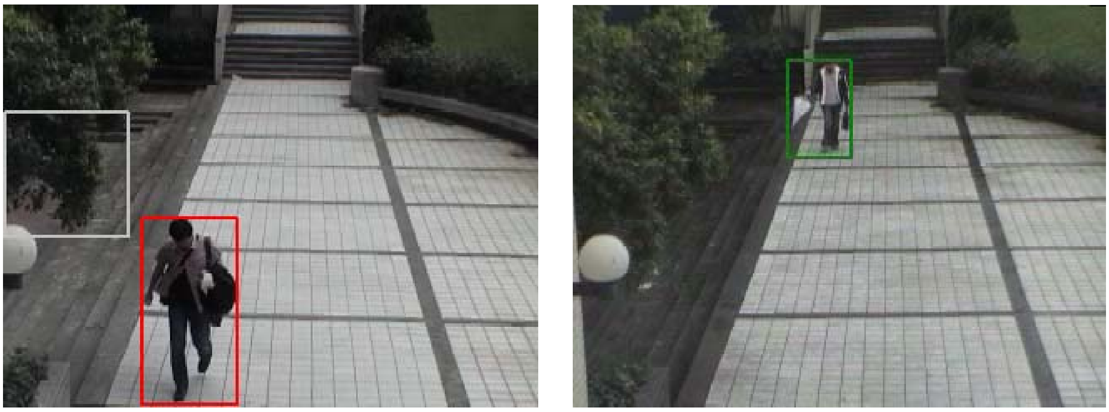
Fig. 4-4 : Detection results of non-human objects



(a) Lateral side of human detection (b) Front side of human detection

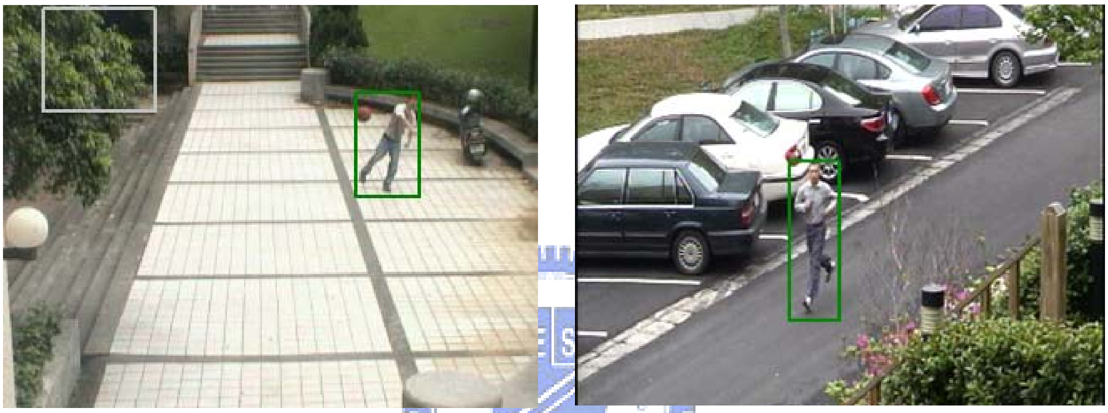
Fig. 4-5 : Results of normal indoor environment

Fig. 4-5 shows the result of frontal and lateral human at indoor environment. People with backpack or carry something with hand are shown in Fig. 4-6. Human have different pose and running through the path are shown in Fig. 4-7.



(a) Human with bag (b) Human with a unbrella

Fig. 4-6 : Results of human carry something.

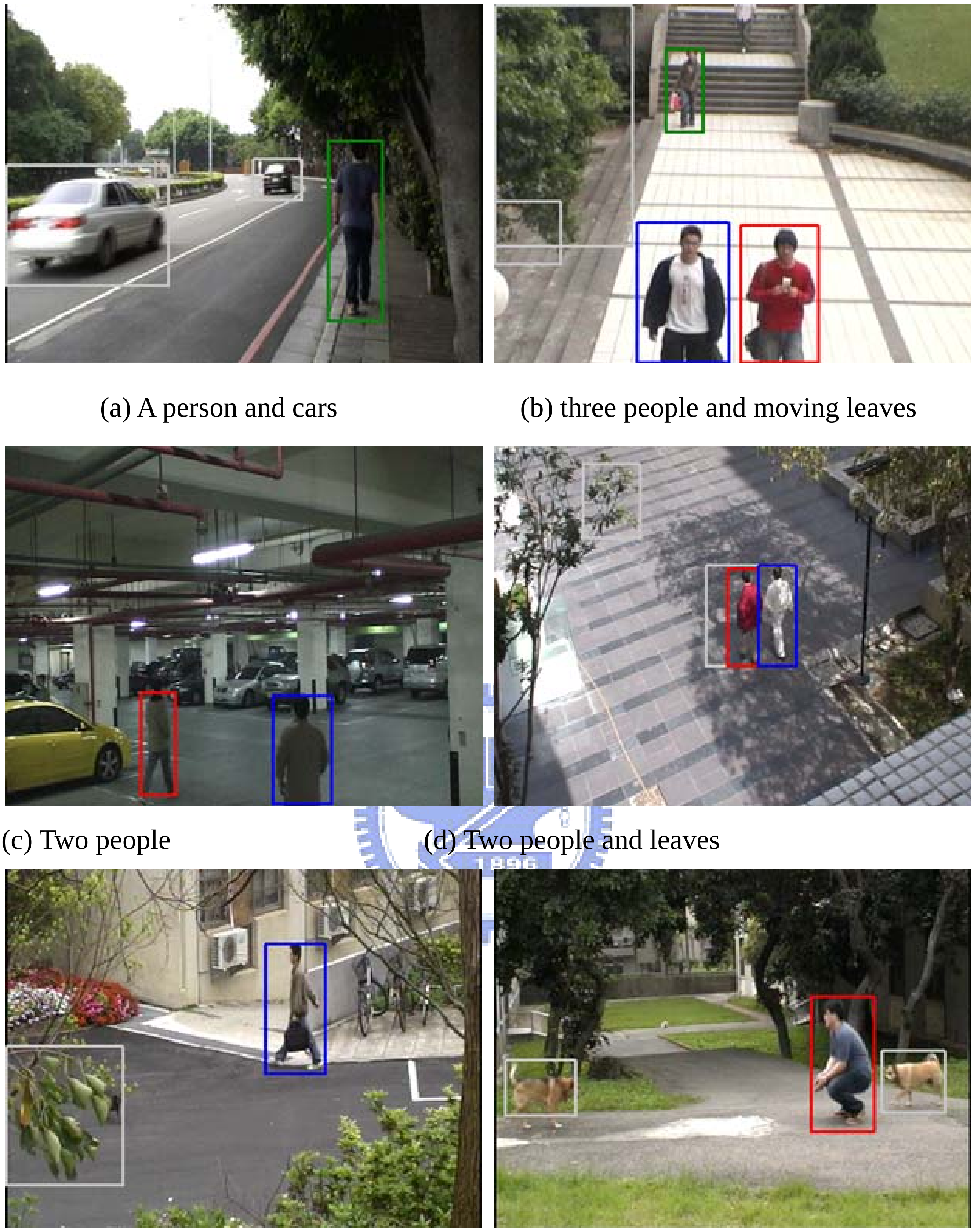


(a) Different pose (b) Running Human

Fig. 4-7 : Results of human with other action.

In different environment, different light, some results of multiple moving objects which include human and non-human in the same frame will be shown in Fig. 4-8. These monitor movements have human, animals or vehicles.

Figure 4-9 simulates a person that the body is partially occluded with other objects.



(a) A person and cars

(b) three people and moving leaves

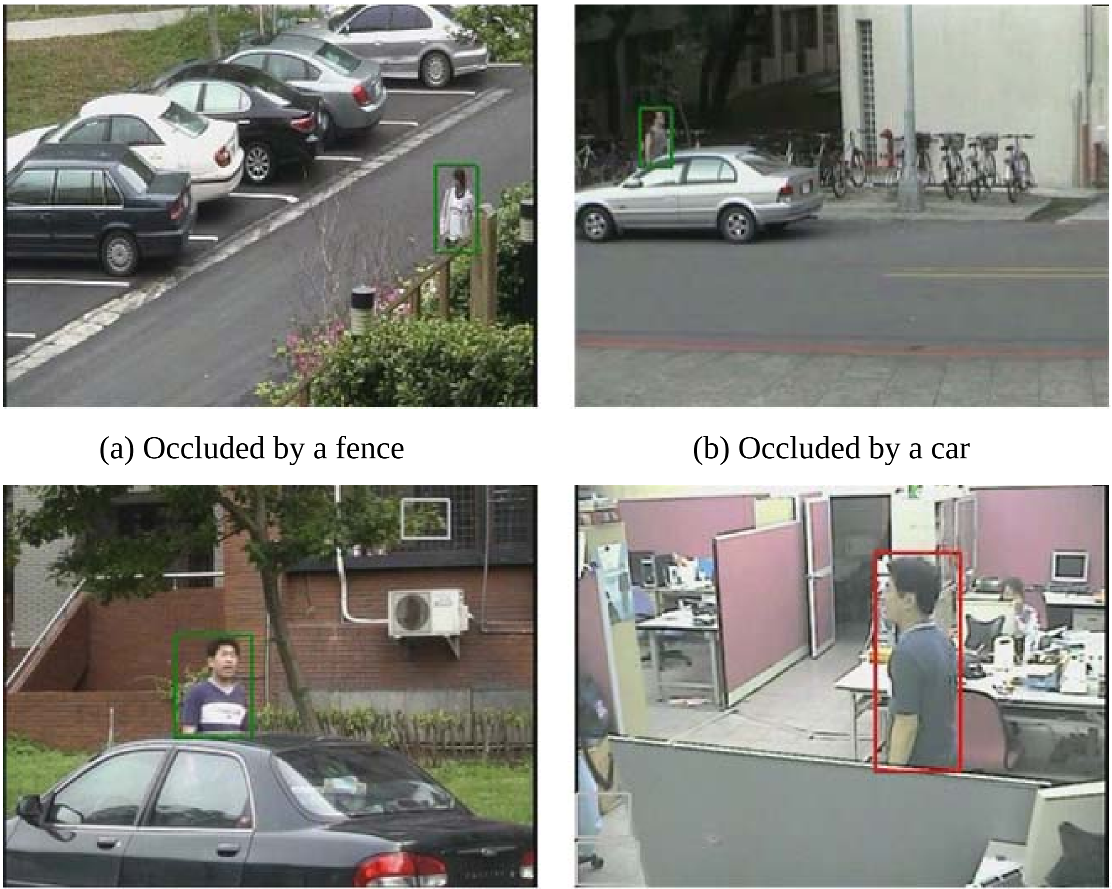
(c) Two people

(d) Two people and leaves

(e) A person and leaves

(f) Two dogs and a person

Fig. 4-8 : Multiple objects in one frame.



(a) Occluded by a fence

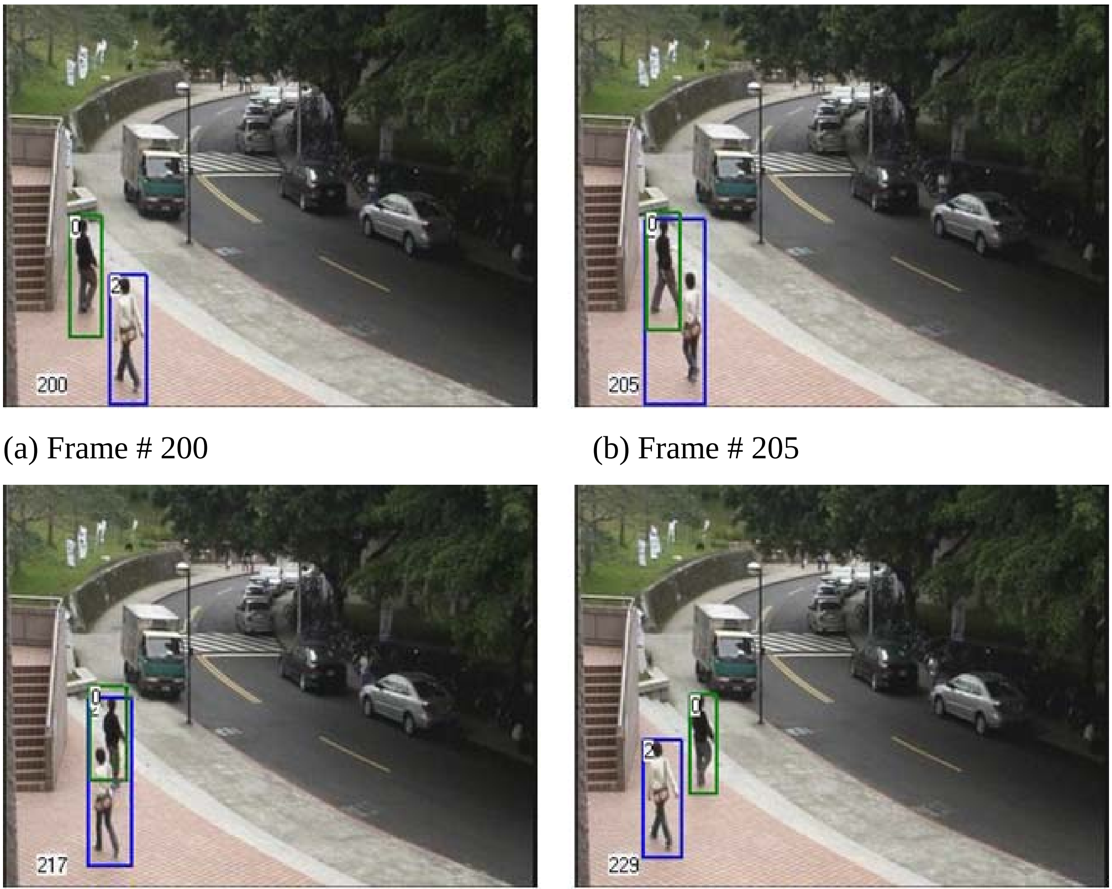
(b) Occluded by a car

(c) Occluded by a car

(d) Occluded by a board

Fig. 4-9 : Results of partially occluded human

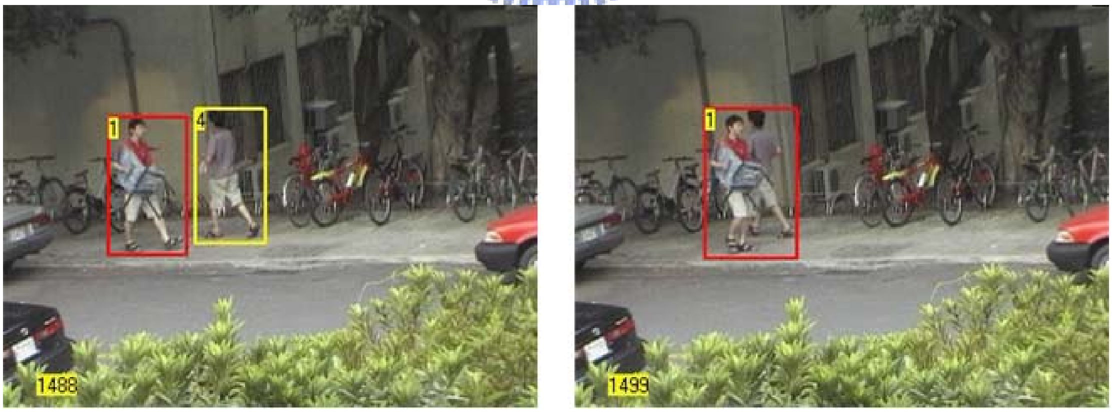
If we confirm that an object is a person after detecting it within a periodic time, then we just need to track it instead of tracking and detecting it at the same time. We can also reducing false alarm by statistics of the detection result. Moreover, tracking processing may help us to analysis the trajectory of the object, and for other behavior analysis. In Fig. 4-10 (b) (c), the position of green black is predicted by Kalman filter, the original foreground combines object 0 and object 2, we only recognize the object 0 at first, but after Kalman filter, the object 2 is also caught in these frames. Figure 4-11 shows the other result of tracking.



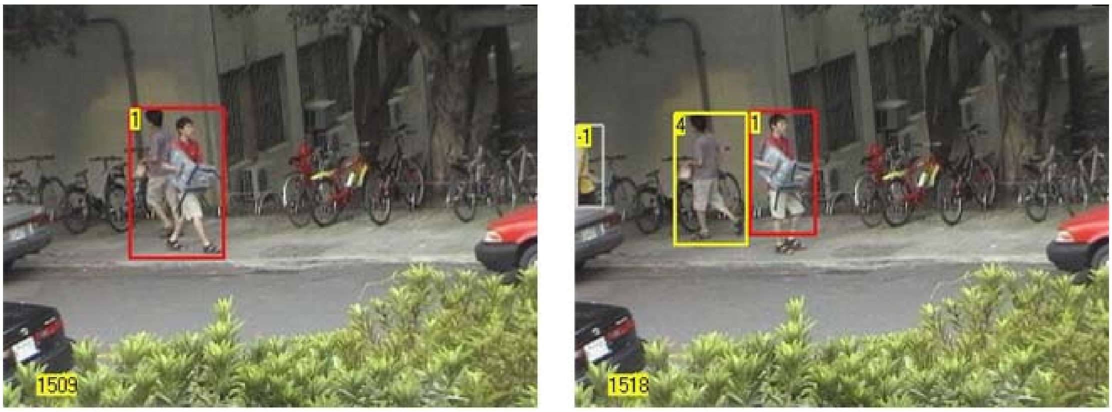
(a) Frame # 200 (b) Frame # 205

(c) Frame # 217 (d) Frame # 229

Fig. 4-10 : The result of Kalman filter



(a) Frame # 1488 (b) Frame # 1499



(c) Frame # 1509 (d) Frame # 1518

Fig. 4-11 :The result of tracking objects

Finally, the accuracy of our system is shown in the following. Here, we establish the training database from 16 different videos, and the testing database from the other 18 different videos. There are 1843 images for positive training data, 2066 images for negative training data, 3178 images for positive testing data, and 2847 images for negative testing data. Each data is normalized to 40 by 40 pixels.

We compare our proposed method with different method. A codebook matching (CBM) algorithm [21] use human shape as the features, and match the moving object with the code vectors of the codebook. The other two ways: ICA-Cosine [9], and ICA -SVM [18] are used for face recognitions, which features are also extracted by ICA. In [9], the features are selected by calculating the ratio of between-class to within-class variability r for each coefficient, the larger r the better distinguishing ability, and cosine similarity measurement is used for classification. In [18], they select the components according to the binary classifier capability which uses Perception or Neural Network, and classifying by SVM. Extracting features using PCA and classifying by Back-Propagation (BP) Neural Network is another way.

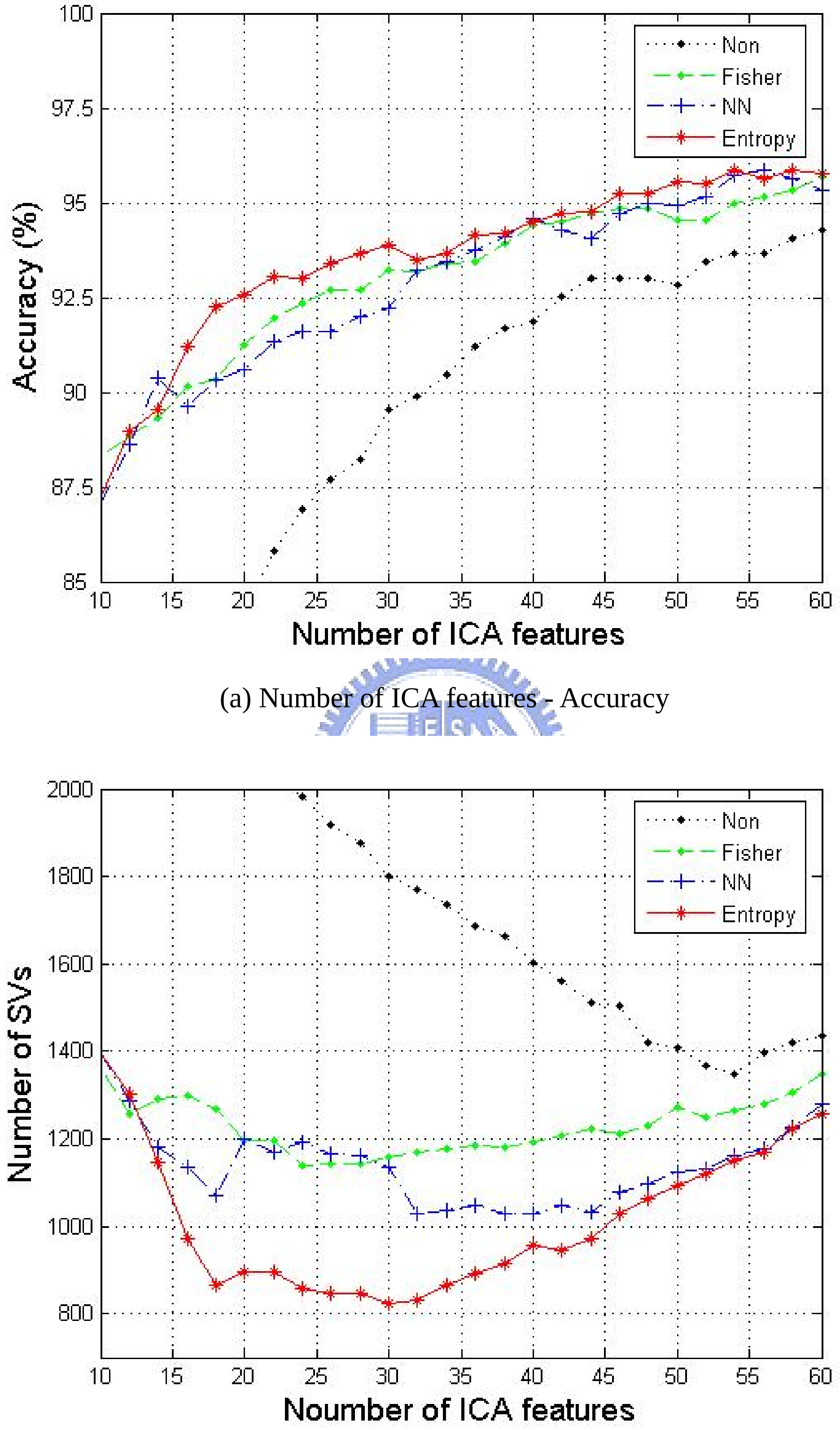
Table 3 : Accuracy of proposed method and the others

Accuracy (%)	**Training Data**		**Testing Data**	
	Human	**NonHuman**	**Human**	**NonHuman**
Features+Classifier	1843	2066	3178	2847
MICA + SVM	97.72	95.84	**94.15**	**93.57**
ICA + Cosine [9]	90.87	85.73	90.34	85.49
ICA + SVM [18]	97.55	93.9	93.17	91.13
CBM [21]	87.95	92.83	90.88	93.68
PCA + BP	99.18	99.46	89.65	94.09

We select 30 bases for ICA and PCA and 40 features of shape to build 256 code vectors in the code book. The result shows in table 3 that our proposed method Modified ICA (MICA) outperforms the others.

Fig. 4-12 illustrate the results of our proposed method and other methods, horizontal axis means the selected number of ICA features, above vertical axis means accuracy of testing data, and under vertical axis means the corresponded number of SVs, where the point-dotted line represents the result before feature selection, the point-dashed line represents the result of Fisher's criterion [9], the plus-dashdot line represents the result of NN [18], and the star-solid line represents the result of our proposed method. The total features after the process of ICA is 76, we select *n* features from them.

We compare each method with 10 to 60 ICA features. In Fig. 4-12, apparently if we did not select a set of better features, but just depend on the creating sort of the bases, the result will very poor for detection. Selecting subsets of coefficients by class discriminability improved the performance of the ICA representation.



(a) Number of ICA features - Accuracy

(b) Number of ICA features - Number of SVs

Fig. 4-12 : Analysis of different feature selection methods

Table 4 : Analysis of computing time

Selection	No. features	No. SVs	Accuracy (%)	ms/object
Entropy	30	825	93.88	**1.13**
Fisher's [9]	30	1157	93.21	1.33
Entropy	40	958	94.51	**1.41**
Fisher's [9]	40	1194	94.4	1.65
NN [18]	40	1028	94.58	1.51

In Fig. 4-12 (a), our proposed method has a greater performance than others, and in Fig. 4-12 (b), the corresponded number of SVs is obviously much less than others. The computation time of detection process are listed in Table 4. Here we use 5 videos, total have 14056 frames, detection times are more than 3000 times. We can see that the costing time of our proposed method is also less than the others, a unit of measurement is millisecond per object. Thus a human detection process which extracting features by ICA, selecting features by conditional entropy, and classifying using SVM, can obtain an optimal result not only in accuracy but also in computing time.

4.3 Discussion

First, table 3 shows that the accuracy of our system is more than ninety percent, and we think it is enough for a warning system. Note that, the training and testing data are included people with full body and half body.

Of course, there still some situation may cause the system fail. Sometime 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 depicted in Fig. 4-13.

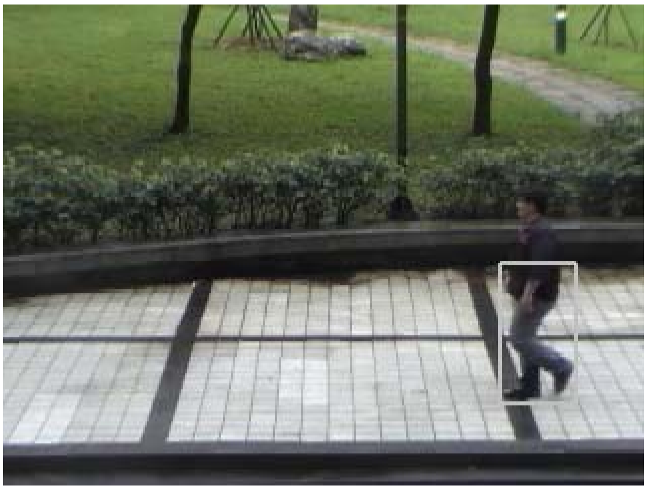


Fig. 4-13 : Example of system fail #1

Because of our shadow elimination algorithm is based on the texture and color relations between shadow range and background, and the threshold to detect shadow is set by the assume of shadow range is much smaller than moving object range, so if the shadow area is too large, or the background texture is not so obvious, the shadow will effect the detection result, see as Fig. 4-14.

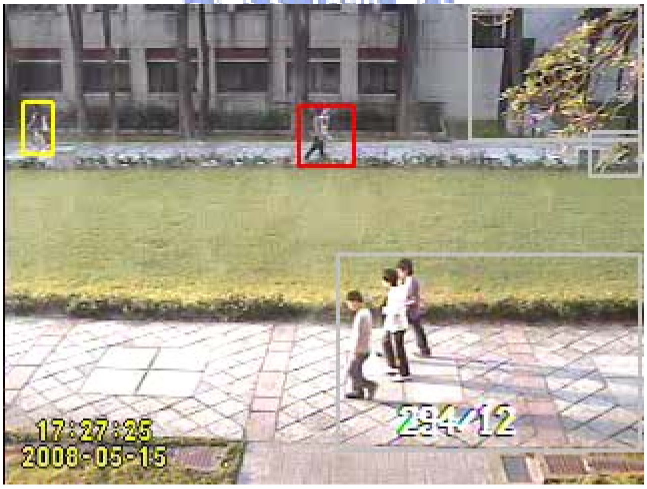


Fig. 4-14 : Example of system fail #2

Another main problem is grouping. Although we have a multiple human separation algorithm, but we just split human when they are walking shoulder by shoulder, and their head is observed. For Kalman filter, it is also useful when the target model are constructed first, if a group of people al the time in the secured area all the time, we have no chance to build the model for each person. The situation can be shown in Fig. 4-15.

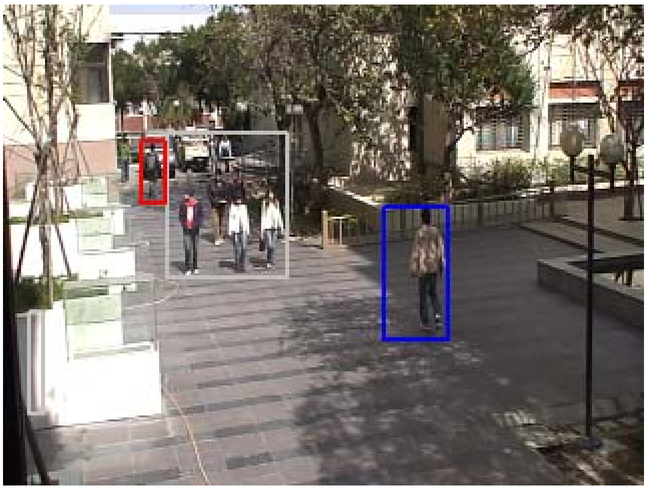


Fig. 4-15 : Example of system fail #3

Because the process of our system does not simplify, it needs so heavy computing load that it can not process in real-time system, this is also a main disadvantage.

Chapter 5

Conclusion and Future Work

In this thesis, we present a system for object-based human detection and tracking. A simple process based on HSV color space is proposed to eliminate shadow for human detection. The experimental results show that the proposed process can actually improve the precision of human detection. For the problem of small groups of people walking partially occluded, we solve them by using a fitting ellipse function which depend on pyramid method, and a simple trajectory tracking based on Kaman filter is used to resolve some other occlusion problem, the tracking sub-system can also decrease the false-alarm rate. ICA have been used in a lot of applications, for example of separating sound or EEG signals , reducing noise, face recognition and so on, but never used in human detection. We not only use ICA for our feature extraction, but also represent a feature selection method to solve the disadvantage of unstable training components. We observed that the class discriminability of independent component does not depend on binary classified capability from the distributions of ICA coefficient, so the conditional entropy is proposed to solving the problem. The conditional entropy if referred to as the entropy of desired output Y conditional on coefficient value X. Moreover, the accuracy of our detection process and computing time are better than other methods, the accuracy is more than 93%.

To improve the performance and the robustness of our system, some enhancements can be done in the future:

(a) A robust shadow elimination algorithm is needed. For our system, we can not handle the large-range shadow. It’s probably to employ an edge based background evaluation method to solve this problem.

(b) For most of case, we can not recognize a person from a group of people, because of our training data is not include the background range, so it is not easy to recognize human if he is not extracted clearly from other image. We may need to training the images include background, thus even the camera is not fixed, the features is also useful.

(c) The computing load of our system is heavy, we need to simplify or find other fast algorithm to reduce the computation time.

References

[1] A. Elgammal, R. Duraiswami, D. Harwood, and L.S. Davis, “Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance,” *Proc. of the IEEE*, Vol. 90, No. 7, July 2002.

[2] Fung G S, Yung N H, Grantham K H, et al. “Effective moving cast shadow detection for monocular color traffic image sequences”. *Optical Engineering,* Vol. 41, No.6, pp. 1425-1440, 2002.

[3] I. Haritaoglu, D. Harwood, and L.S. Davis. “Hydra: Multiple people detection and tracking using silhouettes,” *In IEEE International Workshop on Visual Surveillance*, pp. 6-13, June 1999.

[4] T. Zhao and R. Nevatia. “Tracking multiple humans in complex situations,” *IEEE T. Pattern Analysis and Machine Intelligence*, Vol. 26, No. 9, pp. 1208-1221, Sept. 2004.

[5] S. J. McKenna, S. Jabri, Z. Duric, and A. Rosenfeld, “Tracking groups of people,” *Comput. Vision Image Understanding,* No. 80, pp. 42–56, 2000.

[6] R. Venkatesh Babu, P. P´erez, and P. Bouthemy. “Robust tracking with motion estimation and local kernel-based color modeling.” *Image Vis. Comput. In Pres*s, 2007.

[7] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, S. “Maybank, Principal axis-based correspondence between multiple cameras for people tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28 No. 4, pp. 663–671, April 2006.

[8] T.-W. Lee and M.S. Lewicki. “Unsupervised image classification, segmentation, and enhancement using ICA mixture models.” *IEEE Trans. on Image Processing*, Vol. 11, No. 3, pp. 270–279, 2002.

[9] M.S. Bartlett, J.R. Movellan and T.J. Sejnowski, “Face recognition by independent component analysis. “ *IEEE Transaction on Neural Networks,* Vol. 13, No. 6 , pp. 1450–1464, 2002

[10] C. Liu and H. Wechsler, “Independent component analysis of Gabor features for face recognition,” *IEEE Trans. Neural Networks*, Vol. 14 No. 4, pp. 919–928, 2003

[11] W. J. Gillner, “Motion based vehicle detection on motorways,” *in Proc. of the Intelligent Vehicles '95 Symposium*, pp.483-487, Sept. 1995.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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, pp. 20-22, Sept. 2004.

[17] K Lo, M Yang, R Lin - “Shadow Removal for Foreground Segmentation,” *PSIVT* , *LNCS* 4319, pp. 342-352, 2006.

[18] Y. Ou, X. Wu,H. Qian and Y. Xu, “A Real Time Race Classification System,” *IEEE International Conference on Information Acquisition*, pp. 378-383, 2005

[19] B.A. Draper, K. Baek, M.S. Bartlett, J.R. Beveridge, “Recognizing Faces with PCA and ICA,” *Computer Vision and Image Understanding: special issue on face recognition*, pp. 115-137, 2003

[20] C. Stauffer and W.E.L Grimson, “Adaptive Background Mixture Models for Real-Time tracking,” *In IEEE Conference on Computer Vision and Pattern Recognition*, pp. 246-252, June 1999.

[21] 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.

[22] 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.

[23] C. Huang, T. Chen, S. Li, E. Chang, and J.L. Zhou, “Analysis of speaker variability,” *Proc. European Conference on Speech Communication and Technology*. *Denmark*, Vol. 2, pp. 1377–1380. 2001

[24] 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.

[25] 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, pp. 13-16, August 2006.

[26] R. Polana and R. Nelson, "Detecting activities," *IEEE Computer Society Conferenceon Computer Vision and Pattern Recognition*, pp. 2-7, 1993.

[27] M. Bertozzi, A. Brogi, M. Del Rose, M. Felisa, A. Rakotomamonjy and F. Suard, "A Pedestrian Detector Using Histograms of Oriented Gradients and a Support Vector Machine Classifier," *IEEE Intelligent Transportation Systems Conference*, pp. 143-148, 2007

[28] C.C. Chang and C.J. Lin, "LIBSVM: a Library for Support Vector Machines," June 14, 2007. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[29] A. Cichocki and S. Amari, "Adaptive Blind Signal and Image Processing," Wiley, 2002.

[30] A. Hyvarinen, J. Karhunen, E. Oja, "Independent Component Analysis," Wiley New York, 2001.

[31] Abe, Shigeo, "Support Vector Machines for Pattern Classification," London :Springer-Verlag London Limited, 2005.

[32] L Wang ed., "Support Vector Machines: Theory and Applications," New York: Springer, Berlin Heidelberg, 2005.

[33] S. R. Gunn, "Support Vector Machines for Classification and Regression," Technical Report, May 1998.