# 國立交通大學

## 電控工程研究所

## 碩士論文

# 結合獨立成份與色彩特徵之平均移動向量人形追蹤演算法-應用於主動式攝影機

# Mean-shift human tracking based on combination of ICA and color feature on active camera.

研究生：劉哲男

指導教授：林進燈 博士

中華民國 九十九 年 七 月

# 結合獨立成份與色彩特徵之平均移動向量人形追蹤演算法

# -應用於主動式攝影機

# Mean-shift human tracking based on combination of ICA and color feature on active camera.

研 究 生：劉哲男　　Student：Che-Nan Liu

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

國立交通大學

電控工程研究所

碩士論文

A Thesis

Submitted to Institute of Electrical Control Engineering

College of Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master

in

Electrical and Control Engineering

June 2010

Hsinchu, Taiwan, Republic of China

中華民國 九十九 年 七 月

# 結合獨立成份與色彩特徵之平均移動向量人形追蹤演算法-應用於主動式攝影機

學生：劉哲男　　　　　　　　　　指導教授：林進燈 博士

國立交通大學電控工程研究所

## 摘要

近幾年來，基於視覺的偵測和追蹤在電腦視覺領域上是一項很重要的課題，基於視覺的監控系統已經被廣泛的應用在停車，病人監控以及安全監控...等領域上。在本論文當中實現了在主動式攝影機上完成人物偵測和追蹤，在過去主動式攝影機追蹤是利用影像相減的方式找出移動物體的位置，透過移動物體的位置來驅動攝影機的雲台，雖然這樣的方法可以對移動物體作追蹤但是為了要找到移動物體的位置攝影機在移動的過程中必須停下來才可以做影像相減，所以會造成攝影機無法連續的控制雲台，換句話說，假如雲台持續的轉動，因攝影機的轉動會得到模糊的影像而且相減時不僅會得到移動物體也會得到包含背景的資訊。針對利用相減的方法無法找到移動物體精確的位置，因此我們提出結合獨立成份與色彩特徵之平均移動向量人形追蹤演算法應用於主動式攝影機來解決上述的問題。

本論文主要分成三大部分，分別是人物偵測、追蹤和攝影機的控制。利用人物偵測系統獨立成分分析和支持向量機(Support Vector Machines)分類現在畫面中的移動物體是人還是非人。當我們系統偵測到人之後就會被鎖定，利用平均移動向量演算法在每一張輸入的影像計算出相似度並且送出控制命令給主動式攝影機，使畫面中的人可以保持在我們所監控的影像中。有時候人會整個或部分的被其他物體遮蔽，因此相似度會劇烈的下降，使平均移動向量在追蹤的時候無法

追蹤目標物，為了解決遮蔽的情況我們採用卡爾曼濾波器對目標物做位置的預測。

當目標物和背景有相同顏色的情況下只使用顏色當作平均移動向量追蹤特徵會有遺失的現象。為了解決遺失的現象，我們提出一個結合獨立成分分析和顏色當特徵的先進平均移動向量演算。因為獨立成分模組在訓練時所輸入的灰階影像具有人物的特性，因此我們提出來的驗算法可以有效在相同顏色下判斷人或背景。

# Mean-shift human tracking based on combination of ICA and color feature on active camera

Student: Che-Nan Liu　　　　Advisor: Dr. Chin-Teng Lin

Institute of Electrical Control Engineering
National Chiao Tung University

## Abstract

In recent years, detection and tracking are important tasks in computer vision for visual-based surveillance system. Visual-based surveillance system is a widespread application in parking, patient monitoring and security surveillance fields. In this thesis, we use human detection and tracking algorithm based on active camera. In the past, active camera based object tracking used temporal difference to find object position and then drive pan or tilt command to control the active camera. Although this process can achieve moving object tracking. However, to find moving object position, the active camera should be stopped for the computation of temporal difference. Therefore, the active camera can not pan/tilt continuously and smoothly. In the other words, if the active camera is able to keep moving the whole time, we will capture blur images, and temporal difference will extract not only moving object but also background. Therefore, it is impossible to accurately locate the position of moving object by using temporal difference while the active camera is moving. So we propose mean-shift human tracking based on combination of ICA and color feature on active camera to solve above problem.

This thesis consists of three major parts: Human detection, human tracking and

pan/tilt control. In human detection system, the independent component analysis (ICA) and support machine vector (SVM) classifier are applied to classify moving objects into human or non-human. When a human is detected then we need to track it. Mean-shift algorithm will track the target by computing the similarity value in every frame and send the position to active camera, and then active camera will drive PTZ to keep the target in the center of FOV (field of view). Sometimes human will be partially or fully occluded by other object, thus the similarity will drastically decrease. Consequently, mean-shift will miss the target. To overcome the above problem, the Kalman filter is applied to predict the target's position in next frame.

The mean-shift using only color feature will miss the tracking target, when the target object and background have the same color. In order to solve the missing problem, we propose a novel mean-shift algorithm based on combination of ICA and color feature. Since the ICAs modeled human characteristic with gray-level training data, the proposed algorithm can distinguish human and background with the same color.

# 致　　謝

兩年的碩士班生活即將告一段落，能夠在此經過各位老師的指點是我此生的榮幸，尤其是我的指導教授林進燈博士，這兩年來的細心指導與督促，讓我學習到許多寶貴的知識與經驗，在學業及研究方法上也受益良多。另外也要感謝口試委員陶金旺教授以及許騰尹教授在口試時的指點與鼓勵，使得本論文更為完整。

其次感謝超視覺影像實驗室的大家長鶴章、剛維學長的指導，以及博士班的學長姐肇廷、Linda 以及東霖給予我指導與建議，尤其是 Linda 學姊給予我許多建議與方向，在生產前一天還是持續給予指導與論文上的指點，直到最後一刻才放鬆心情生產，本論文多虧學姊的幫忙。也感謝同學佳芳、聖傑、嶸健及健豪的相互砥礪，還有實驗室的學弟妹郁光、開澄、奎理、庭立、奕竹、慶峰及皕文，在生活上給予諸多的協助，讓我在這兩年的研究生涯，無論是課業上、學業上或是生活上都不孤單。

最後，更要感謝我的父母，從小到大的教育及栽培，讓我有勇氣去追求我想要的，讓我無後顧之憂，如果沒有他們無怨無悔的付出，沒有他們就沒有今日的我。

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

In recent years, detection and tracking are important task in computer vision for visual-based surveillance system. Visual-based surveillance system has widespread application in parking, patient monitoring and security surveillance fields, etc. In visual-based surveillance, human eyes are used to watch surveillance in front of monitor in the past years. Although using human eyes to watch surveillance system works well, but it still has some problems. For example, finding human resource to watch monitor all day long is expensive and it can not achieve real-time surveillance. In order to achieve real-time and all day long surveillance, automatic visual surveillance in computer vision plays an important role in security area.

Human detection and tracking surveillance are important topics in computer vision and security area. The human detection system consists of two parts: moving object extraction and human classification. The moving object extraction to extracting objects from background and locate its position and size. Meanwhile, the human classification is classifying the moving object into human or other objects. After human detection, the human tracking system will trace the target object all the time. Sometimes, target human would be occluded with other object while tracking, thus tracking system which has prediction ability is needed in this case.

There are two kinds of camera used in security surveillance, fixed and active camera. The fixed camera is very popular and low cost, but it's FOV (filed of view) limited. If human is walking out of monitoring area, tracking system can not keep monitoring. In order to keep target human in the camera scene, active camera is the best choice because it has the ability to perform pan, tilt and zoom in/out automatically. In this thesis, we integrate the real-time human detection, human tracking and kalman filter to achieve an active camera-based visual surveillance system.

## 1.1 Motivation

In this thesis, we present a human tracking system on active camera. We concern on the active camera because the ability of active camera to drive pan-tilt-zoom (PTZ) to keep the target human in the monitoring area (FOV). Mostly, tracking system on active camera only track a moving objects even the moving object is not human being. Therefore, it motivates us to research a system which used active camera and has the ability to track only target human. We proposed a human detection system using ICA (independent component analysis) base on conditional entropy for feature extraction and SVM (support vector machine) to classify human or other objects.

Generally, tracking system on active camera is using temporal difference to extract moving object from background. It will cause the active camera unable to continuously and smoothly drive PTZ. In the other words, if the camera still drive PTZ, we will capture blur images and temporal difference will extract not only moving object but also object that have edge includes background. Therefore, it will impossible accurately to locate the position of moving object. We applied mean-shift tracking algorithm to solve this problem. The mean-shift algorithm will locate the maxima of a density function given histogram feature through iteration and it works frame by frame. The histogram feature is combination of color and ICA features. The color feature will keep the human color information for example clothes, meanwhile ICA feature will keep the feature of human being.

When target human partially or fully occluded with other object, we will use Kalman filter to predict the position in the next frame.

## 1.2 Related Work

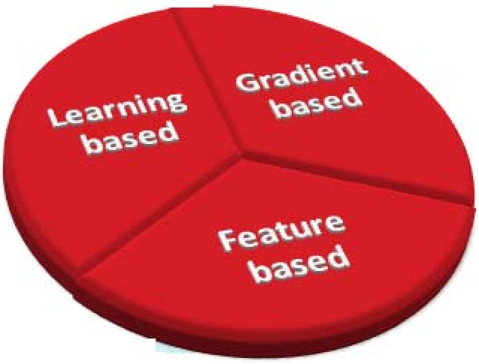


Fig. 1-1 Three categories of tracking

In recent years, many human detection systems are developed. The human detection system determines the object position and size in the image. A good human detection system can provide valuable insight on how one might approach other similar feature and pattern detection system. There are two parts of human detection system, segmentation of moving object from background and discrimination of humans from nonhuman objects. A moving object occurred in the image will be extracted from background image in the human detection system. There are many human detection method are used to detect the human that presence in an image. Optical flow is used to estimate independently moving object, but it expense complex computation and sensitive to change of intensity. Optical flow is used in [2,6] to detect vehicle. Zhao *et al* [8] 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. Dalal and Triggs [7] use gray scale image to get edge image and using it to extract orientated gradient. They select the dominant orientated gradients to detect human. In [8] a stereo-based segmentation algorithm is used to extract objects from the background, followed by a neural network-based recognition. Sebastian and Alvaro [10] present a new computer vision algorithm designed to operate with moving cameras and to detect humans in different poses under partial or complete view of the human body. Boosting detector cascades have been introduced by Viola *et al*. [13]. In this, AdaBoost is used in each layer to iteratively construct a strong classifier guided by user-specified performance criteria. Viola *et al*. contribute to the high processing speed of the cascade approach, since usually only a few feature evaluations in the early cascade layers are necessary to quickly reject non-pedestrian examples. In [14] presents a template-based approach to detecting human silhouettes in a specific walking pose, templates consist of short sequences of 2D silhouettes obtained from motion capture data.

Human tracking system is used to follow target human through the sequence images in terms of changes in scale and position. As mentioned in introduction, there are two type of real-time tracking system. One is tracking object with fixed camera and the other is with an active camera. Tracking object with an active camera can keep the object in the scene of camera by drive pan and tilt. The active camera also have zoom in/out function that can used to zoom in/out object when the object's image resolution is too small to track. The active camera used in surveillance system needs to achieve real-time tracking. If an tracking algorithm expense more computation time then the active camera will not track object immediately. Many tracking system are used active camera work only on pan-tilt or zoom. The objective of our approach is to correctly select the target human, and drive pan/tilt to keep target in the center of FOV. If the target's size smaller or larger than a minimum or maximum size the zoom function will operate zoom-in or zoom-out. There are many application of active camera. Murray *et al.* [30] utilized morphological filtering of motion images for background compensation. This motion tracking method can track a moving object from dynamic images with pan/tilt angles. C. Lin *et al.* [31] use an image mosaic technique to track moving objects with a single pan/tilt camera indoors. Collins *et al.* [32] developed a system with multiple cameras that tracked a moving figure using pan/tilt cameras alone. This system used a kernel-based tracking approach to overcome the apparent motion of the background as the camera moved. L. Fiore *et al.* [33] use wide angle and active camera to achieve human tracking. In this work the target object was found by wide angle camera and through camera calibration method tell active camera the pan/tilt angle to track the object.

In Fig. 1-1, there are three kinds of tracking method. Feature based is the most commonly method. Color, edge or motion are commonly used as tracking feature. The edge detection methods, such as Sobel method [3], Laplacian method [3], and Marr–Hildreth method [4] etc., utilize masks to do convolution on the image to detect the edges based on the abrupt change of the gray level. Wei Guo *et al.* [1] proposed human tracking system based on shape analysis. Law et al. [5] design fuzzy rules use in edge based human tracking, although this method requires a rather large and complicated rules set. However, these methods are need more computation time and edge pixels can not be always detected continuously. We noted that all those methods mentioned above detect edges using gray level images, and those methods will be neglect for color images because the representation of a pixel is not only a gray level but a vector in a color space. The edge occurring in the adjacent pixels which have the same values in any one color component may not be detected. So edge detection only in gray level image is not sufficient and robust. Pattern recognition learning the target object to search them in sequence image can achieve human tracking. Williams et al.

[26] extended the approach to the nonlinear translation predictors learned by Relevance Vector Machine. Agarwal and Triggs [27] used RVM to learn the linear and nonlinear mapping for tracking of 3D human poses from silhouettes. Bohyung Han and Larry Davis [28] use PCA to extract feature from color and use these feature in mean-shift algorithm to implement object tracking. Robert T. *et al.* [29] presents an online feature selection mechanism for evaluating multiple features while tracking and adjusting the set of features used to improve tracking performance. Their feature evaluation mechanism is embedded in a mean-shift tracking system adaptively selects features for tracking. The mean-shift algorithm was originally proposed by Fukunaga and Hosterler [9] for clustering data. The kernel-based object tracking proposed by Meer *et al* [19]. This method tracks an object region represented by a spatial weighted intensity histogram. So the target and candidate object distinguish whether they are similar by Bhattacharyya coefficient, and tracking is achieve by optimizing this objective function using the iterative mean-shift algorithm. Later, many variants of the mean-shift algorithm were proposed for various applications [20-23].

Though the mean-shift object tracking algorithm performs well on sequences with relatively small object displacement, its performance is not guaranteed when objects undergo partial or full occlusion. In order to improve the performance of mean-shift tracker, in the event of object undergoing partial occlusion, there some method was be proposed Kalman filter [34,35] and particle filter [24,25]. K. Nummiaro *et al* [25] use the idea of particle filter to apply a recursive Bayesian filter based on sample sets. They use color as feature. Their work have evolved from the condensation algorithm which was developed in the computer vision community.

Plamen P. *et al* [15] use the mean-shift method for face detection and automated control of an active camera to follow a person’s face and keep his image centered in the camera view.

# Chapter 2

# System Overview

## 2.1 Active camera control

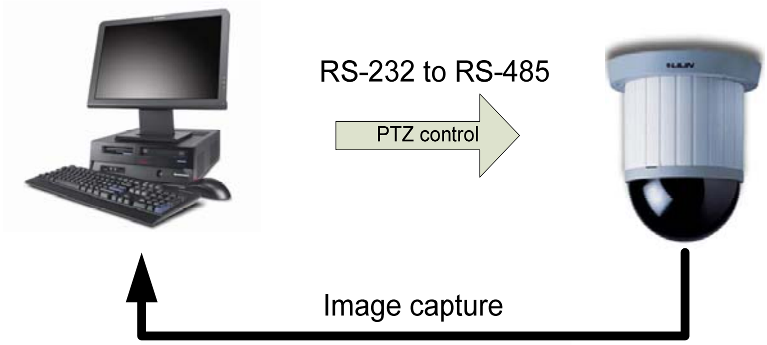


Fig.2-1 Active camera control through RS-485

The proposed system is using active camera to track and keep human in the center of monitor screen or camera's FOV (field of view). The active camera is controlled by pelco P-protocol [16] through RS-232 to RS-485 converter. We have to control pan (left, right direction), tilt (up, down direction) angle, and zoom's step to achieve our tracking purpose.

The pelco P-protocol has 8 bytes data with message format as show in Fig. 2-2. Byte1 and byte7 are start and stop byte, and always set to 0xA0 and 0xAF, respectively. Byte2 is the receiver or camera address. In our case, we only use one camera therefore byte2 always sets to 0x01. Byte3-6 are use to control pan-tilt-zoom ( PTZ) as shown in Fig. 2-3. The last byte is an XOR check sum byte.

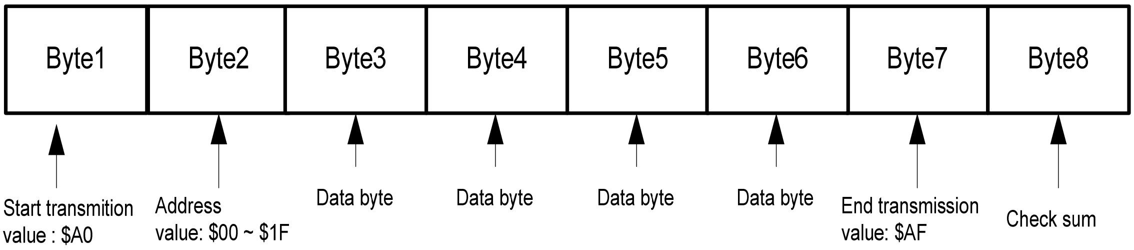


Fig.2-2 message format

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Data byte1 | Fixed to 0 | Camera on | Auto scan on | Camera On/Off | Iris close | Iris open | Focus near | Focus far |
| Data byte2 | Fixed to 0 | Zoom wide | Zoom tele | Tilt down | Tilt up | Pan left | Pan right | 0 (For pan/tilt) |
| Data byte3 | Pan speed $00 (stop) to $3F (high speed) and %40 for Turbo | | | | | | | |
| Data byte4 | Tilt speed $00 (stop) to $3F (high speed) | | | | | | | |

Fig2-3 Data byte 1 to 4 format

In this thesis, we divide the image view into 25 regions to drive PTZ and keep moving object in the center of FOV. Every region has specific direction, angle and speed as shown in Fig. 2-5. If the target is located on stop-region, then camera will set to stop. Otherwise, if the target is located on other regions. For example in the region H the camera will drive to up direction, so on. The zoom-in and zoom-out will be activated if the target's size become smaller or larger than user's define size. For example, if the target's size 1.5 times larger than user's define size then the zoom-out will activated. Otherwise, if the target's size is 0.5 times smaller than user's define size, then zoom-in will activated. During the period of the active camera moving, the tracking system is still running as well as the camera control system. So, if the moving object changes its direction, we can easily change the control action immediately.

| | | | | |
|---|---|---|---|---|
| A | B | C | D | E |
| F | G | H | I | J |
| K | L | Stop | M | N |
| O | P | Q | R | S |
| T | U | V | W | X |

Fig.2-4 Divide screen into 25 regions

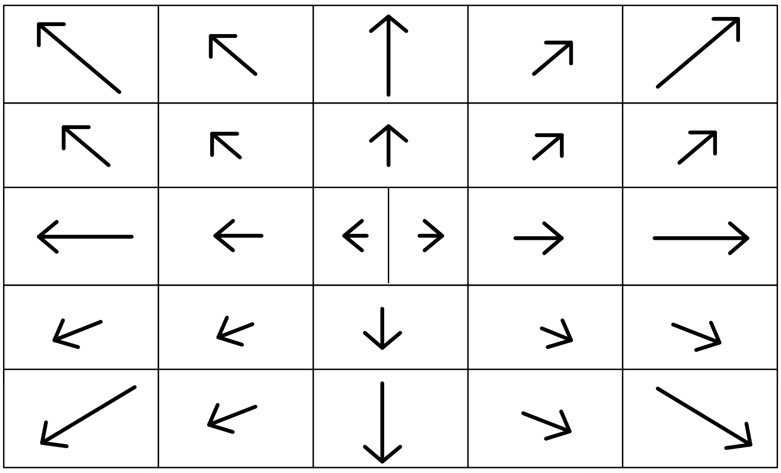


Fig.2-5 Control direction for each regions

## 2.2 Software system

The whole system consists of three major parts: Human detection, human tracking and PTZ control. Input frame are captured by active camera with resolution 320x240 and moving object will be extracted by background difference. We use previous 20 frames to build a background image. The independent component analysis (ICA) and support machine vector (SVM) classifier are applied to classify the moving object into human or non-human. When a human is detected then we need to focus on it. Mean-shift algorithm will track the target by compute the similarity value in every frames and send the position to active camera, then active camera will drive PTZ to keep the target in the center of FOV. Sometimes human will be partially or fully occluded by other object, thus the similarity will drastically decrease, consequently mean-shift will miss track the target. In this case, the Kalman filter is activated to predict the target's position in next frame.

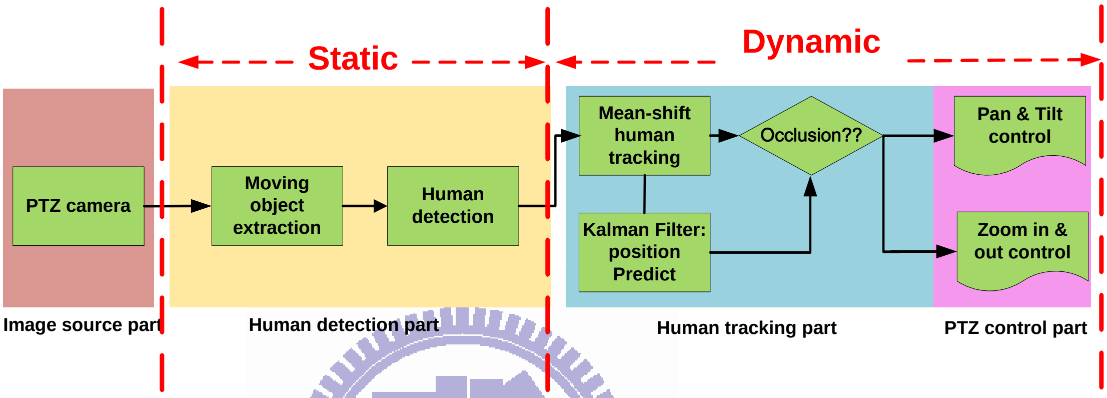


Fig2-2 system overview

# Chapter 3

# Object extraction and Human detection

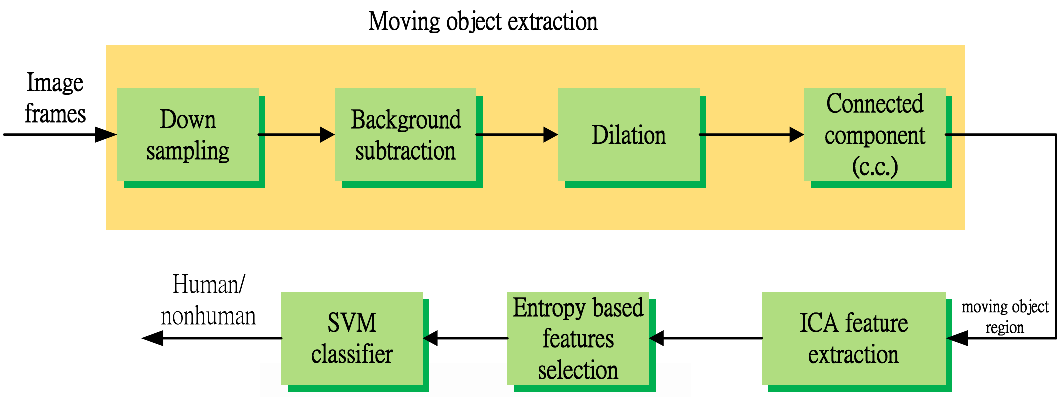


Fig. 3-1 Human detection system

Chapter 3 describes moving object extraction, including preprocessing and classifying moving object into human and nonhuman. The architecture of moving object extraction was indicated in the dotted-line block in Fig. 3-1 and the remained blocks represented our human feature extraction and classification.

## 3.1 Moving object extraction

Mostly, in surveillance system, the default position of camera is fixed even the camera is an active camera, so we can use the still image as background image. Background subtraction is the simplest way to extract moving object from an image. Besides, we used the background subtraction method in order to meet the real-time requirement.

Our background image was constructed by using first 20 frames. The difference for each pixel (x, y) could be calculated by

$$I_{BS}(x,y)=|I_C(x,y)-I_B(x,y)| \quad (3.1)$$

where $I_C$ and $I_B$ denote the current and background gray image, respectively. $I_{BS}$ denotes the background subtraction image. A threshold value ths is choose to produce binary moving object $M_{obj}$ as described in following equation.

$$M_{obj}(x,y)=\begin{cases}1 & if \quad I_{BS}(x,y)\geq ths \\ 0 & id \quad I_{BS}(x,y)< ths\end{cases} \quad (3.2)$$

The dilation process applied on $M_{obj}$ to gradually enlarge the boundaries of moving object pixels (foreground). Thus areas of foreground pixels grow in size while holes within those regions become smaller.

$$T_{foreground}=\sum_{i=-1}^{1}\sum_{j=-1}^{1}M_{obj}(i,j) \quad (3.3)$$

$$I_D(i,j)=\begin{cases}1 & if \quad T_{foreground}>1 \\ 0 & otherwise\end{cases} \quad (3.4)$$

$T_{foreground}$ in Eq. 3.3 denotes the total foreground pixel in a (3x3) dilation mask. If at least one pixel coincides with foreground pixel, then the center of (3x3) mask is sets to the foreground value. If all the corresponding pixels are background then it will set to background value.

Connectivity between pixels is a fundamental concept that simplifies the definition of numerous digital image concepts, such as regions and boundaries. To establish whether these two pixels are connected, it is determined by their neighbors and finds their gray levels satisfy a specified criterion or similarity. For instance, in binary image with values 0 and 1, two pixels maybe 4-neighbors, but they are said to be connected only if they have the same value. Connected component works by scanning an image, pixel-by-pixel in order to identify connected pixel regions and labeling the pixel that connected together with same label. Once all groups have been determined, each pixel is labeled with a gray level or a color according to the component it was assigned to as shown in Fig. 3-2. After connected component, we observe the exist labels. In actual situation not all of them are belong to moving object pixels, therefore we use a (9x9) size filter to eliminate noise labels. The connected component without noise clearly observed in Fig. 3-2. All of these moving object extract result are show in Fig. 3-3.

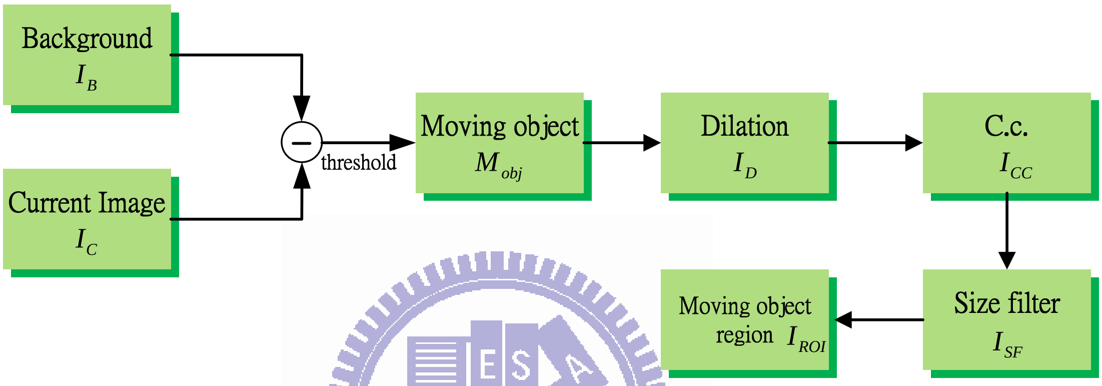
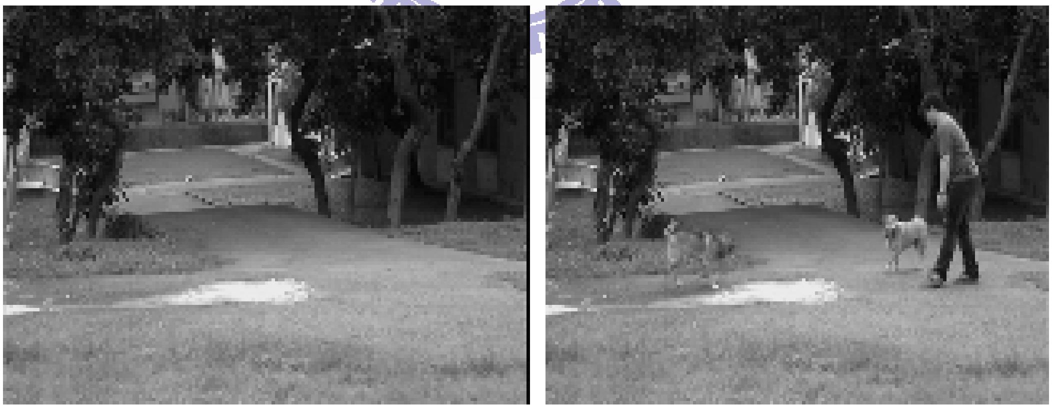
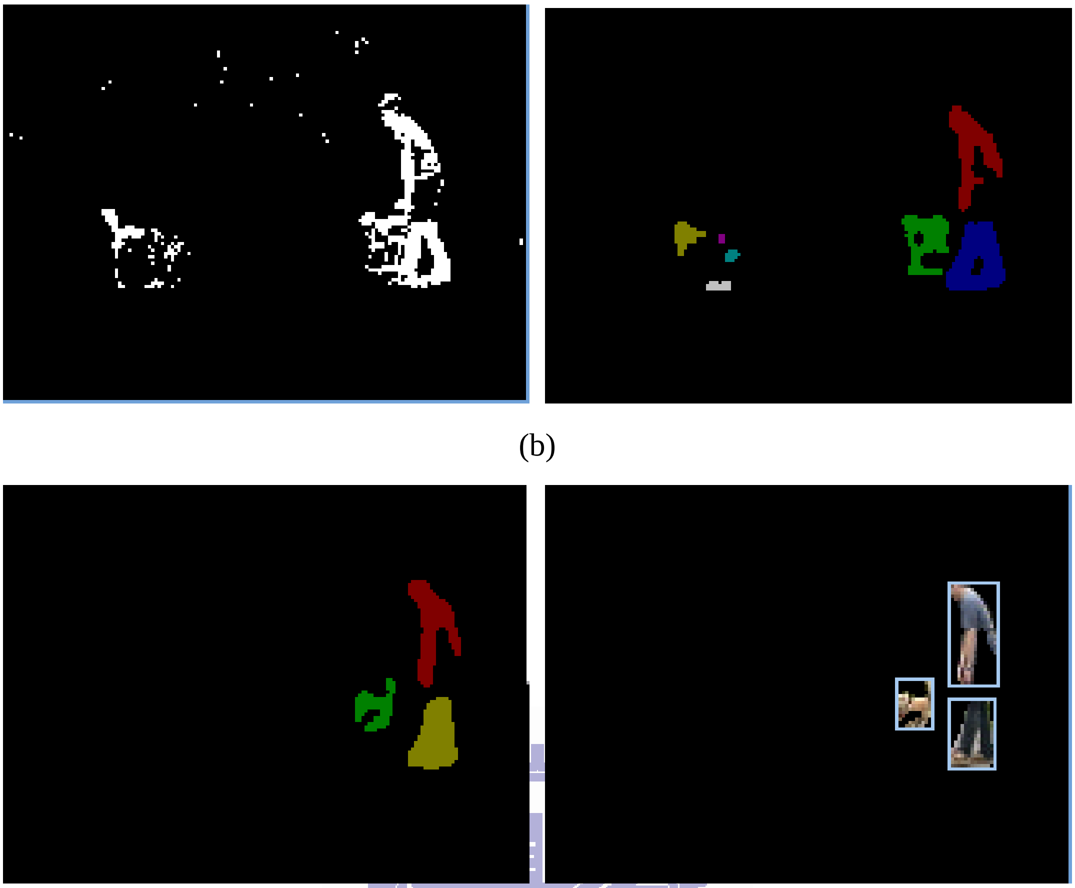


Fig. 3-2 Moving object extraction



(a)



(b)

(c)

Fig. 3-3 Moving object extract processing. (a) Left is $I_B$. Right is $I_C$. (b) left is $M_{obj}$. Right is $I_{CC}$. (c) Left is $I_{SF}$. Right is $I_{ROI}$.

## 3.2 Human detection

We applied independent component analysis (ICA) on the foreground regions for feature extraction proposed. Then classified it into human or nonhuman by SVM classifier shown in Fig. 3-4.

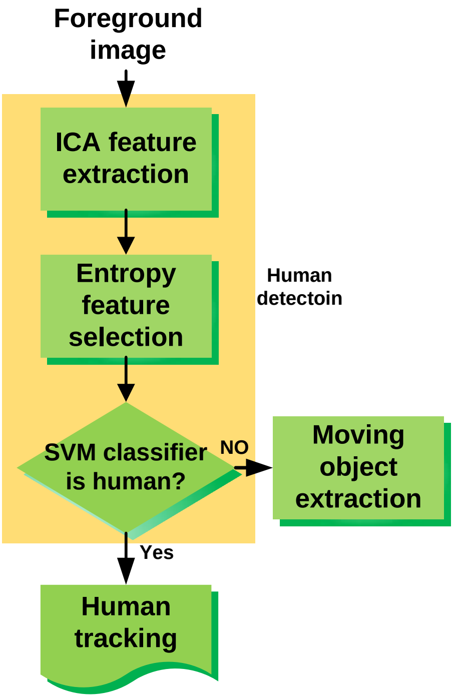


Fig. 3-4 Human detection flow chart

### 3.2.1 ICA feature extraction

In recent years, ICA [38] has been applied to human feature extraction for constructing a sufficient set of features describing human beings. 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. As Fig. 3-5 shows ICA transforms each input image to the combination of bases and its corresponding coefficients.

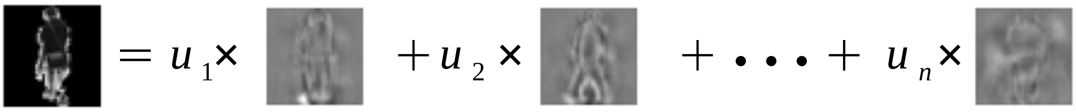


Fig. 3-5 Image decomposition.

Let us have $m$ training images which include both human and non-human with images size are ($n_r \times n_c$). Reshape each image into a N-vector, then, the mixture data

**X** is an (m x n) matrix.

Also the mixture data $x_1, x_2, ..., x_m$ are the linear combination of *n* independent and zero-mean of the source signal $s_1, s_2, ..., s_n$ (typically *m* $\geq n$) as described in

$$x_j = h_{j1}s_1 + h_{j2}s_2 + ... + h_{jn}s_n \qquad (3.5)$$

The matrix **H** is expressed in terms of the elements $h_{ji}$, and it is an unknown full rank (*m* x *N*) mixture matrix. Since all vectors are column vectors and the transpose of **X** is a row vector, we can rewrite Eq. 3.5 to Eq. 3.6 by using vector-matrix notations

$$\mathbf{X=HS} \qquad (3.6)$$

Without loss of generality, we assume that both the mixture variables and independent components have zero mean and non-Gaussian distributions. For the nonzero mean distributions, the observable variables $x_j$ can always be centered by subtracting the sample mean to become the zero mean distribution. If W denotes the inverse of the basis matrix **S**, the coefficients matrix **U** for training matrix $X^T$ will be expressed in

$$U = WX^T \qquad (3.7)$$

The *n*-component base vectors which have the best distinguish ability for detecting humans and nonhumans should be chosen from many candidate components.

## 3.2.2 Entropy feature selection

Unlike PCA features, the ICA features are not sorted, thus the conditional entropy is applied to feature selection, the sorting process and choosing an appropriate subset of ICA features. Sorting variables may be an important step to enhance the high-dimensional dataset, which gave us the idea to place correlated or similar dimensions close to each other in high-dimensional value space to help human user perceive relationships among those variables easier.

If the entropy is the amount of information provided by a random variable, then our conditional entropy can be defined as the amount of information about one random variable provided another random variable. The entropy of a random variable reflects the more truthful information of the observed variable. If the variable is more random, it means unpredictable, which may result in the large entropy value.

The 2-D data space obtained from ICA feature extraction needs to be discretized into a matrix of a grid cell by separating each dimension into a set of intervals or bins. The discretization process begins with calculating the mean value of data in one dimension and dividing the data into two halves with that mean value. Recursively, each half is divided into halves with its own mean value. The recursion will stop when we obtain the required number of intervals or meet the constraint of total bins. Let a discrete random variable Z be with proposed values $\{z_1, z_2, ..., z_m\}$. The information entropy of *Z* with the probability density *p(z)* is defined in

$$H(Z) = -\sum_{i=1}^{n} p(z_i)\log p(z_i) \tag{3.8}$$

The conditional entropy quantifies the uncertainty of a random variable Y if given that the value of a second random variable *Z* is known. Each coefficient has to be normalized to [-1, 1] and quantized to n bins. Let Y= {-1, 1} be the desired class, the n the conditional entropy can be described in

$$H(Y \mid Z) = -\sum_{z}\sum_{y} p(y,z)\log p(y \mid z) = H(Y,Z) - H(Z) \tag{3.9}$$

The conditional entropy (*Y* | *Z*) is a weighted sum of the entropy values in all columns, where the joint entropy is defined by

$$H(Z,Y) = -\sum_{z}\sum_{y} p(z,y)\log p(z,y) \tag{3.10}$$

We sort the conditional entropy (*Y* | *Z*) and use the sorted results to select corresponding independent components. The coefficients or independent components with the better classification ability are associated with the small conditional entropy.

The selected ICA features will be used in the SVM classifier to identify humans to nonhumans. The training database consisted of 1843 human and 840 nonhuman images with image size (40x40) pixels. The maximum number if ICA to 76 and we choose 30 best bases for SVM classifier.

### 3.2.3 SVM classifier

Support Vector Machines (SVMS) are developed to solve the classification and regression problem. SVM is a way which starts with a linear separable problem. For classification, the goal of SVM is to separate two classes by a function which is induced available example. Consider the example in Fig. 3-6, there are two classes of data and many possible linear classifier that can separate these data, but only one of them is the best classifier which can maximize the distance between two class-margin, this linear classifier is called optimal separate hyperplane. Fig. 3-7 shows the accuracy rate and the number of SV (support vector) foe all number of ICA to be 76. Based on this experiment result, the accuracy rate of human detection system will increase if number of IC is increasing. But in practice, we select 30 ICs because number of SVs is almost close to minimum and the accuracy rate more than 90%. The main reason to choose minimum number of SVs is to decrease the computation cost, in order to meet the real-time requirement.

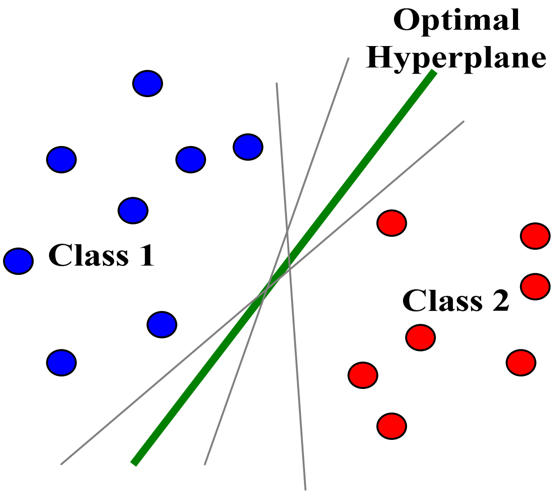
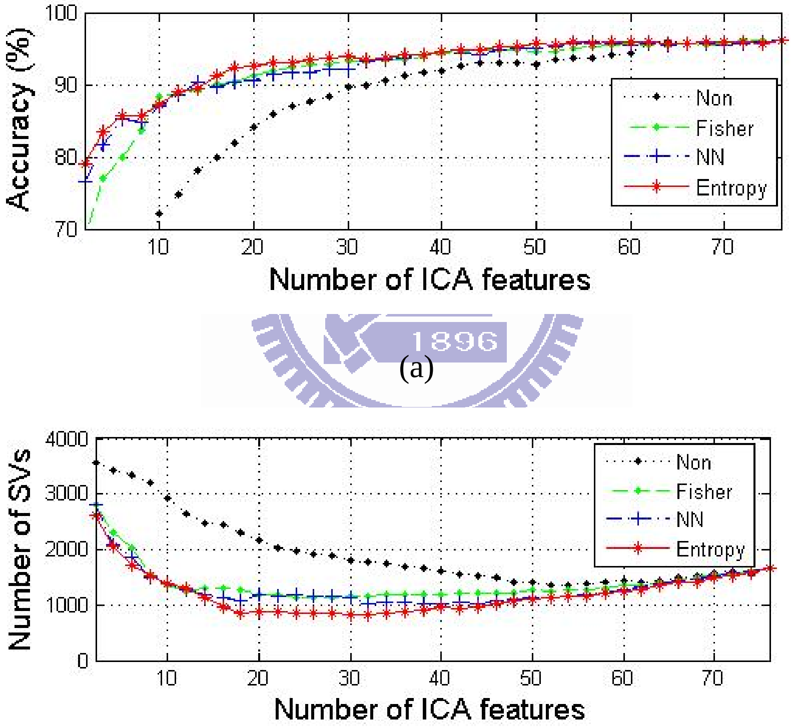


Fig. 3-6 Optimal separating hyperplane



(a)

(b)

Fig. 3-7 Analysis of feature selection. (a) Accuracy rate. (b) Number of SV.

# Chapter 4

## Human tracking

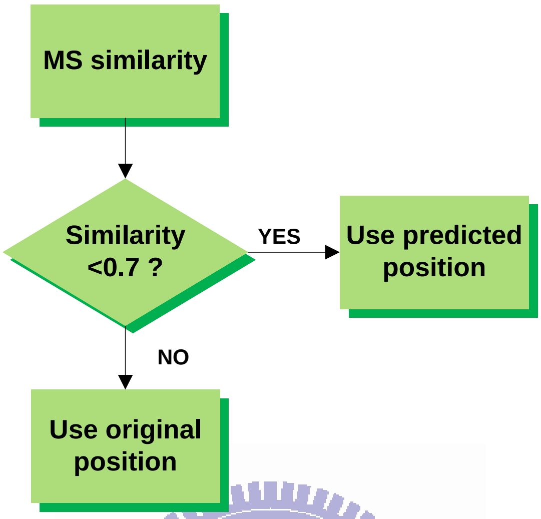


Fig. 4-1 Human tracking flow chart

A human tracking system based on mean-shift algorithm is proposed in this thesis. The mean-shift algorithm is a simple iterative procedure that shifts each data point in its neighborhood and locating the maxima of a density function given discrete data sampled from that function [36]. Generally, the mean-shift algorithm uses color feature but in our thesis, we combine color feature and ICs (Independent Components analysis) feature. The idea behind the combination is come from human detection based on ICs feature in chap3. Although color feature able to track moving object but the combination color and ICs able to track not only moving object but moving human.

The Kalman filter is applied to predict the location of moving human in next frame. The Kalman filter actives when moving human partially occluded with other object, in other hand the mean-shift similarity value decrease until smaller than a threshold value in Fig. 4-1.

# 4.1 features

## 4.1.1 color feature

Most kernel based object tracking use color as feature. Color information extends into three dimensions of original grey scale image so it will increase good tracking performance. Most papers use mean-shift as tracking algorithm usually consider color as feature to accomplish object lock. We compare the performance of three color space: RGB, HSV and Y'UV. Based on experiment, the HSV color space give a good performance while tracking, thus we apply it on tracking with active camera.

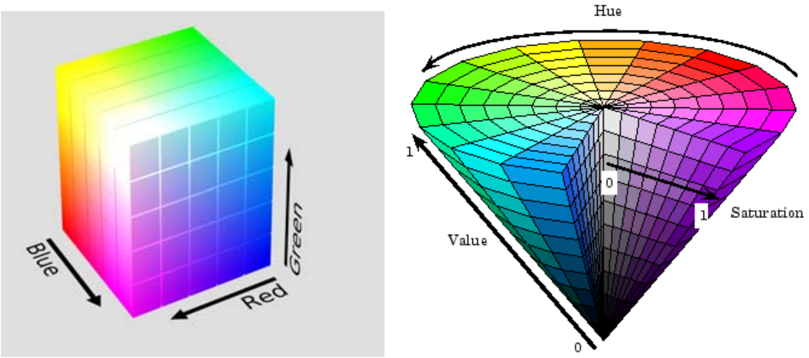
**HSV color transformation:**

The main purpose of HSV color space transformation is to reduce the sensitivity of illumination or lightness information in RGB color space. In Fig. 3-6, Fig. 3-7 are RGB and HSV color space, respectively.

The HSV model, also known as HSB model, was created in 1978 by Alvy Ray Smith. It is a nonlinear transformation of the RGB color space. It defines a color space in terms of three components: hue, saturation, and value. The definition is described below: [16]

1. Hue: It is the color type and ranges from 0 ~ 360 degree. Each value corresponds to one color. For example, 0 is red, 45 is orange and 55 yellow. When it comes to 360 degree, it is also equal to 0 degree.
2. Saturation: It is the intensity of the color, and ranges from 0%~100%. 0 means no color, and that means only gray value between black and white exists. 100 means the intense color with the most color variety.
3. Value: It is the brightness of the color, and also ranges from 0%~100%. 0 is

always black. Depending on the saturation, 100 may be white or a more or less saturated color.



(a) (b)

Fig.4-2 (a) RGB color model [17] (b)HSV color model [18]

The transformation algorithm from RGB color model to HSV color model could be described in the following equation.

$$H=\begin{cases} 0 & if \quad MAX=MIN \\ 60\times\dfrac{G-B}{MAX-MIN}+0 & if \quad MAX=R, G\geq B \\ 60\times\dfrac{G-B}{MAX-MIN}+360 & if \quad MAX=R, G<B \\ 60\times\dfrac{B-R}{MAX-MIN}+120 & if \quad MAX=G \\ 60\times\dfrac{R-G}{MAX-MIN}+240 & if \quad MAX=B \end{cases} \tag{4.1}$$

$$S=\begin{cases} 0 & if \quad MAX=0 \\ 1-\dfrac{MIN}{MAX} & \quad Otherwise \end{cases} \tag{4.2}$$

$$V=MAX \tag{4.3}$$

where MAX, MIN are maximum and minimum value of (R, G, B), respectively.

**Y’UV color transformation**

The Y’UV color model defines a color space in terms of one luma (Y') and two chrominance (UV) components. The Y'UV color model is used in the NTSC, PAL, and SECAM composite color video standards. Y' stands for the brightness component and U and V are the chrominance components. The transform equation are show following:

$$\begin{bmatrix} Y' \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.14713 & -0.28886 & 0.436 \\ 0.615 & -0.51499 & -0.10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4.4}$$

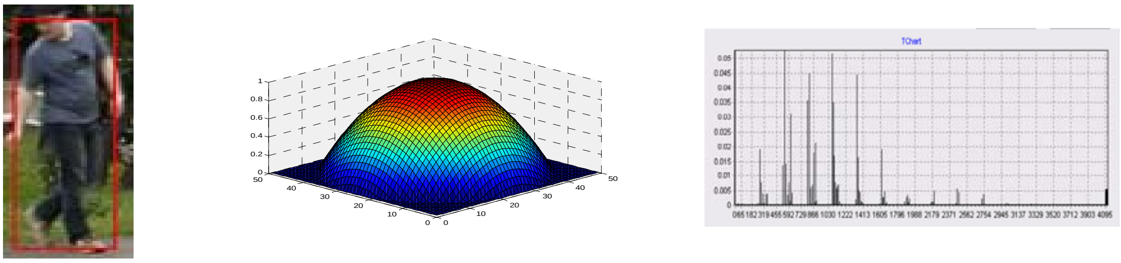
Each channel of color space has 8-bits data, means (255x255x255). We quantize the color space into (16x16x16). Therefore the histogram of color feature consists of 4096 bins.

### 4.1.2 ICA feature

In chapter 3, the human detection algorithm uses entropy feature selection to select 30 ICs from ICs. These 30 ICs are process by SVM classifier to classify whether the foreground object is human or nonhuman. If SVM classifier is classify the foreground object as human then these will use in mean-shift tracking algorithm.

In practice, the combination of color and ICs will construct a histogram with total bins equal to 4126. The combination procedure is described as follow:

1. Select the ROT (Region of interest), in this case is human's ROI.
2. Compute the color histogram by kernel function, as shown in Fig. 4-3 (b)
3. ICs extracted.
4. Combined the histogram of color and ICs features, as shown in Fig. 4-4



(a) (b) (c)

Fig. 4-3 (a) Human's ROI (b) Kernel function (c) Histogram of color feature

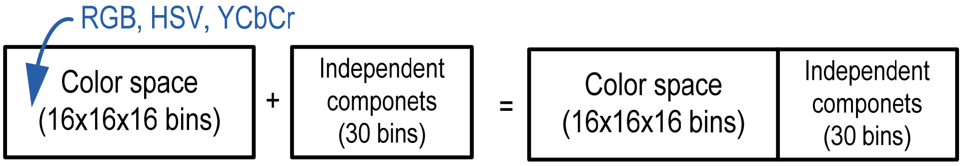


Fig. 4-4 Color and ICA feature histogram

## 4.2 Kernel functions

The feature histogram based target representations are regularized by spatial masking with an isotropic kernel. The masking induces spatial-smooth similarity functions suitable for gradient-based optimization, hence, the target localization problem can be formulated using the basin of attraction of the local maxima [19].

There are many kinds of kernel functions, such as Gaussian kernel, Flat kernel and Epanechnikov kernel. Let x be normalized pixel as location in the region defined as target model, then the Gaussian kernel, Flat kernel and Epanechnikov kernel [36] are defined as follows.

1. Gaussian kernel

$$k(x)=\frac{1}{2\pi}\exp(-\frac{1}{2}\|x^2\|) \tag{4.5}$$

2. Flat kernel

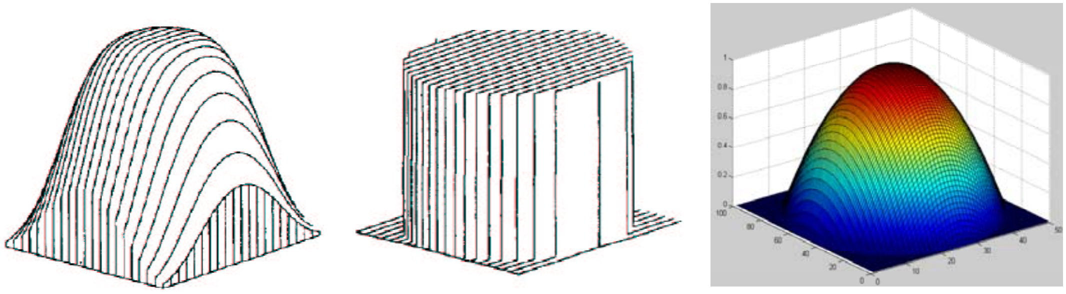
$$k(x)=\begin{cases}1 & if \quad \|x\|\leq 1\\ 0 & otherwise\end{cases} \tag{4.6}$$

3. Epanechnikov kernel

$$k(x)=\begin{cases}\frac{3}{4}(1-x^2) & if \quad \|x\|\leq 1\\ 0 & otherwise\end{cases} \tag{4.7}$$

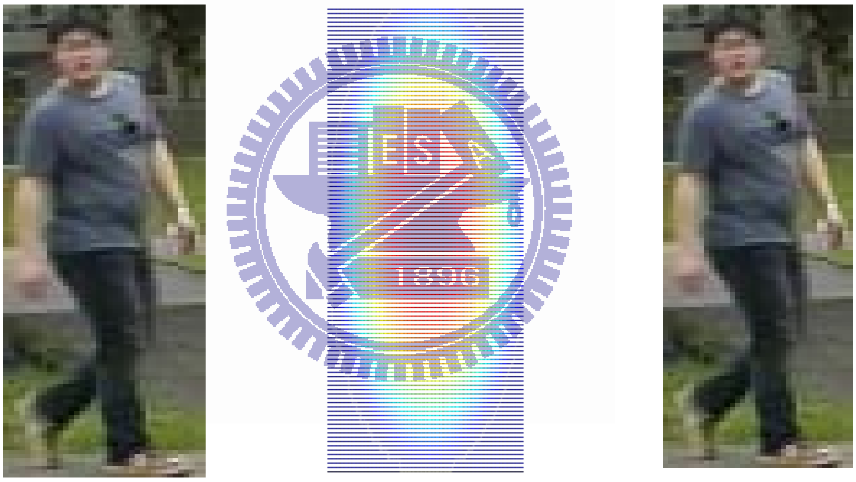
Fig. 4-5 (a) and (c) show that the Gaussian and Epanechnikov kernel are similar. They have highest value are the center distribution. If we take a looking the ROI of target model, the more closer to the center of ROI is containing more important pixels the background information mostly near at ROI boundary as shown in Fig. 4-6.

Therefore, Gaussian and Epanechnikov kernel can regardless the boundary information and the accuracy will larger than flat kernel.



(a) (b) (c)

Fig. 4-5 (a) Gaussian kernel (b) flat kernel (c) Epanechnikov kernel



(a) (b) (c)

Fig. 4-6 (a) Target object (b) Kernel function (c) Target object and Kernel function

## 4.3 Mean-shift algorithm

In order to characterize the target, first a feature space is chosen. The reference target model is represent by normalized histogram *q* in the feature space. The target model can be considered as centered at the spatial location 0. In the subsequent frame, the candidate model is defined at location y, be expressed as *p(y)*. We use Eq. 4.8 and Eq. 4.9 as our target and candidate model, respectively.

$$\hat{q} = \{\hat{q}_u\}_{u=1...m} \quad \sum_{u=1}^{m} \hat{q}_u = 1 \tag{4.8}$$

$$\hat{p}(y) = \{\hat{p}_u(y)\}_{u=1...m} \quad \sum_{u=1}^{m} \hat{p}_u = 1 \tag{4.9}$$

The similarity value between $\hat{p}$ and $\hat{q}$ is defined as Eq. 4.10

$$\rho[\hat{p}(y), \hat{q}(y)] \tag{4.10}$$

**Target model**

Let $\{x_i^*\}_{i=1...n}$ be normalized pixel locations in the region defined as the target model. The region is centered at 0. Here we use Epanechnikov kernel , using these weights increases the robustness of the density estimation since the peripheral pixels are the least reliable.

The function $b : R^2 \rightarrow \{1...m\}$ associates to the pixel at location $x^*$ the index b( $x^*$ ) of its bin in the quantized feature space. The probability of the feature u=1…m in the target model is then computed as

$$\hat{q}_u = C \sum_{i=1}^{n} k(\|x_i^*\|^2)\delta[b(x_i^*) - u] \tag{4.11}$$

where $\delta$ is the Kronecker delta function. The normalization constant C is derived by imposing the condition $\sum_{u=1}^{m} \hat{q}_u = 1$ from where

$$C = \frac{1}{\sum_{i=1}^{n} k(\|x_i^*\|^2)} \tag{4.12}$$

since the summation of delta functions for u=1…m is equal to one.

**Candidate model**

Let $\{x_i^*\}_{i=1...n}$ be the normalized pixel locations of the candidate model, center at y in the current frame. The normalization is succeed from the frame containing the target model. Here we use Epanechnikov kernel same as target model, but with bandwidth h, the probability of the feature u=1…m in the candidate model is given by

$$\hat{p}_u(y) = C_h \sum_{i=1}^{n_h} k(\left\| \frac{y-x_i}{h} \right\|^2)\delta[b(x_i)-u] \tag{4.13}$$

where

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k(\left\| \frac{y-x_i}{h} \right\|^2)} \tag{4.14}$$

is the normalization constant. Note the $C_h$ does not depend on y because the pixel locations $x_i$ are organized in a regular lattice and y is one of the lattice nodes. The bandwidth h defined the scale of the target candidate.

**Similarity measure**

The similarity function defines a distance among target model and candidate model. The Bhattacharyya coefficient, which evaluates the similarity of the target model and the candidate model, is defined as

$$\hat{\rho}(y) \equiv \rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y)\hat{q}_u} \tag{4. 15}$$

$$d(y) = \sqrt{1-\rho[\hat{p}(y), \hat{q}]} \tag{4.16}$$

To find the location corresponding to the target in the current frame, the Bhattacharyya coefficient in Eq. (4.16) should be maximized as function of y which can be solved by running the mean-shift iterations.

**Object localization**

Color and ICs information were chosen as the features, however, the same framework can be used for texture and edges, or any combination of them. In the sequential, it is assumed that the following information is available: a. d detection and localization of the objects track in the initial frame b. Every objects periodic analysis accounting for possible updates of the target models due to significant changes in color.

Minimize the distance *d(y)* is equivalent to maximizing the Bhattacharyya coefficient $\hat{p}(y)$. The search for the new target location in the current frame starts at the location $\hat{y}_0$ of the target in the previous frame. So the probabilities $\{\hat{p}_u(\hat{y}_0)\}_{u=1\ldots m}$ of the candidate at location $\hat{y}_0$ in the current frame have to be computed first. Using Taylor expansion around the values $\hat{p}_u(\hat{y}_0)$

$$\rho[\hat{p}(y),\hat{q}] = \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(\hat{y}_0)\hat{q}_u} + \frac{1}{2}\sum_{u=1}^{m}\hat{p}_u(\hat{y})\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \qquad (4.17)$$

In order to minimize the distance *d(y),* the second term in Eq. 4.17 has to be the maximized, the first term being independent of y. The second term represents the density estimate computed kernel function at y in the current frame, with the data being weighted by $w_i$ Eq. 4.18. In this process, the kernel is recursively moved from current location $\hat{y}_0$ to new location $\hat{y}_1$ according to the relation. The distance between $\hat{y}_1$ and $\hat{y}_0$ is mean-shift vector.

$$w_i = \sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}}\,\delta[b(x_i)-u] \qquad (4.18)$$

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)} \qquad (4.19)$$

where $g(x) = -k'(x)$.

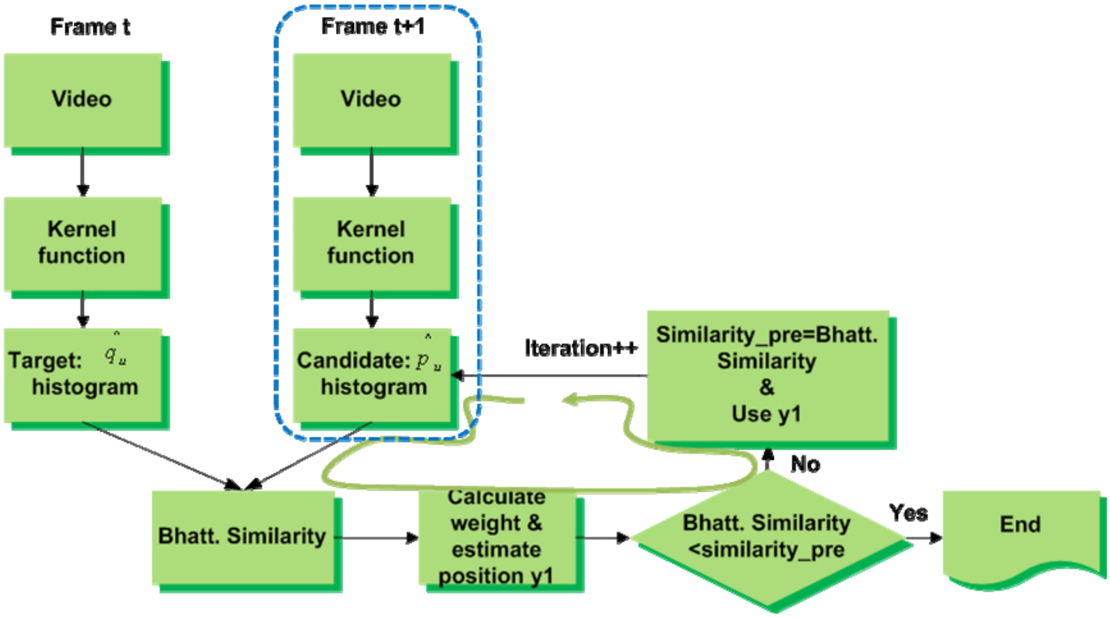

Frame t
Video
Kernel function
Target: $\hat{q}_u$ histogram
Frame t+1
Video
Kernel function
Candidate: $\hat{p}_u$ histogram
Iteration++
Similarity_pre=Bhatt. Similarity & Use y1
No
Bhatt. Similarity
Calculate weight & estimate position y1
Bhatt. Similarity <similarity_pre
Yes
End

Fig. 4-7 Mean-shift algorithm flow chart

Given the target model $\{\hat{q}_u\}_{u=1\ldots m}$ and its location $\hat{y}_0$ in the previous frame, set initial previous similarity value equal to 0, then the mean-shift algorithm is described as following,

1. Initialize the location of the target in the current frame with $\hat{y}_0$, compute $\{\hat{p}_u(\hat{y}_0)\}_{u=1\ldots m}$, and evaluate

$$\rho[\hat{p}(y_0),\hat{q}]=\sum_{u=1}^{m}\sqrt{\hat{p}_u(\hat{y}_0)\hat{q}_u}$$

2. Derive the weights $\{w_i\}_{i=1\ldots n_h}$ according to Eq. 4.18.

3. Find the next location of the candidate according to Eq. 4.19.

4. Compute $\{\hat{p}_u(\hat{y}_1)\}_{u=1\ldots m}$ ,and evaluate

$$\rho[\hat{p}(y_1),\hat{q}]=\sum_{u=1}^{m}\sqrt{\hat{p}_u(\hat{y}_1)\hat{q}_u}$$

5. If similarity_previous <= $\rho[\hat{p}(y_0),\hat{q}]$

Do $\hat{y}_1 \leftarrow \frac{1}{2}(\hat{y}_1+\hat{y}_0)$ and similarity_previous = $\rho[\hat{p}(y_0),\hat{q}]$

Evaluate $\rho[\hat{p}(y_1),\hat{q}]$ and go to step 2.

Otherwise $\hat{y}_0 \leftarrow \hat{y}_1$ and break.

## 4.4 ROI resizing

In real situation, human probably walk toward or keep away from camera, thus fixed ROI’s scale is not suitable because the ROI will contain some background pixels or only some parts of human. Consequently, it will influence tracking result shown in Fig. 4-8 and a similarity value smaller than a threshold value. Therefore, the ability to resize ROI’s scale is an important issue. In our system the ROI scale is adjusting every 100 frames.

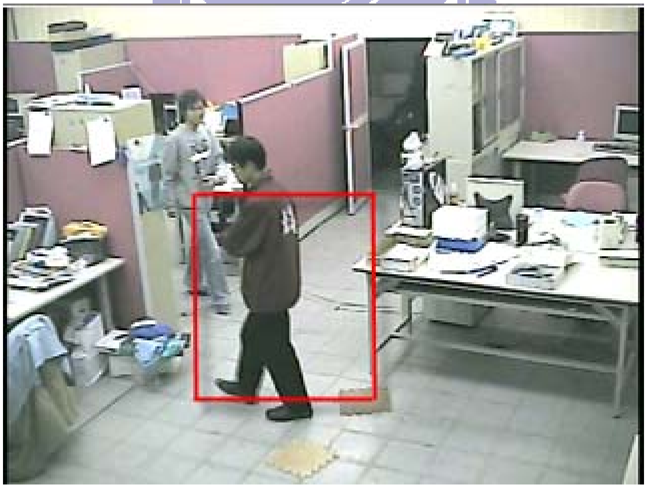


Fig. 4-8 ROI scale larger than object

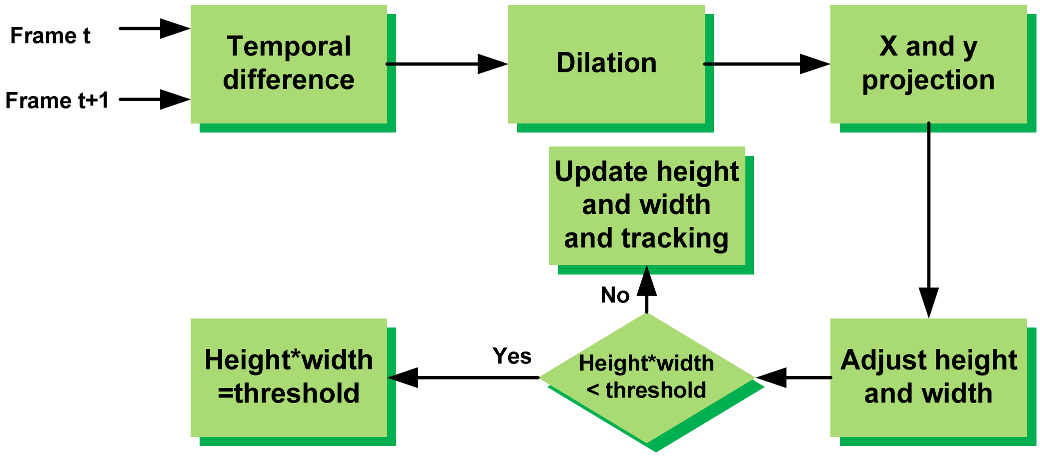


Fig. 4-9 ROI resize flow chart

In order to adjust ROI scale adaptively, first we use temporal difference to find the foreground image and its position we do the temporal difference only in ROI's regions. The dilation process is applied to gradually enlarge the boundaries of for ground pixel and link the broken boundary parts which obtain temporal difference. The projection of dilation image into x-axis and y-axis will produce the current width ($width_{current}$) and height ($height_{current}$) shown in Fig 4-10. Finally, the new ROI's size is determined with Eq. 4.20. Sometimes, the dilation process unable to link broken boundaries perfectly, thus the projection toward x-axis and y-axis will produce ROI smaller than actual human region. So we will set the minimum size of ROI. After the new ROI's scale is determined then the target model will be update, too. The update method is use histogram of $\hat{q}_u$ shown in Eq. 4.21, where $\alpha$ is set to 0.6.

$$\begin{cases} height_{new} = (height_{previous} + height_{current})/2 \\ width_{new} = (width_{previous} + width_{current})/2 \end{cases} \tag{4.20}$$

$$t\arg et\_histogram = (1-\alpha)*cansidate\_histogram + \alpha * t\arg et\_histogram \tag{4.21}$$

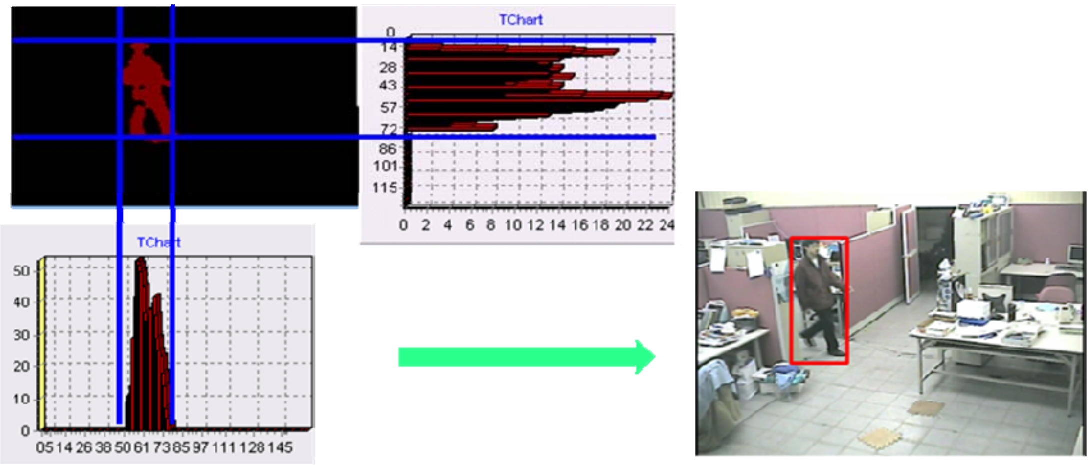
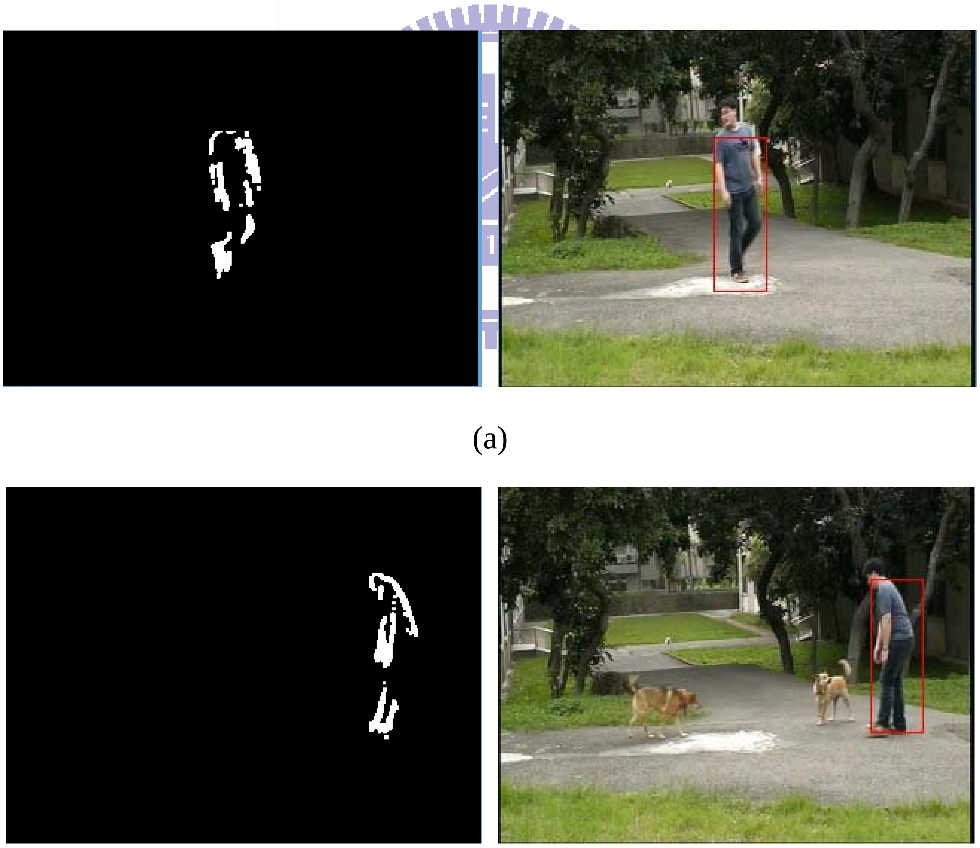


Fig. 4-10 X-axis and Y-axis image projection

In Fig. 4-11 and 4-12 show the ROI resize in fixed and active camera, respectively.

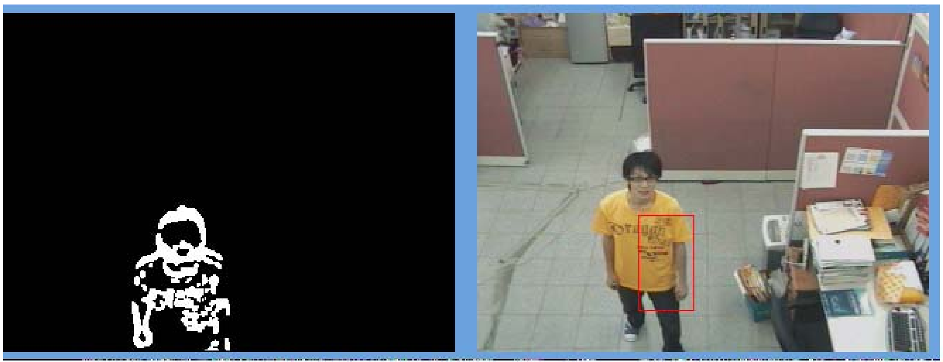


(a)

(b)

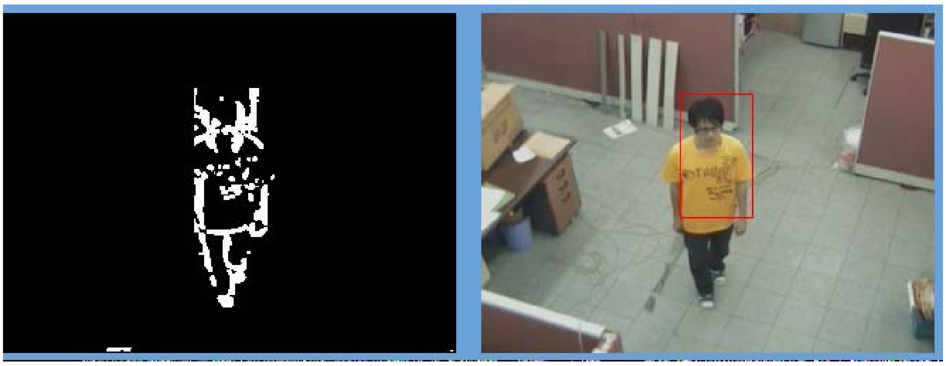
Fig. 4-11 Fixed camera condition (a) and (b) left is difference image. Right is after resize scale.

(a)

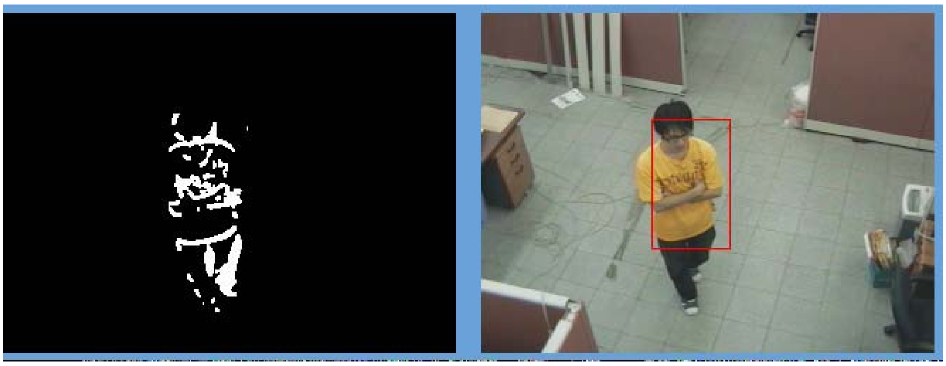


(b)

(c)



(d)



(e)

Fig. 4-12 Active camera condition. (a) original image (c-e) left are difference images. Right is ROI resizing scale image

## 4.5 Kalman filter

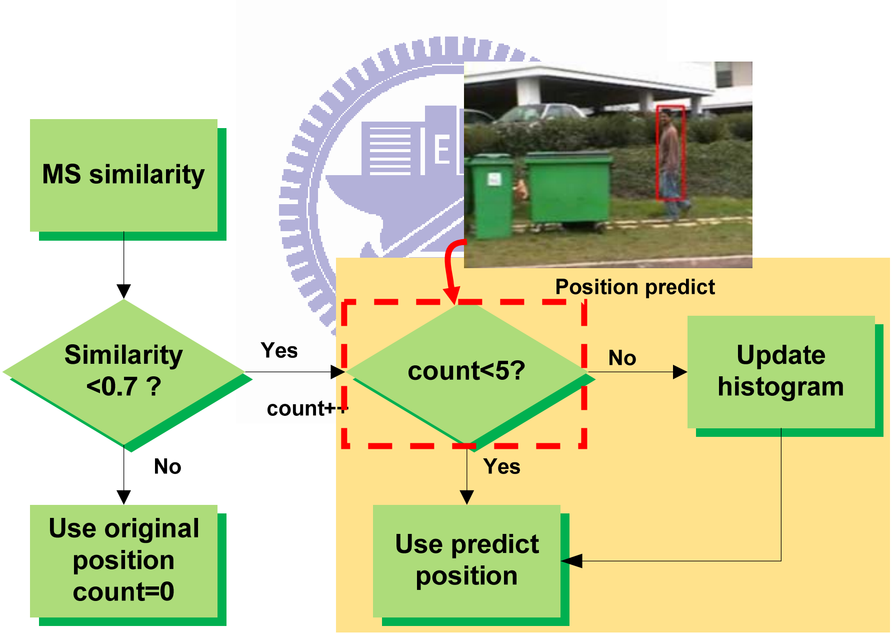


Fig. 4-13 Kalman filter flow chart

Fig. 4-13 shows the Kalman filter algorithm with use to predict the target location when the target is occluded with other object or if Bhattacharyya similarity value smaller than a threshold value (in this case we use threshold value equal to 0.65). if the Count smaller than 5, then it means object was occluded and the predicted position will be used while tracking. If the predicted position used in continuous 5 frames the target model histogram will be updated by Eq. 4.21.

In our system, the Kalman filter is integrated into mean-shift object tracking method. First, Kalman filter initialized by mean-shift target position. Second, the searching result of mean-shift is feedback as the measurement of Kalman filter and estimating its parameters.

$$\begin{cases} X_k = AX_{k-1} + W_k & (a) \\ Z_k = HX_k(t) + V_k(t) & (b) \end{cases} \qquad (4.22)$$

We assume that $W_k$ and $V_k$ are Gaussian random variable with zero mean, so their probability density function are *N*[0,**Q**(t-1)] and *N*[0,**R**(t)], where the covariance matrix **Q**(t-1) and **R**(t) are referred to as the transition noise covariance matrix and measurement noise covariance matrix. Here $X_k = [x, y, Vx, Vy]^T$ is state of the system at the moment k, $Z_k = [x, y]^T$ is measurement value of system state at the moment k. $x$ and $V_x$ are the horizontal position and velocity respectively. The value of state transition matrix A, measurement matrix H, process noise covariance matrix Q and measurement noise covariance matrix R list as following Eq. 4.23,

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad (4.23)$$

The detail procedures of Kalman position prediction are listed bellow:

1. Predict the position of the target at moment k by kalman filter, and compute the prior error estimate covariance.

$$\hat{x}_k^{'} = A\hat{x}_{k-1}^{'} + w_k \qquad \text{(a)}$$

$$P_k^{'} = AP_{k-1}A^T + Q \qquad \text{(b)} \qquad (4.24)$$

2. Centered with predicted position $\hat{x}_k'$ , acquire the observation value $Z_k$ according to Eq. 4.22(b).
3. Correct measurement with Kalman filter, compute the revision matrix and renew poster state estimation as well as posterior error estimate covariance.

$$K_k = P_k' H^T (HP_k' H^T + R)^{-1} \quad \text{(a)}$$

$$\hat{x}_k = \hat{x}_k + K_k (Z_k - H\,\hat{x}_k') \quad \text{(b)}$$

$$P_k = (1 - K_k H) P_k' \quad \text{(c)} \qquad (4.25)$$

Here *Vx* and *Vy* are x and y motions, respectively. In most application, we usually consider current and previous frame motion. If the moving object move in the same direction, using two frames motion to predict new position will obtain small error with respect to actual position. If the moving object moves in different direction use two frame motion are not good enough to represent because the predicted position. Here we consider more frames motion to get more accuracy, therefore we choose 5 frames motion's average to get the more represent of moving direction.

$$\hat{V}_x = \alpha_x \sum_{i=n+1}^{n+f} (x_i - x_{i-1}) / f$$

$$\hat{V}_y = \alpha_y \sum_{i=n+1} (y_i - y_{i-1}) / f \qquad (4.26)$$

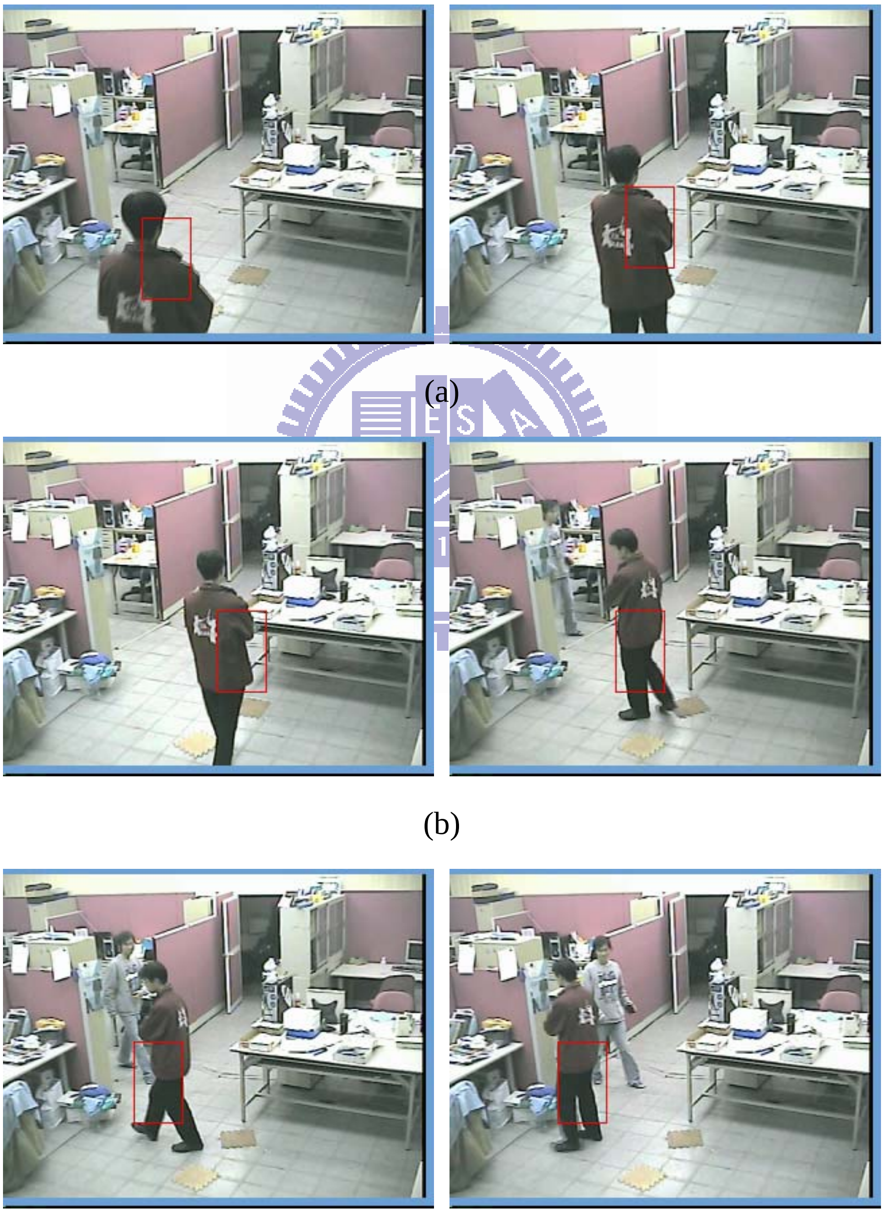
In both formulas, *xi* and *yi* are the horizontal and vertical coordinate of the target center respectively, $\alpha_x$ and $\alpha_y$ are proportional coefficients, f is the number of continuous frames. The distance between Kalman and mean-shift position are calculated by Euclidean distance.

In this section we use the position predicted by kalman filter compare with mean-shift. This comparison is helpful us to know the accuracy of the predicted position. We use Eq. 4-27 to calculate the error between mean-shift and Kalman filter.

$$MAE = \frac{\sum_{i=1}^{n} | x_i - \hat{x}_i |}{n} \tag{4.27}$$

Where xi is mean-shift position and x^ is predicted position by Kalman filter.

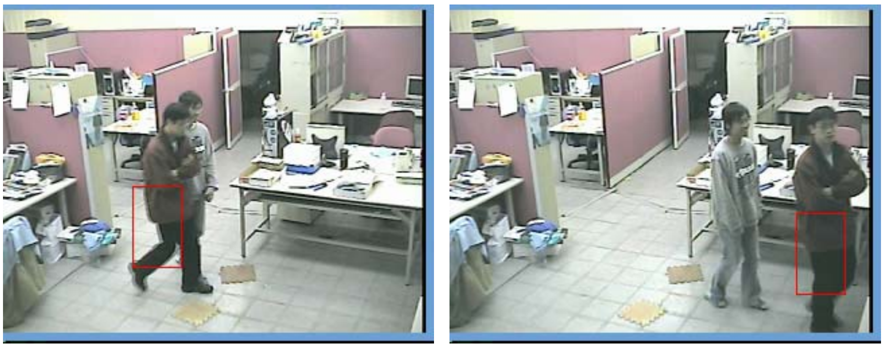
We do experiment both in indoor and outdoor environments. In the indoor environment, the human is not walking with a certain direction or path. Meanwhile, in the outdoor environment, the human is walking with the same direction.



(a)

(b)

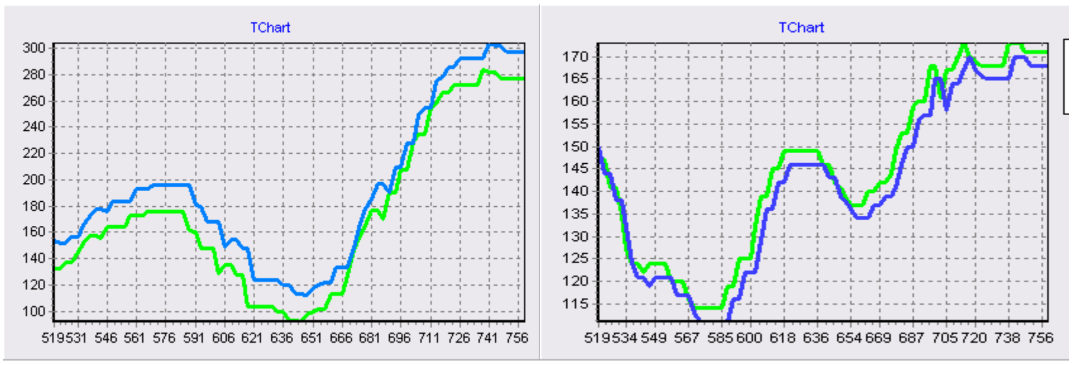
(c)



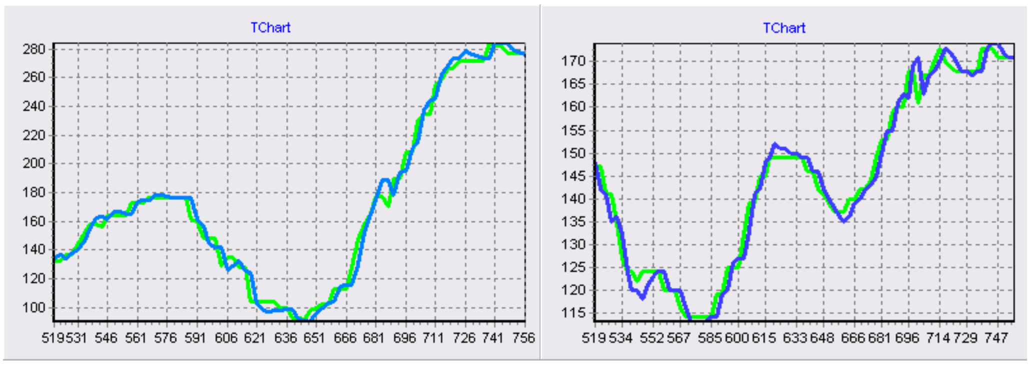
(d)

Fig. 4-14 Indoor environment (a) frame 643 and 662 (b) frame 679 and 710 (c) frame 718 and 728 (d) frame 761 and 797

The MAE is calculated separately in x-direction and y-direction Fig.4-15 and Fig. 4-17 show the curves of x and y-position, where green line and blue line indicate mean-shift and Kalman filter prediction, respectively. The MAE position analysis uses 2 and 5-previous frames motion. Table 4-1 shows the MAE of 5-previous frame motion is smaller than 2-previous frame motion. Meanwhile, Table 4-2 shows that the MAE of 2 and 5-previous frame motion do not differ greatly. The reason is the MAE in Table 4-1 are generated from human that walking in several directions, thus using as much as possible frame to compute Kalman prediction will produce position better than 2-previous frame. But in the case of Table 4-2, the human is walking with the same direction , thus 2-previous frame enough to represent the Kalman prediction.



(a)

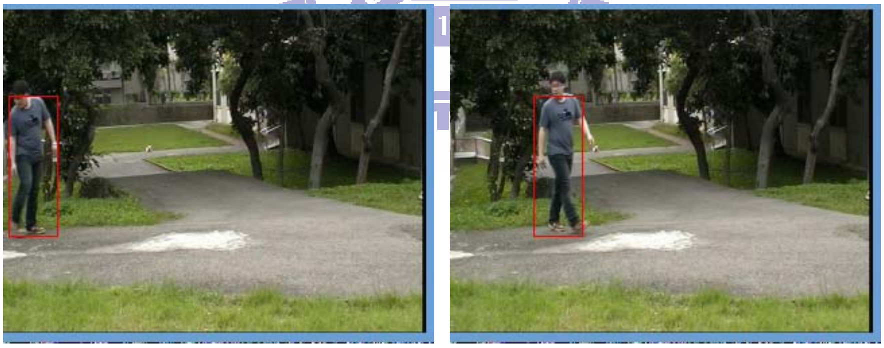


(b)

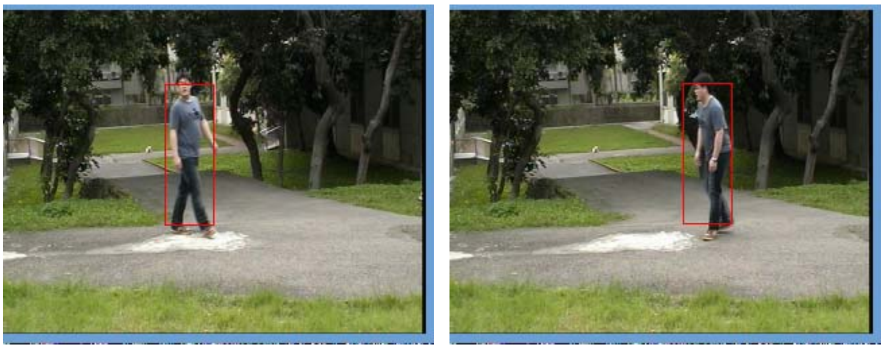
Fig. 4-15 (a) previous 2 frame motion left is x and right y position (b) previous 5 frame motion left is x and right y position

Table 4-1 MAE in different frame motion

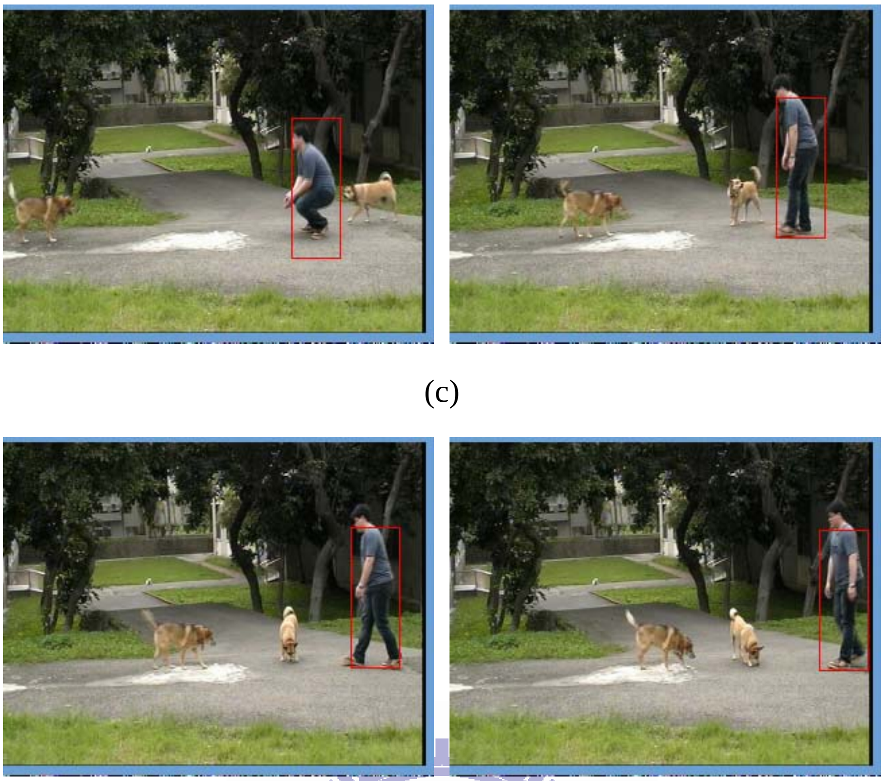
| | X position error (pixel) | Y position error (pixel) | distance |
|---|---|---|---|
| Previous 2 frame motion | 18.25 | 3.74 | 18.9 |
| Previous 5 frame motion | 5.02 | 2.15 | 6.02 |



(a)



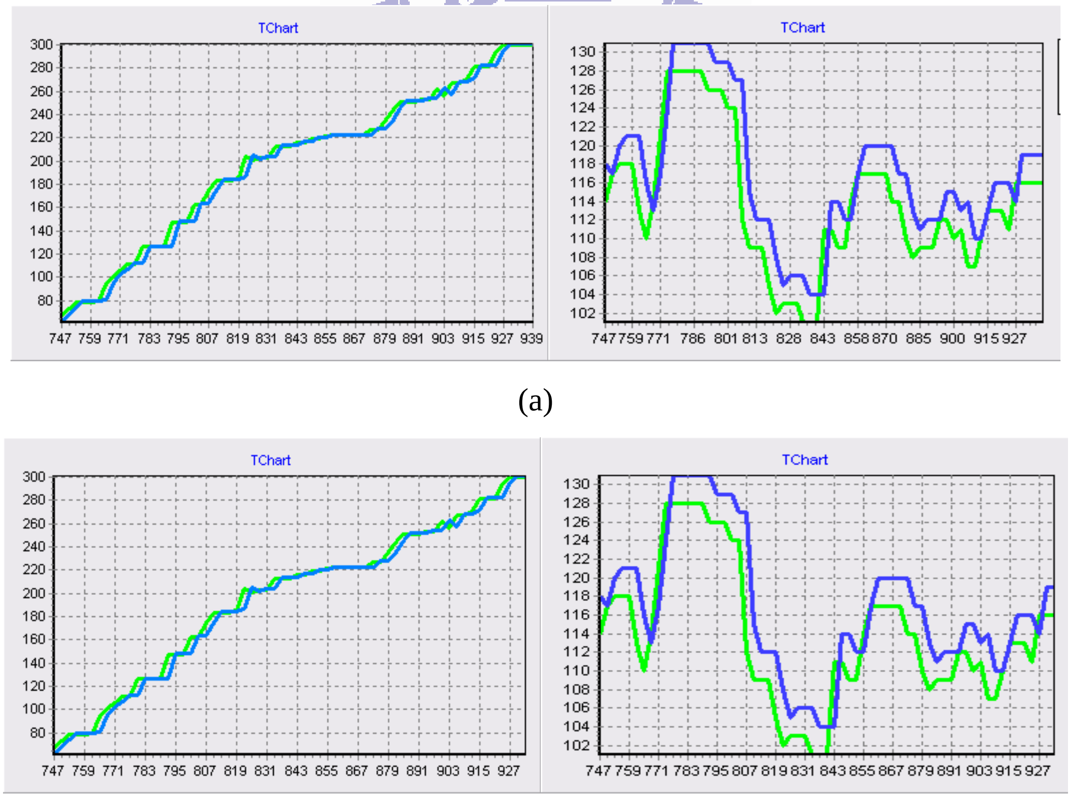
(b)



(c)

(d)

Fig. 4-16 Outdoor environment (a) frame 708 and 738 (b) frame 762 and 783 (c)frame 812 and 837 (d) frame 847 and 857



(a)

(b)

Fig. 4-17 (a) previous 2 frame motion left is x and right y position (b) previous 5 frame motion left is x and right y position

Table 4-2 MAE in different frame motion

| | X position error (pixel) | Y position error (pixel) | distance |
|---|---|---|---|
| Previous 2 frame motion | 3.98 | 3.6 | 5.89 |
| Previous 5 frame motion | 4.07 | 3.619 | 5.49 |

# Chapter 5

# Experimental results

In this chapter, we will reveal the human detection and tracking system on active camera. Our algorithm was implemented on the platform of PC with Intel Core2 Quad 2.4GHz and 2GB RAM. Our algorithm was developed in Borland C++ Builder 6.0 on Window XP. Because our human detection and tracking system will run in real time video surveillance with an active pan-tilt-zoom camera, we should do some experiments to test its performance and stability under several kinds of environments.

In section 5.1, introduce the experimental environment. In section 5.2, we will experiment our kalman position predict and compare with mean-shift position and calculate their error by MAE (Mean absolute error). The kalman filter used in real-time situation to solve object occlusion problem. In section 5.3, we use experiment single object in the scenario. In section 5.4, we use experiment multiple objects in the scenario.

## 5.1 Environment setup

The environment of experimental locates in our laboratory. The complexity of the environment is enough to verify our system while tracking and detecting moving human. Fig. 5-1 show several images of our laboratory environment without zoom in/out operation. Fig. 5-2 shows several images for zoom in/out condition.

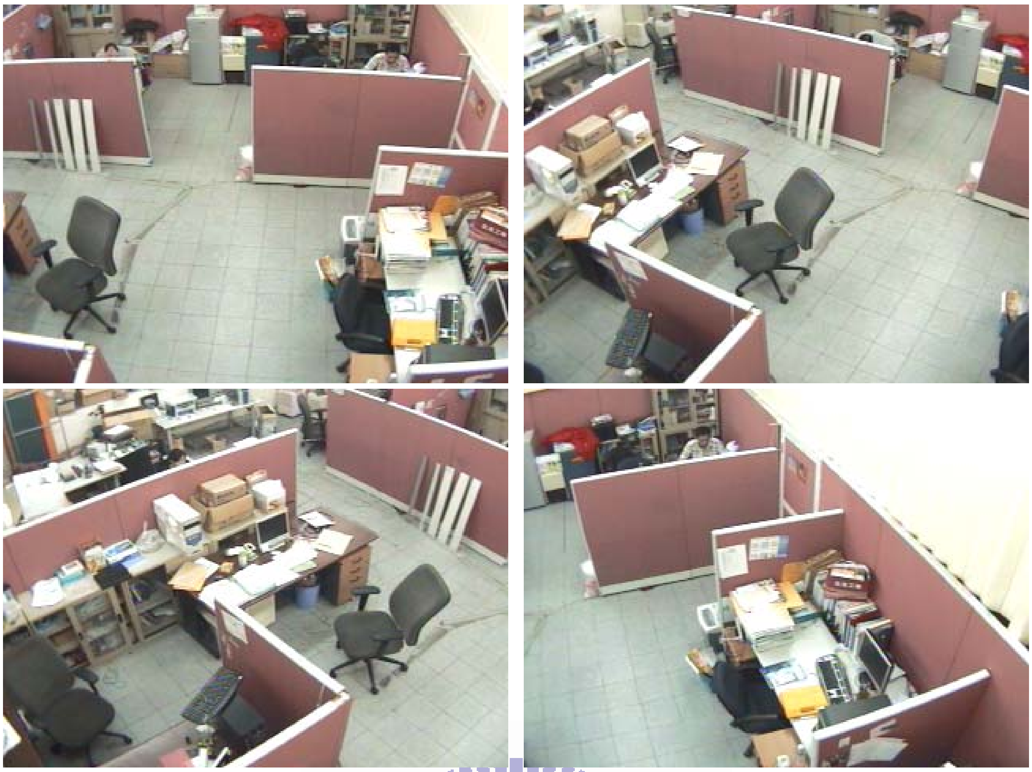


Fig. 5-1 Experimental environment

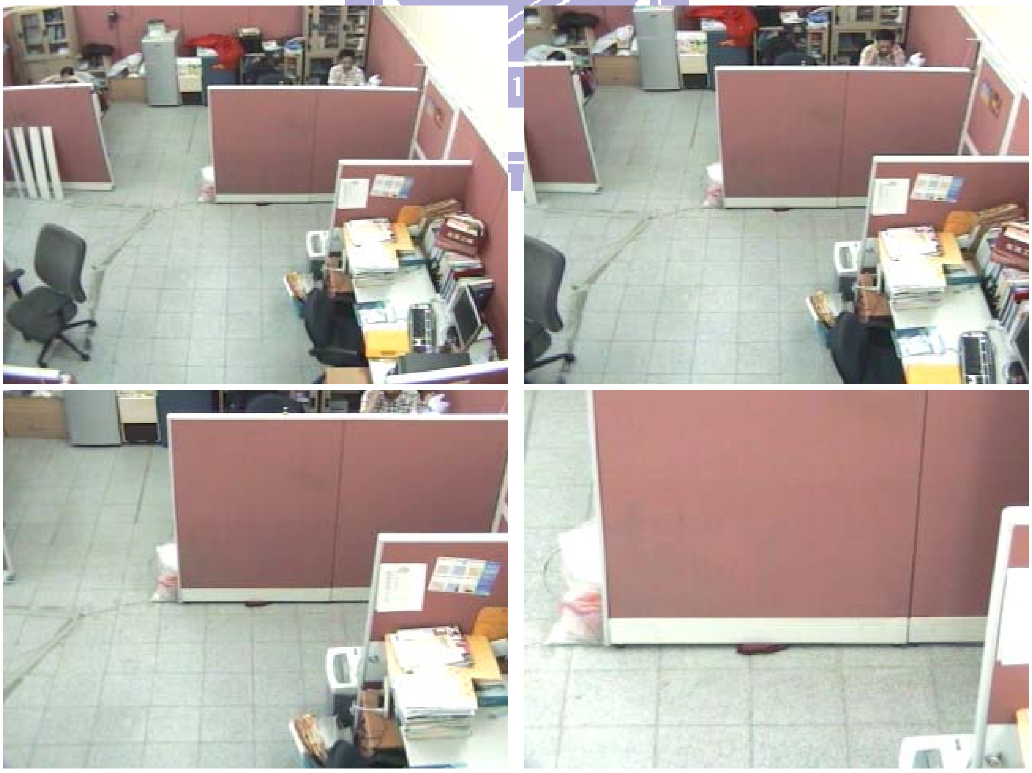


Fig. 5-2 Experiment zoom condition

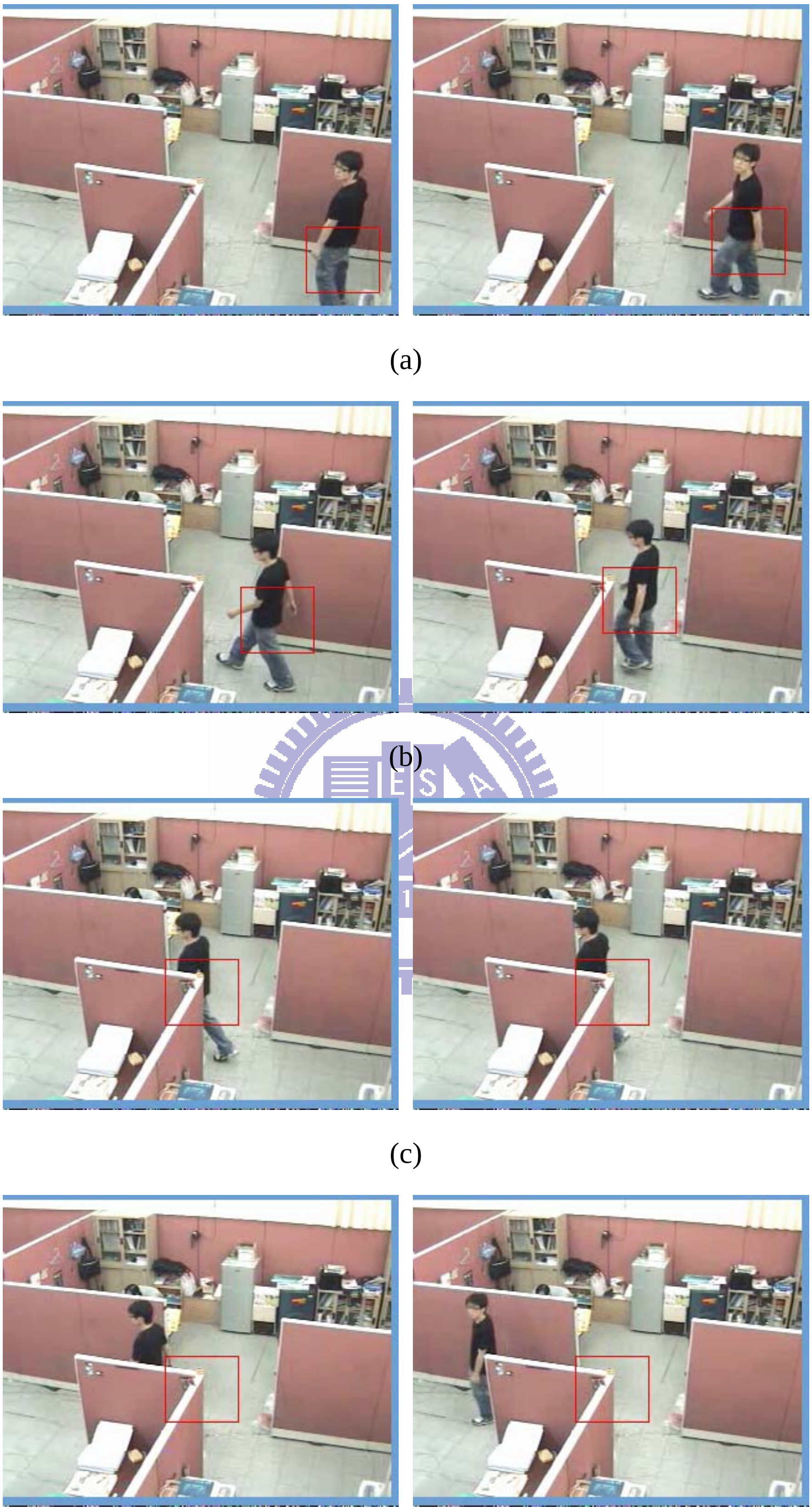
## 5.2 Kalman filter

The following two figures (Fig. 5-3 and Fig. 5-4) are compared under occlusion situation the difference between Kalman filter and no Kalman filter. In Fig. 5-3 We can observe a human pass the red screen will result the Bhattacharyya coefficient small than threshold so in frame 227 the mean-shift tracking will miss lock the object under occlusion problem. Fig. 5-4 Kalman filter was embedded in mean-shift algorithm the occlusion problem can be solve in frame 227 as shown in Table 5-1. We can also observe the object occluded for long time but the human was still be locked. Because the target object histogram be update so that kalman filter is not always be used in occlusion situation. Using histogram update idea in occlusion will increase tracking accuracy and precision.

There is one occlusion testing case in indoor environment. A man walking and then be occluded by a tall and long screen. In this situation the similarity measure will be drastically low so that the Kalman filter will be used to predict position.

Table 5-1 Predicted position

| Frame number | Similarity | Object center |
|---|---|---|
| 225 | 0.6874 | (167,153) |
| 226 | 0.6366 | (165,153) |
| 227 | 0.5933 | (176,143) |
| 259 | 0.6937 | (112,118) |



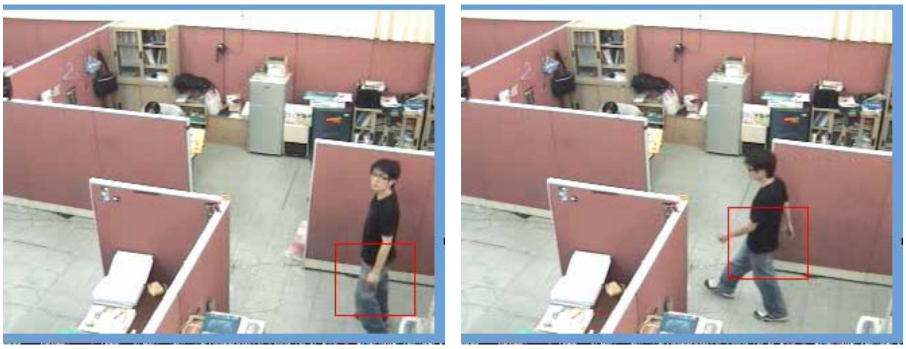
(a)

(b)

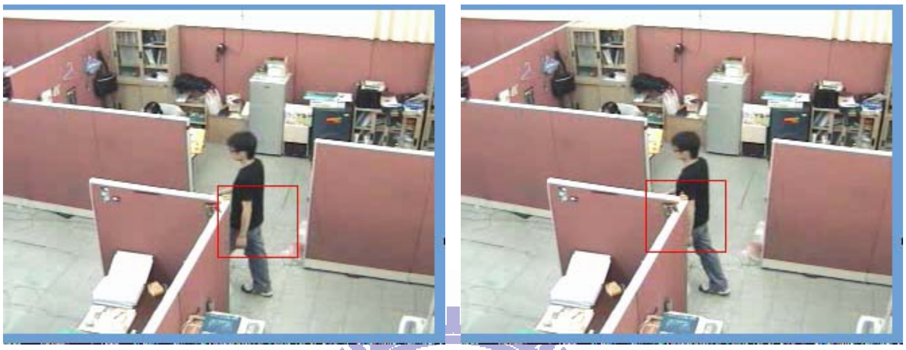
(c)

(d)

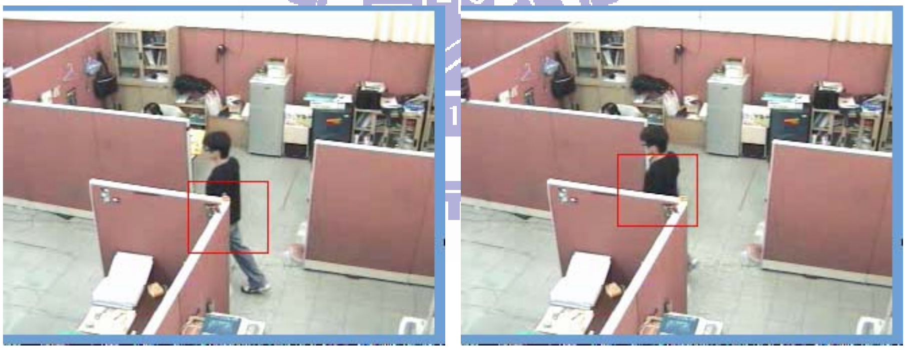
Fig. 5-3 Human tracking use mean-shift. (a) frame 194 and 206 (b) frame 225 and 234 (c) frame 243 and 246 (d) frame 255 and 277



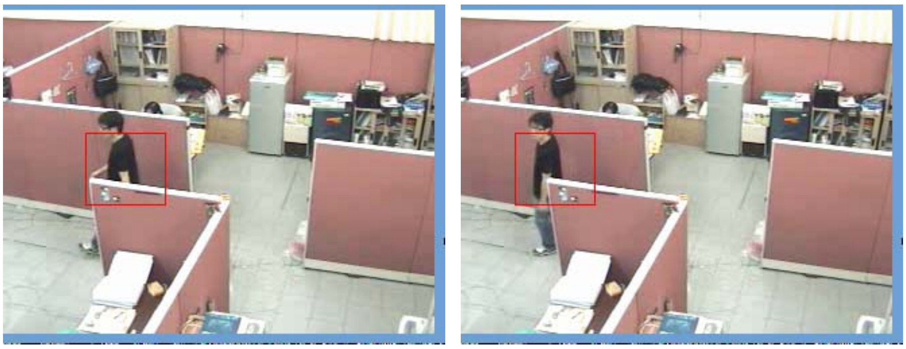
(a)



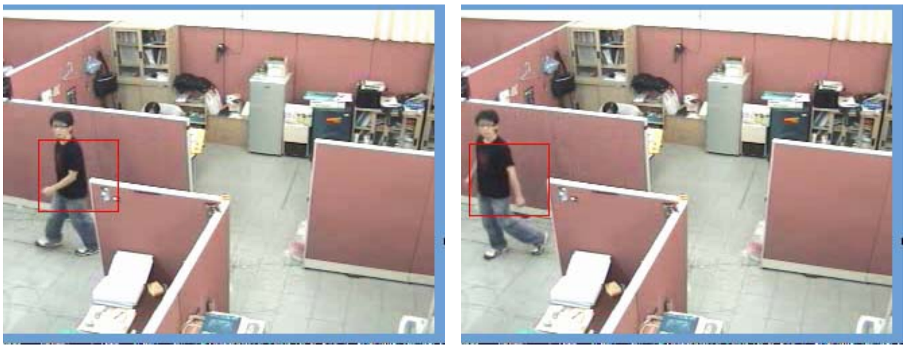
(b)



(c)



(d)



(e)

Fig. 5-4 Human tracking use mean-shift and Kalman filter. (a) Frame 186 and 210 (b)Frame 224 and 227 (c) Frame 229 and 233 (d) Frame 255 and 263 (e)Frame 269 and 278

## 5.3 Active camera with single object experiment

In this section we experiment a single objects move in the scenario that active camera will smoothly trace the object. So there 5 topics to discuss one object in various situation that have different performance.

### 5.3.1 9 regions and 26 regions experiments

There are many methods to control the active camera direction, for example motion or position … etc. In this thesis we use position based method to control the camera pan/tilt directions. The image size is 320x240 so we divide into 9 and 25 regions. Every region implies a direction and speed so that the object in one of these regions the algorithm sends command to active camera tell it to pan/tilt.

In 9 regions, the speed of each regions have fixed speed and the speed was be stopped by stop command that means the pan/tilt angle was limited by stop command so in visual situation we can observe the camera stop and go repeat forever shown in Fig. 5-5. The stop and go phenomenon that result in observer uncomfortable. The 9 regions direction are show in Fig. 5-6. In real situation the object will be missed

In order to solve stop and go phenomenon, the regions of image divides into 25 regions shown in chap2 Fig. 2-5. Each regions have different speed and the stop command will not be used anymore so the we not limit the angle of pan/tilt. Thus stop and go problem will be solved show in Fig. 5-7.

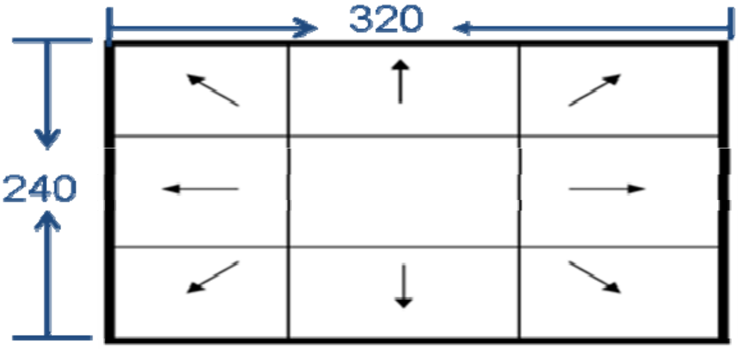
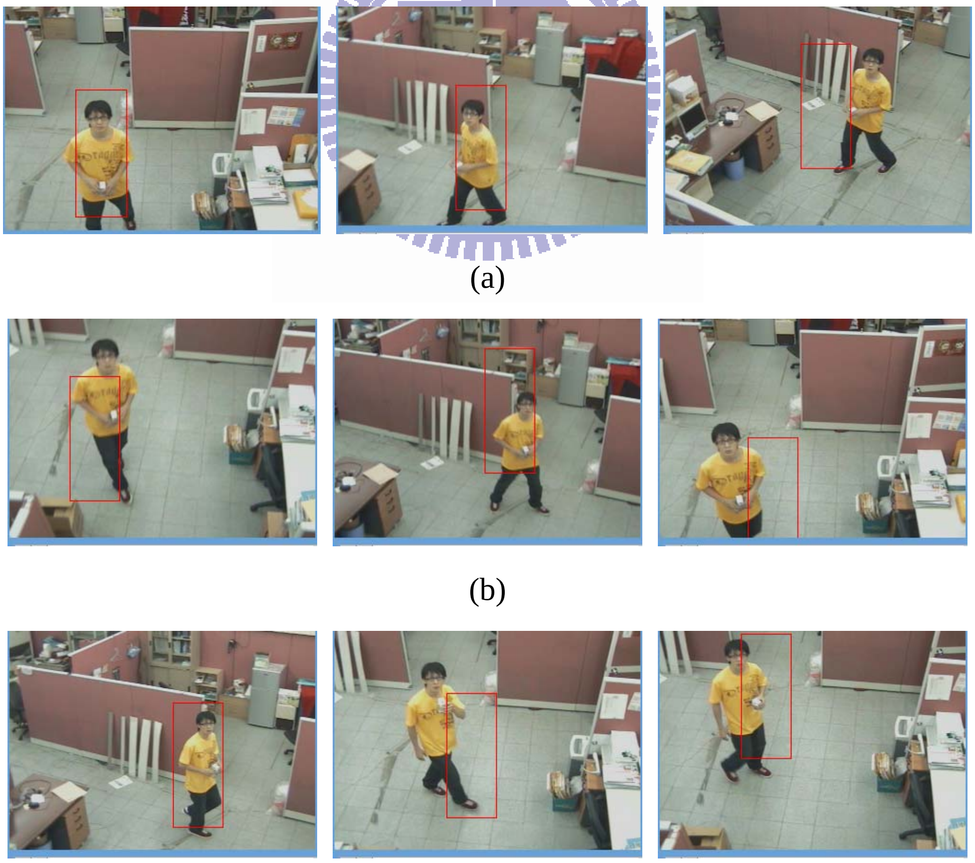


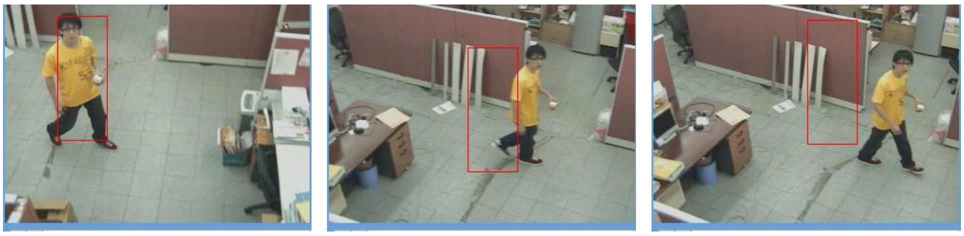
Fig. 5-5 9 regions pan/tilt control



(a)

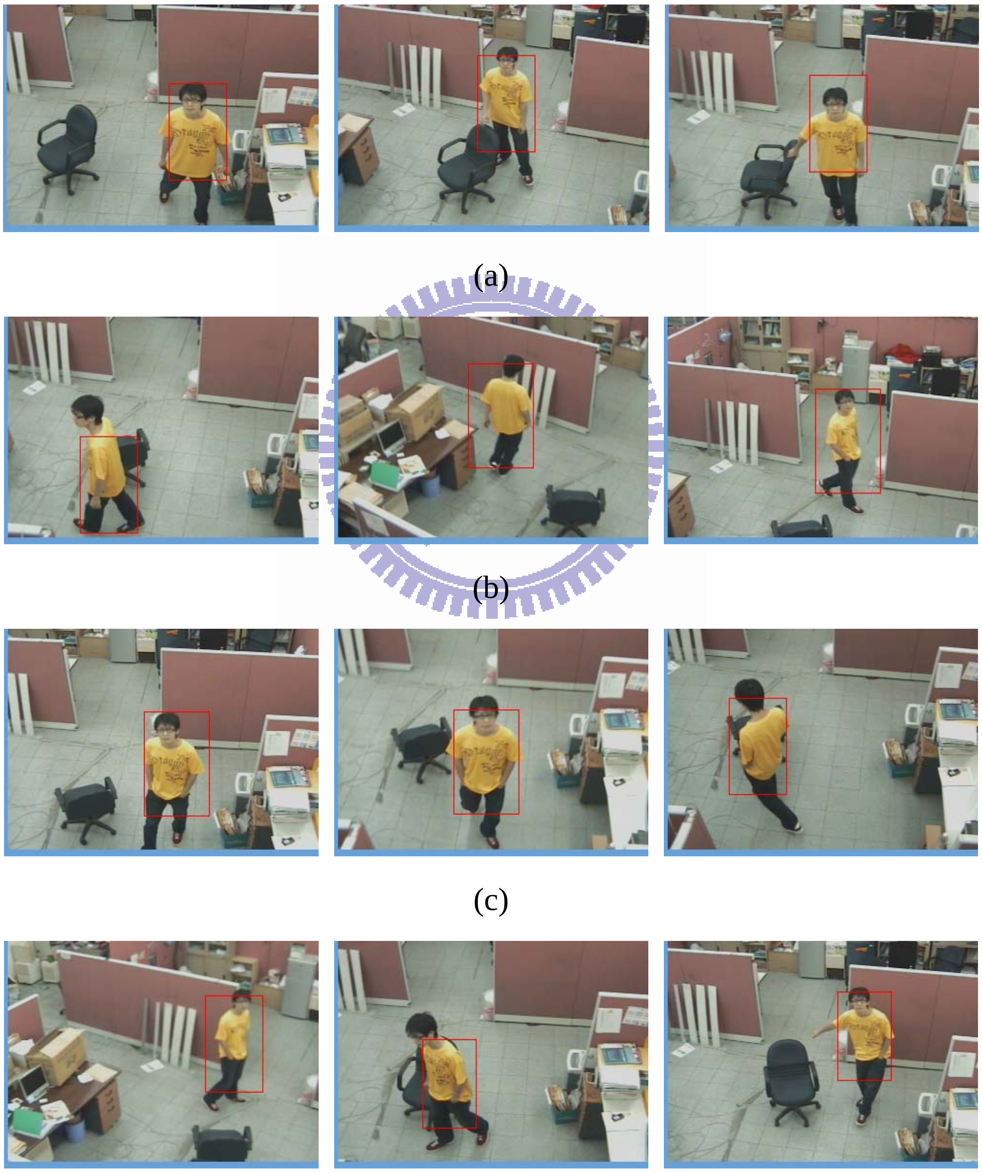
(b)

(c)



(d)

Fig. 5-6 Position based use 9 regions (a)frame 180, 244 and 272 (b)frame 345, 383 and 445 (c)frame 510, 560 and 564 (d)frame 567, 595 and 601

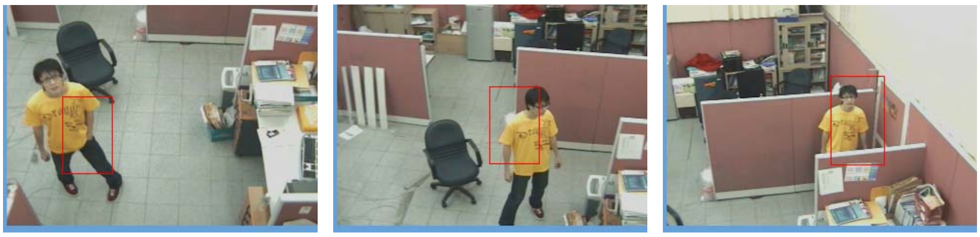


(a)

(b)

(c)

(d)



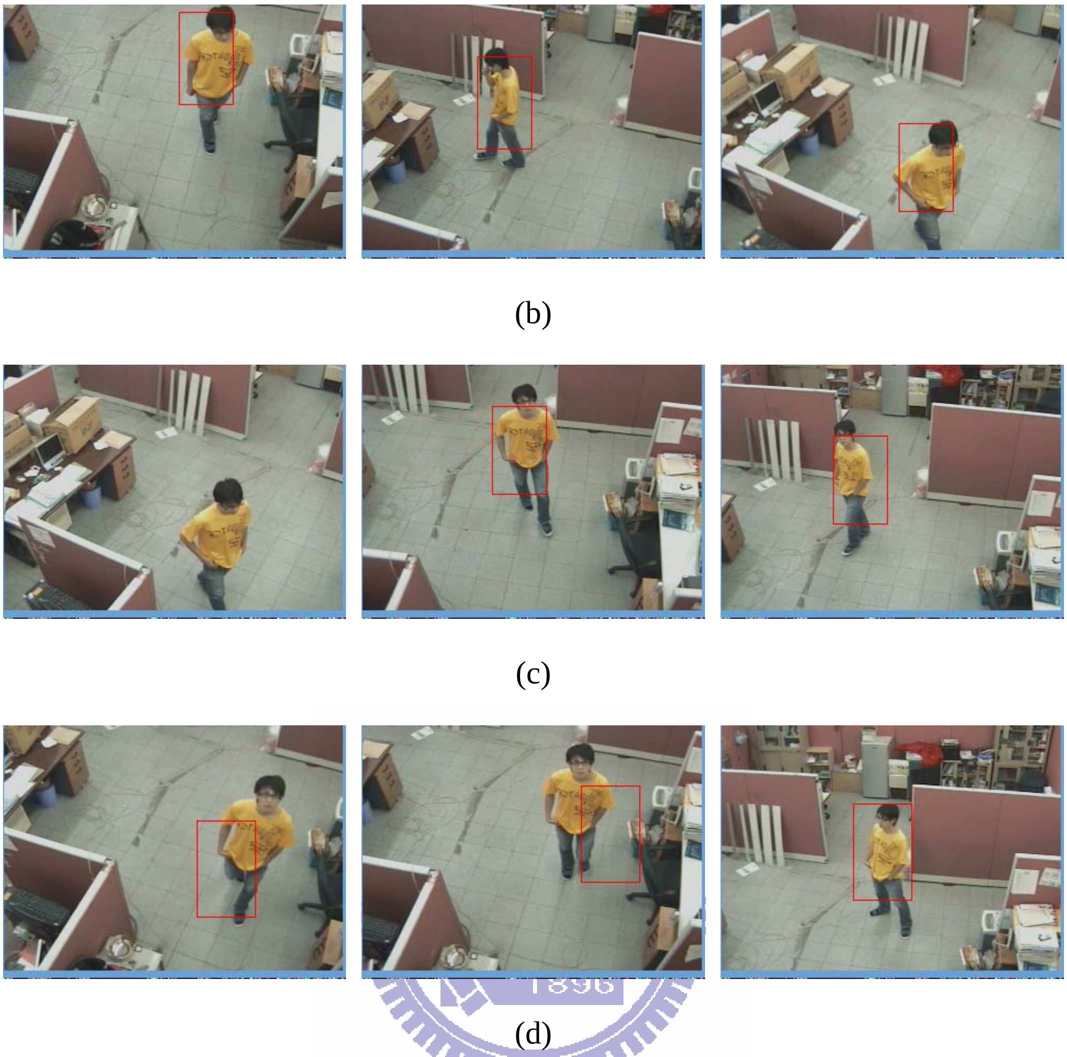
(e)

Fig. 5-7 Position based use 25 regions (a)frame 127, 189 and 213 (b)frame 241, 280 and 232 (c)frame 366, 385 and 418 (d)frame 473, 565 and 612 (e)frame 693, 775 and 843

## 5.3.2 Color spaces experiments

In section 5.2.1, there are two samples use HSV as color feature. We can observe this color space can focus on object shown in Fig. 5-7. So our experiment HSV is the main color space used on active camera. Because object walking in the scenario the lightness will result in object occurring essence change. In case of brighten occurred the object essence become different than original. Y’UV and RGB color space used to compare with HSV. In Fig. 5-8 is RGB color space used in our algorithm. In these figures we can track the object in the scenario smoothly, but the ROI position not always focus on object’s body. Sometimes it focuses on floor that the similarity values drastically down. The phenomenon in RGB color space has more sensitivity to lightness. In Fig. 5-9 the Y’UV color space is better than RGB.

(a)

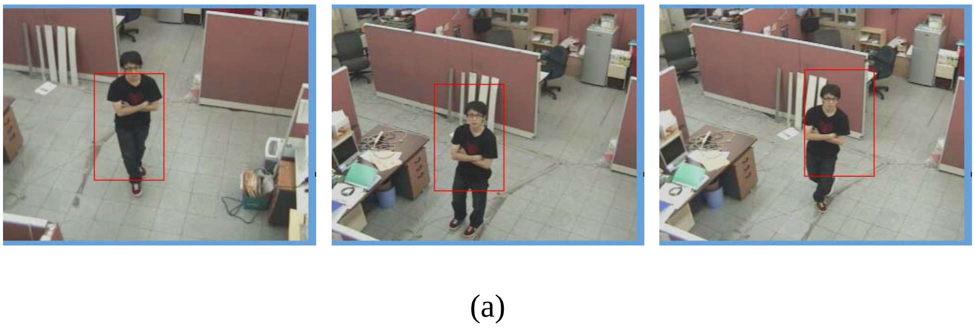


(b)

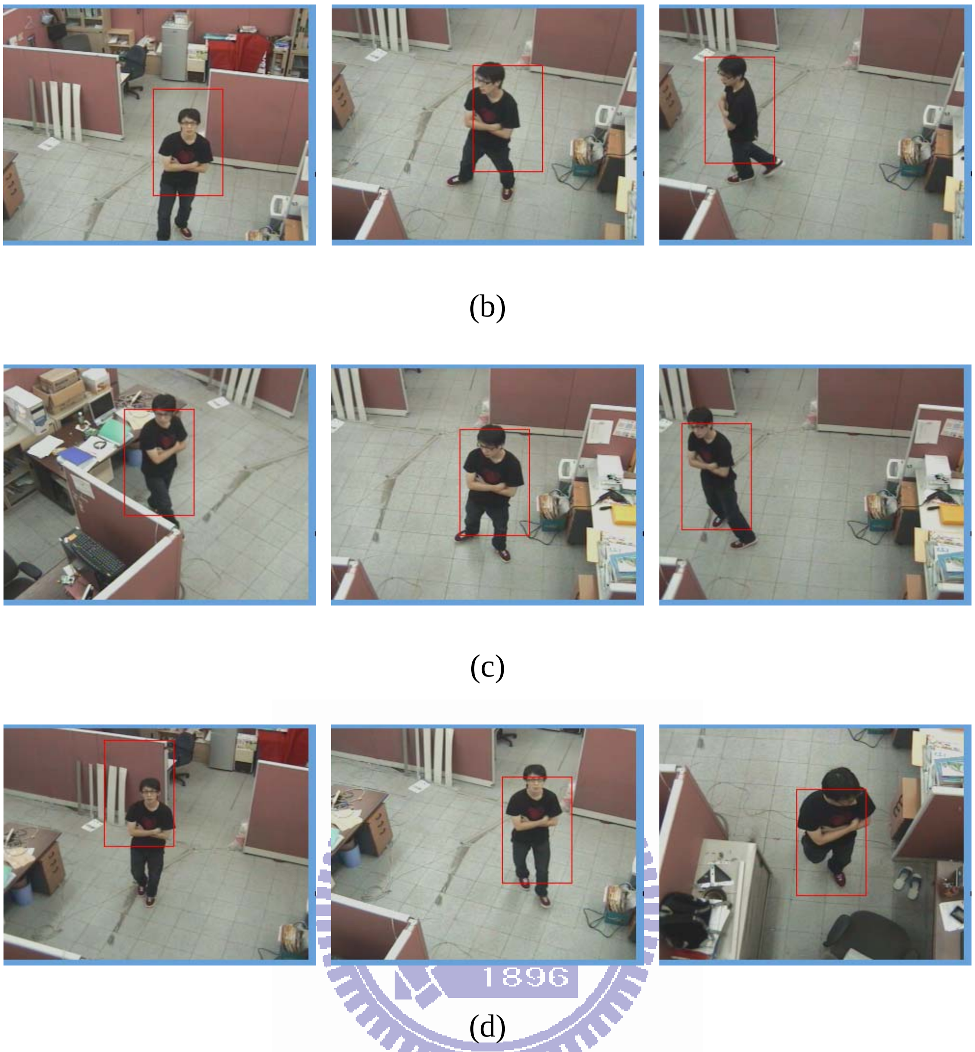
(c)

(d)

Fig. 5-8 Choice RGB color space (a) frame 169, 215 and 283 (b) frame 328, 391 and 446 (c) frame 477, 506 and 537 (d) frame 623, 633 and 675



(a)



(b)

(c)

(d)

Fig. 5-9 Choice Y’VU color space (a) frame 134, 175 and 215 (b) frame 272, 306 and 325 (c) frame 401,449 and 477 (d) frame 497, 528 and 582

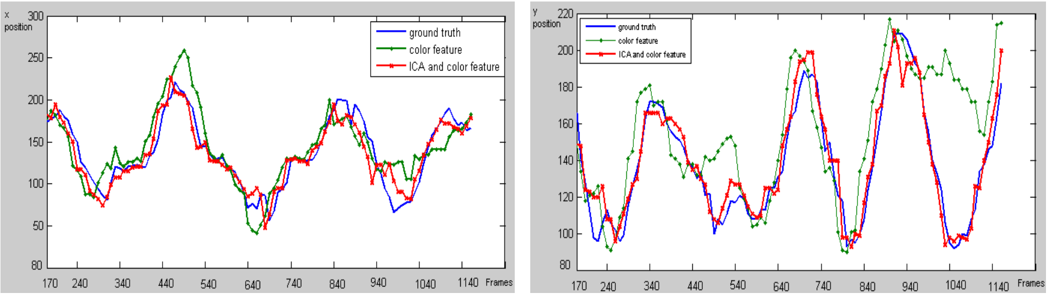
### 5.3.3 Using color and ICA features in human tracking experiment

In this section, we will experiment the ICA features embedded in our mean-shift algorithm. In chapter 4, we have described how to combine color and ICA features in mean-shift algorithm. The purpose of ICA features is used to solve tracking miss problem. Target object is tracked by mean-shift algorithm when a background or nonhuman object has the same color with target object the tracking system maybe track the nonhuman object that will cause tracking miss. So the ICA features are used to solve tracking miss problem when target object and nonhuman object have the same color.

There are 4 cases to experiment moving object and background in the same color situation. In case chair, a man moving in the scenario and drag a chair that has the same color with him. In case chair2 and chair3, there is a chair in the scenario and then the human appeared and then the man walks near the chair. In case same color, the man’s cloth has the same color with floor. All of these four samples have same character that human have same color with background objects. So we use color and ICA embedded in color as feature to observe the tracking performance.

In Fig. 5-10 to Fig. 5-13, straight line, line with cross and line with circle are ground truth, color feature and ICA embedded color feature tracking position, respectively. All of these features are present for x and y position in 2-D coordinate.

The line with circle has big error to ground truth line because when human and chair have the same color so in tracking system they have same similarity so the tracking system tracks the chair as shown in Fig. 5-10. In Table 5-2 shows the MAE (mean absolute error) in color feature is 33.573. In Fig. 5-11, the line with cross has small error to ground truth line because the human and nonhuman have different essence in the same color situation. In Table 5-2 shows the MAE in color feature is 14.59. In Fig. 5-11, the ICA embedded in color feature still has good performance. Fig. 5-12 and Fig. 5-13 are sample of chair2 and same color, respectively.



(a) (b)

Fig. 5-10 Sample of chair. (a) x position (b) y position

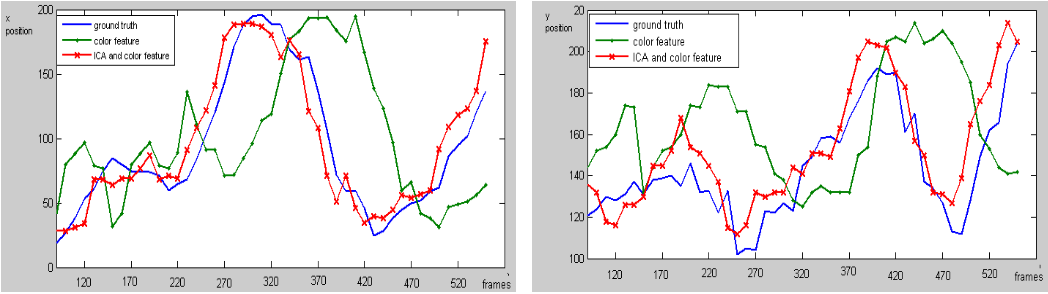
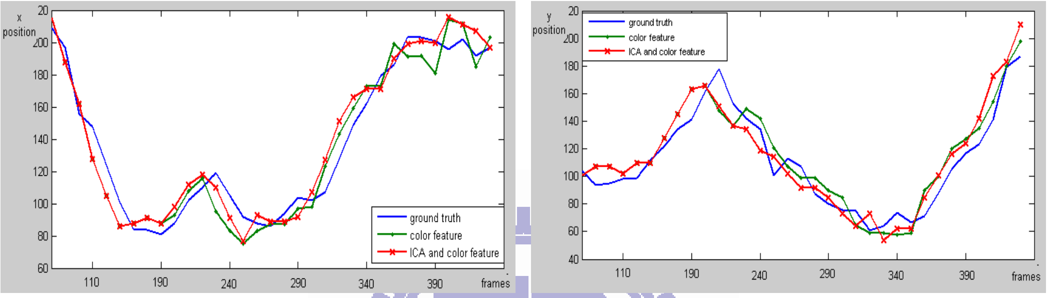
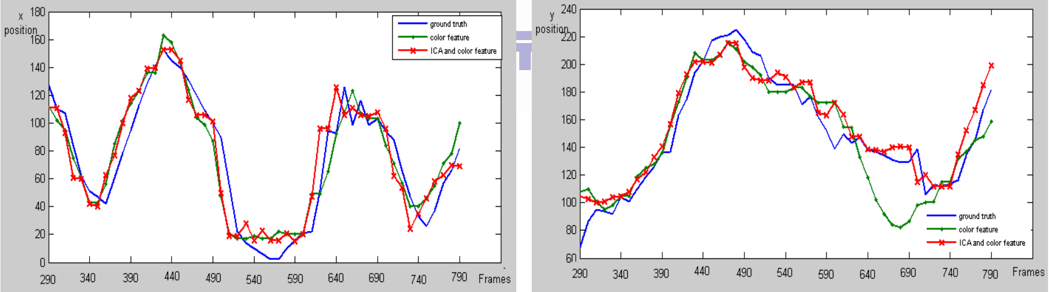


Fig. 5-11 Sample of chir3. (a) x position (b) y position



(a) (b)

Fig. 5-12 Sample of chair2 (a) x position (b) y position



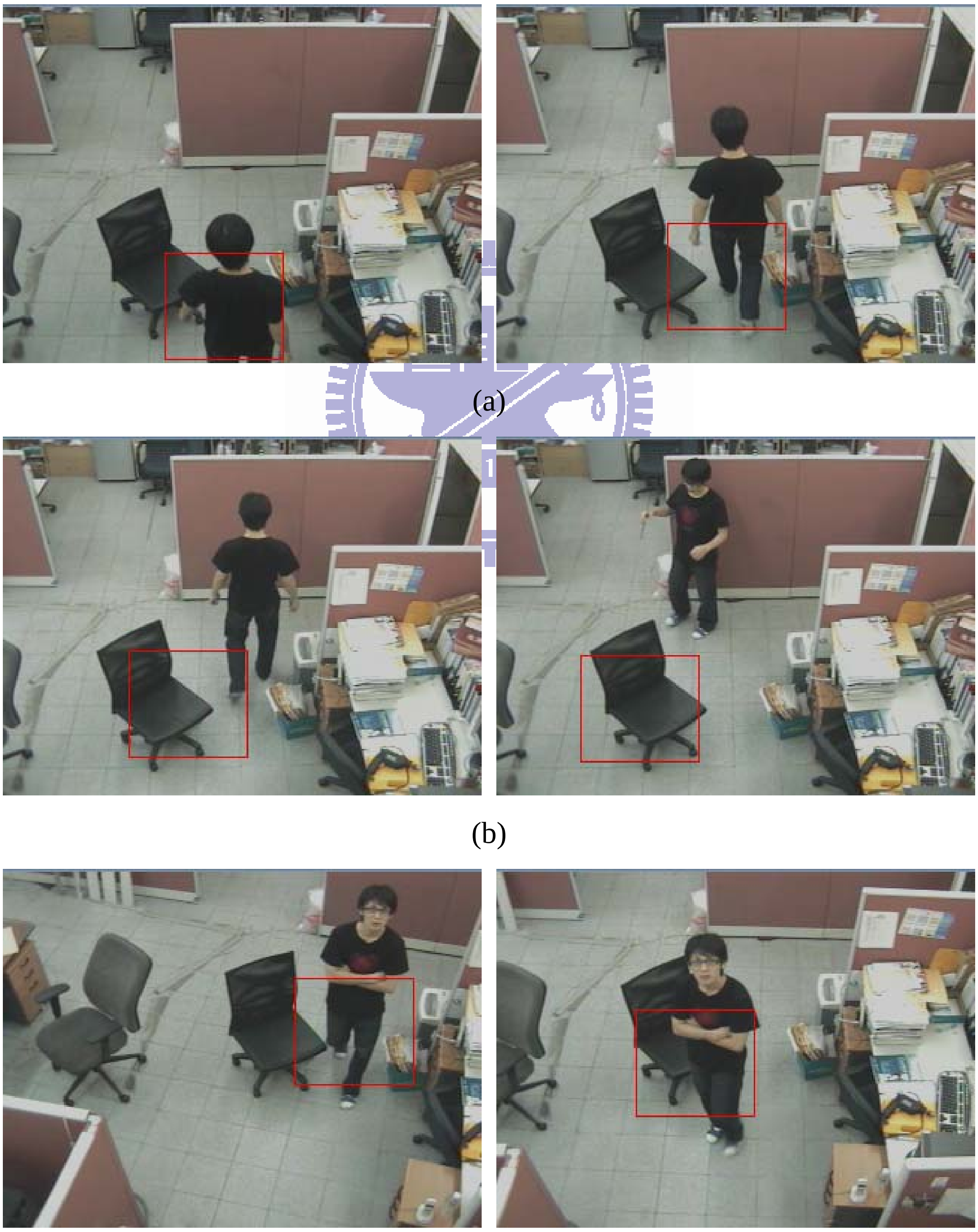
(a) (b)

Fig. 5-13 Sample of chair2 (a) x position (b) y position

Table 5-2 Color and ICA feature in different case

| Sample | Color feature MAE | ICA + color feature MAE |
|---|---|---|
| Chair | 33.573 | 14.59 |
| Chair3 | 68.97 | 21.4 |
| Chair2 | 16.1 | 16.3 |
| Same color | 21.5 | 19.2 |

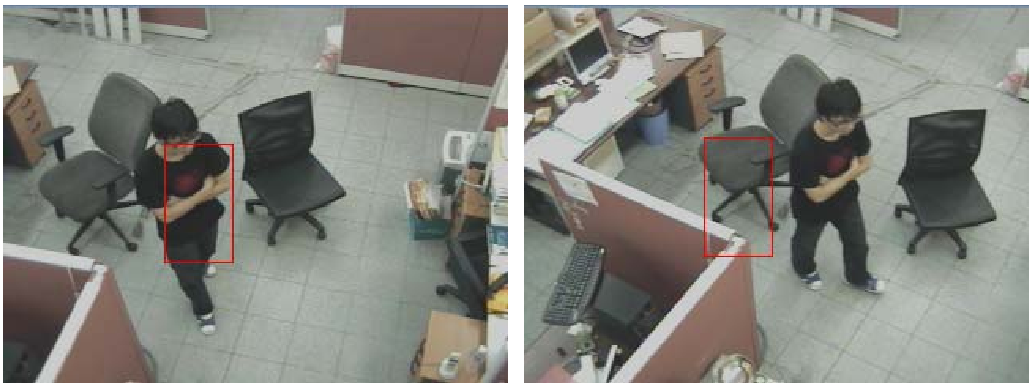
In Fig. 5-14 shows that only use color as feature the tracking will miss tracking when human and chair have same color. In Fig. 5-15 color and ICA features used and the tracking system will not missed when human and chair have the same color. Fig. 5-15 sample shows tracking algorithm will not miss anymore that ICA can solve human and chair have the same color situation because the ICA features have human essence so the ICA in human and chair are not similar.



(a)

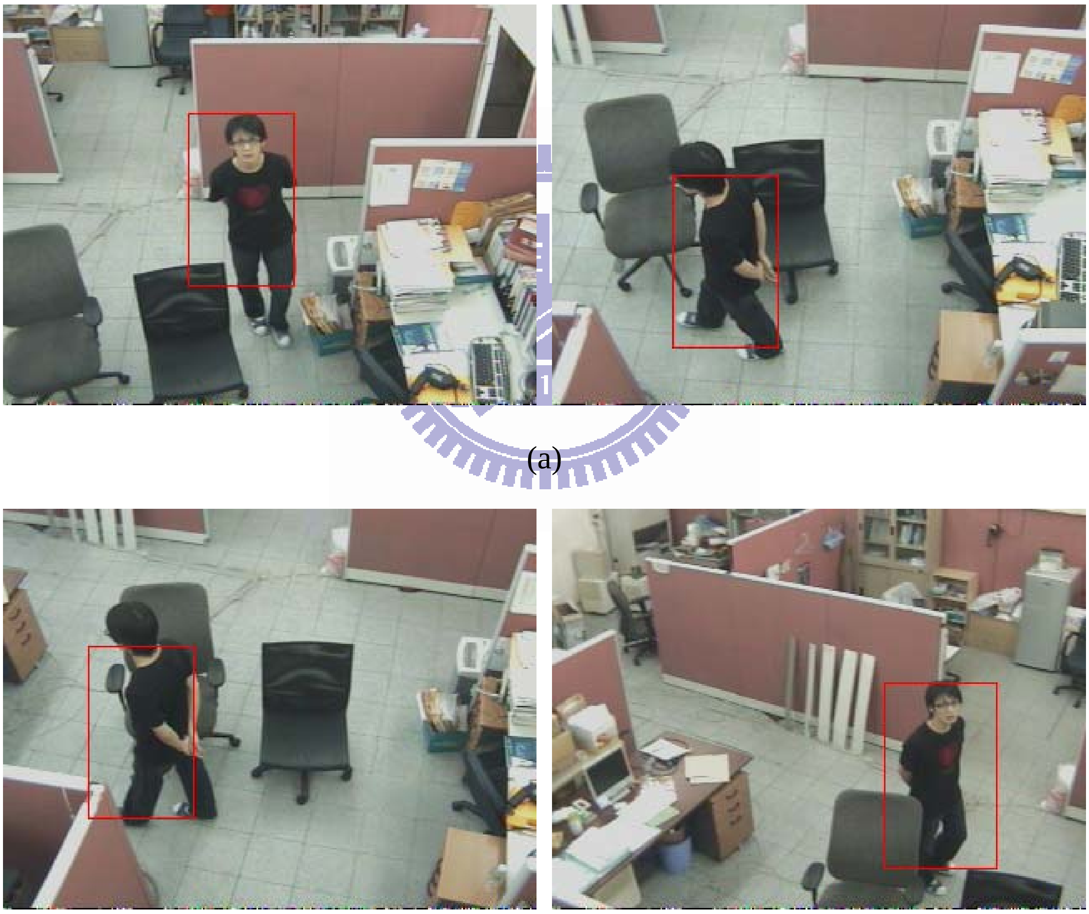
(b)

(c)



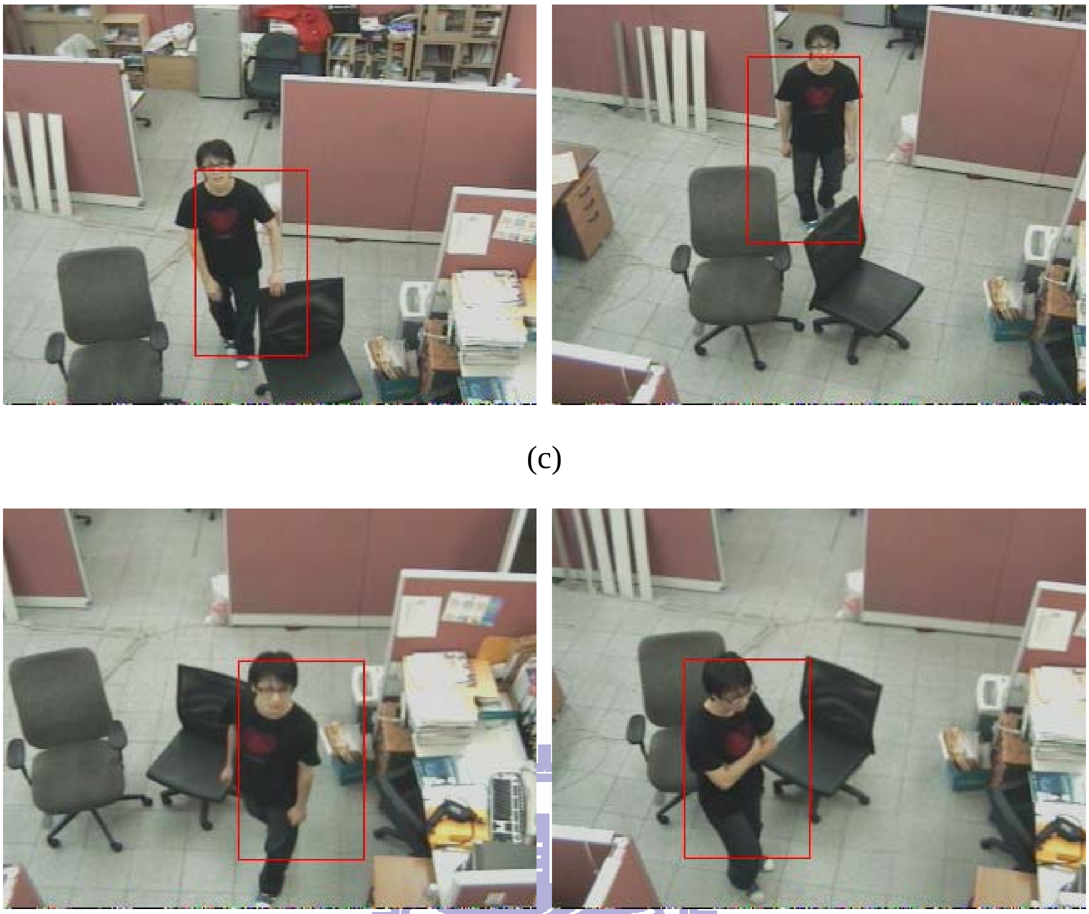
(d)

Fig. 5-14 Using color feature. (a) frame 619 and 634 (b) frame 645 and 677 (c) frame 746 and 761 (d) frame 792and 813



(a)

(b)



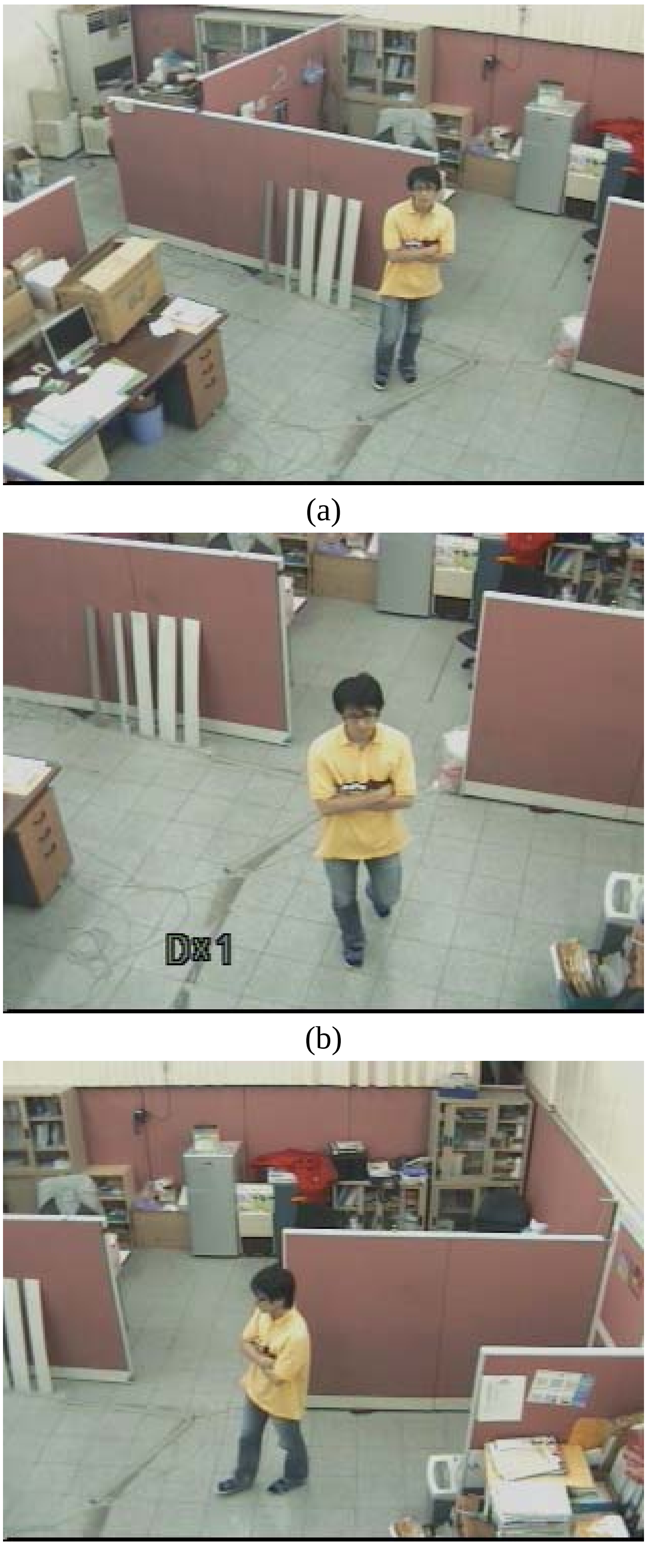
(c)

(d)

Fig. 5-15 Using color and ICA features. (a) frame 369 and 422 (b) frame 453 and 518 (c)frame 593 and 638 (d) frame 694 and 723

## 5.3.4 Human tracking by zoom in/out control experiment

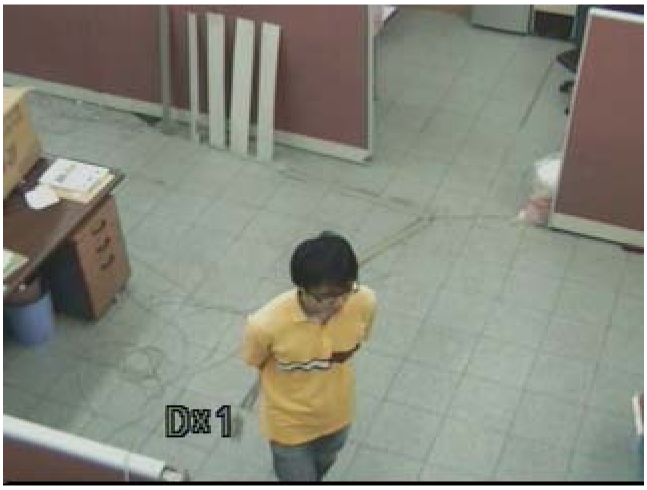
Human tracking by active camera control successfully implement in our system. Although it can achieve target object always keep in field of view and in the center of monitor screen. Sometimes the target object may be far away our camera the object's size too small result resolution unclear. So the zoom in/out operation was used to improve this problem. In chapter 4, we have introduced the ROI scale resize method that successfully used in human tracking when camera is moving. ROI scale resize still can used in zoom in/out operate. Here when the ROI scale was be adjusted if the new size of ROI is small than the old one then the zoom out will be operated. Otherwise when the ROI scale was be adjusted if the new size of ROI is larger than the old one then the zoom in will be operated as shown in Fig. 5-16.



(a)

(b)

(c)



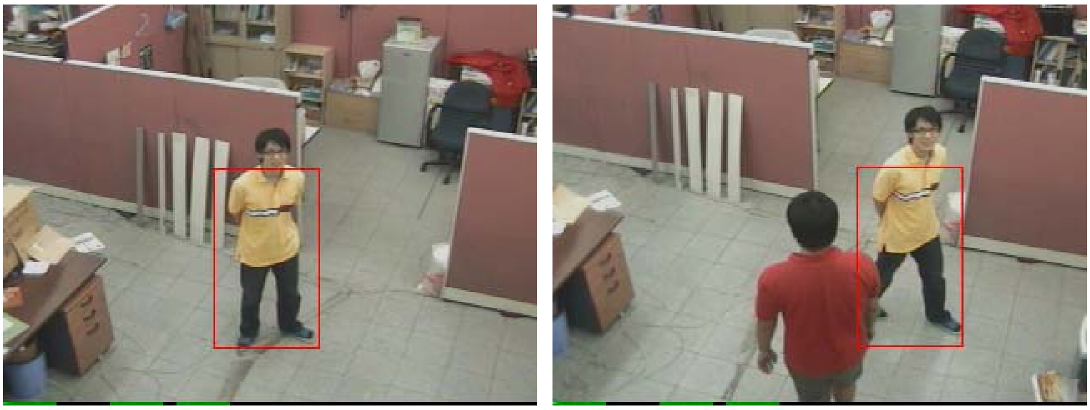
(d)

Fig. 5-16 Zoom in/out tracking (a) original image at frame 494 (b) At frame 529 zoom in (c) At frame 744 zoom out (d) At frame 798 zoom in.

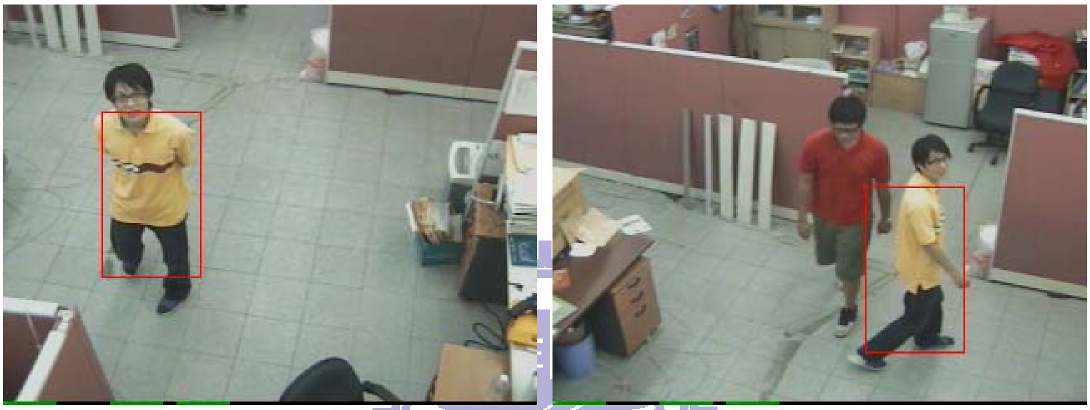
## 5.4 Human tracking in multiple objects experiment

In previous section, we have experiment the single object tracked by our algorithm on active camera. And it can achieve good performance just like pan/tilt control, zoom in/out operation, ICA feature and ROI scale resize promote tracking robustness in single object situation. But in generally situation it is not always only single object in the scenario so in this section we will experiment multiple object in the scenario. Sometimes the object may occlude our target object or across with each other. So in this section objects occlude and across are our main experiment topics.

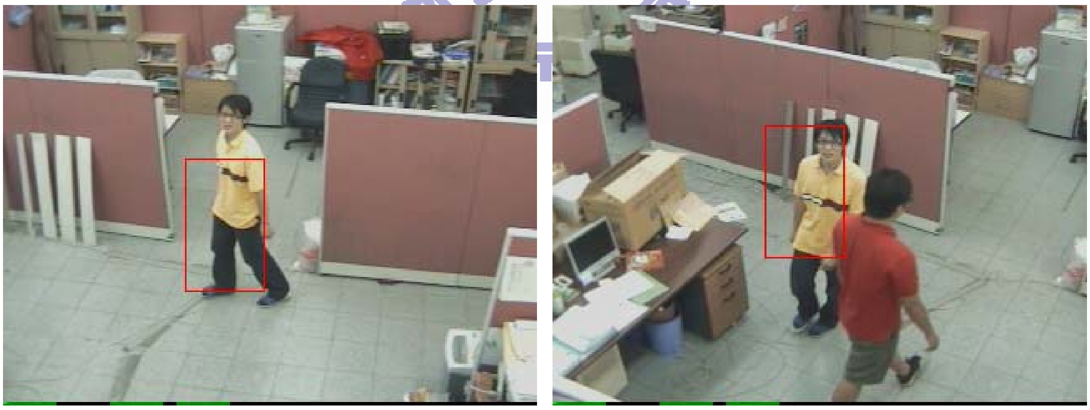
Fig. 5-17 shows the target object was be locked by our tracking algorithm, after a period of time a person with red cloth walk into the scenario. The person will across the target object and our tracking algorithm and active camera will move smoothly and stably. Fig. 5-18 shows the target object was be occluded by the person with red clothes. But the person will not influence our system. That means our system has robust lock ability.



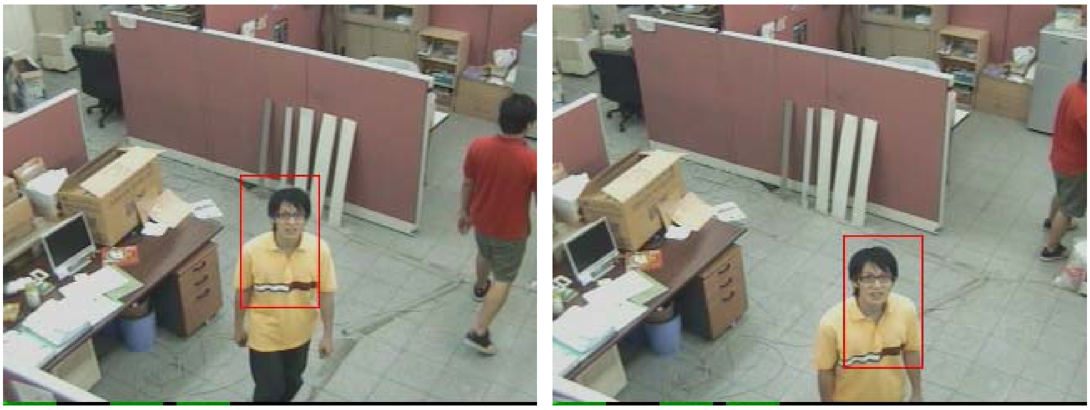
(a)



(b)

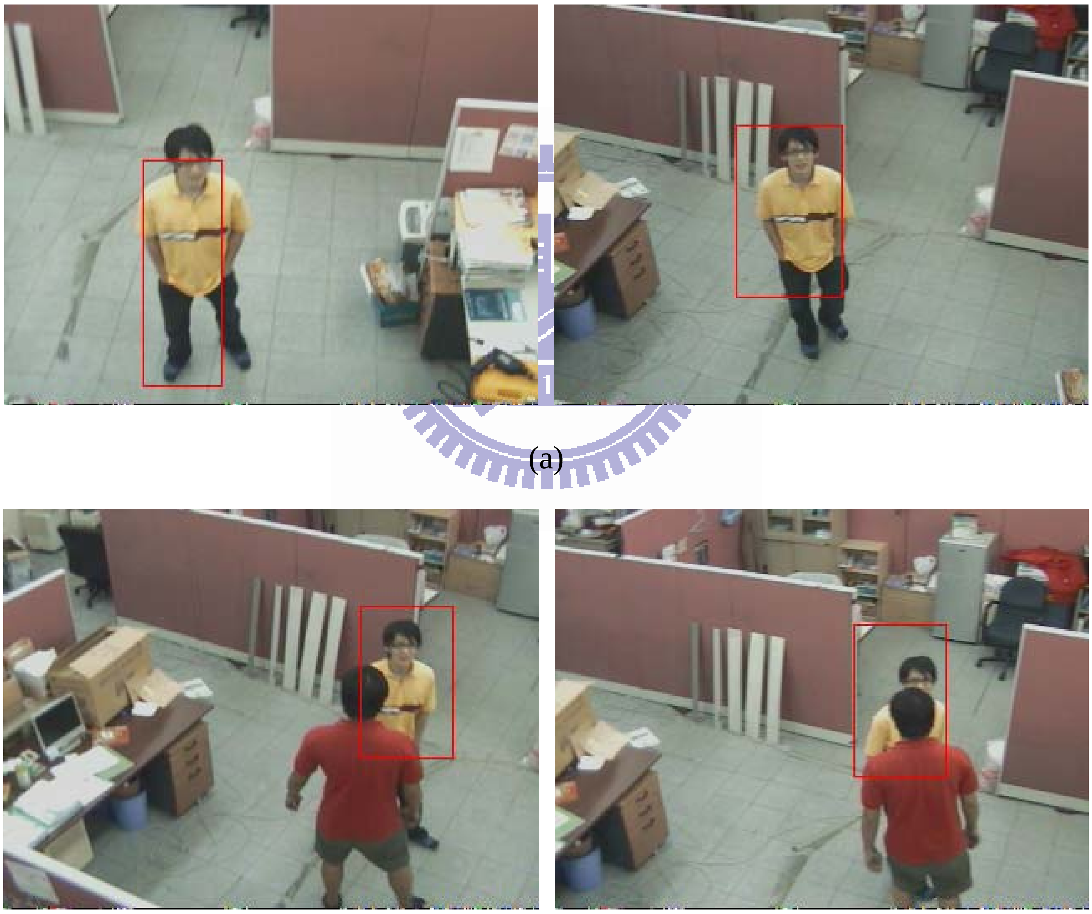


(c)



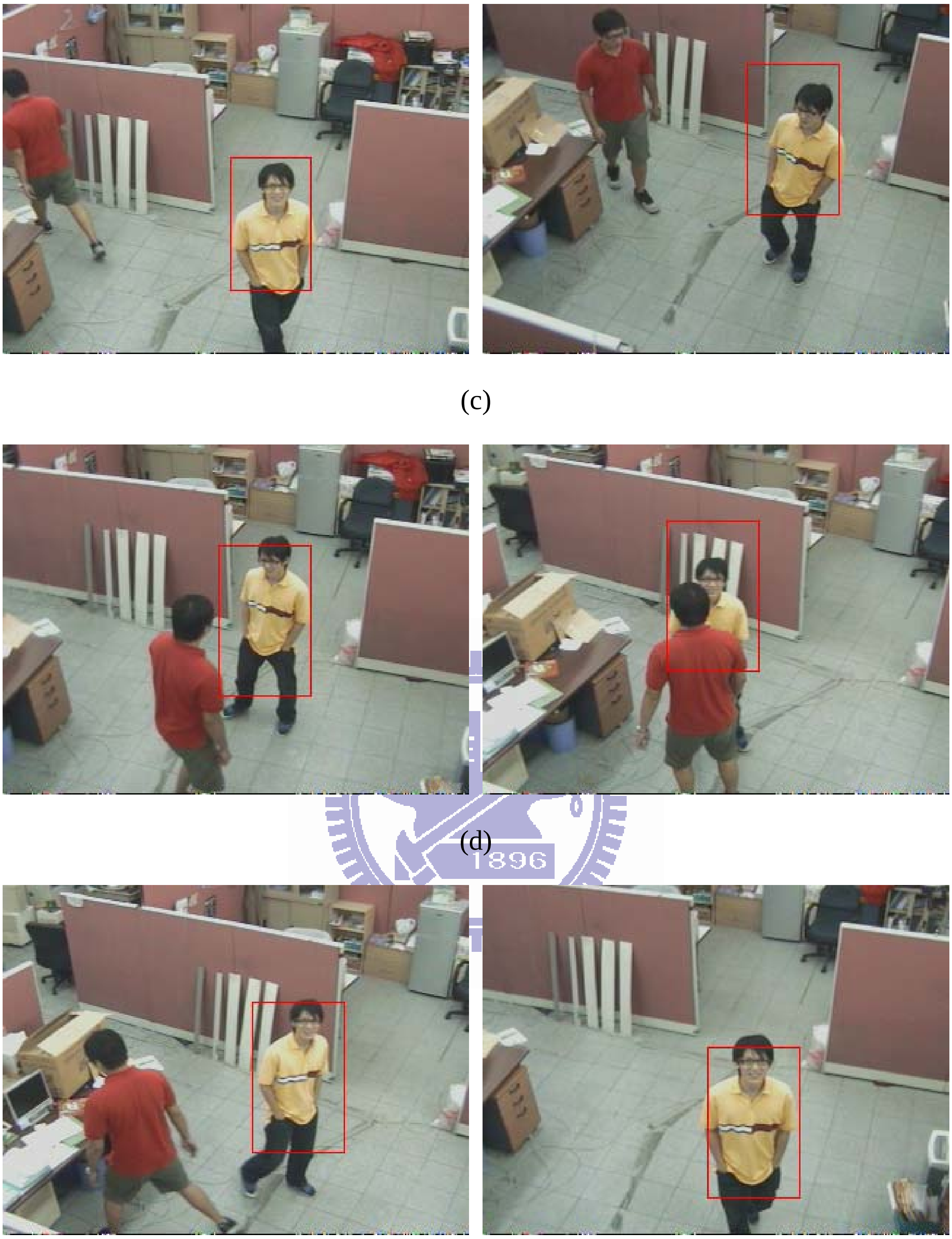
(d)

Fig. 5-17 Across case. (a) frame 2 and 32 (b)frame 88 and 174 (c)frame 209 and 243



(a)

(b)



(c)

(d)

(e)

Fig. 5-18 Occlude case. (a) frame 211 and 252 (b)frame 390 and 403 (c)frame 431 and 572 (d) frame 601 and 627 (e) frame 635 and 667

# Chap 6

# Conclusions and Future work

## 6.1 Conclusions

The experiment results show that the system is capable to track moving humans by mean-shift algorithm with active camera. The system can tracks human not only in single object scenario but also in multiple objects scenario as shown in chap 5. In low resolution situation, the system adaptively operates zoom in/out to adjust resolution.

There are several contributions in this research:

1. Our system can exactly distinguish human and nonhuman.
2. The extracted ICA features include human essence, so human tracking can still lock target object when the target and background have the same color.
3. In the pass, temporal difference has been used to determine the moving object position; therefore, active camera needs to be in fixed circumstance. The active camera is able to keep moving the whole time, because the temporal difference is neglected.
4. ROI can automatically be resized to match the real target's size.

## 6.2 Future work

In our system, the moving human can be detected and tracked smoothly and continuously but if the moving human in the complex environment that will result tracking lose. For example, the background has very bright light that will result moving human changes it character.

We use temporal difference, x-axis and y-axis to redefined target object ROI size. The experiment result has shown in chap5. There is a problem in this method. When the active camera move and use temporal difference, in the general situation we get blur and unclear image. This problem will influence our ROI resizing. There are some methods to solve this problem. For example [37], use different kernel function scale to adjust height and width avoid temporal difference used.

The active camera is driven by pelco P protocol and use position based control pan or tilt. Although, drive active camera to control direction successfully. But the speed of pan or tilt is fixed. Maybe the moving human motion can be considered and use motion to drive active camera speed and direction. That will result active camera moves more reliable.

# References

[1] Wei Guo, Dong-Liang Bi, Lu Liu "Human motion tracking based on shape analysis" in proc. of the 2007 international conference on wavelet analysis and pattern recognition, Beijing , china, 2-4 nov. 2007.

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

[3] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Addison-Wesley, New York, 1992.

[4] D. Marr, E. Hildreth, Theory of edge detection, Proc. R. Soc. London 207 (1980) 197–217.

[5] T. Law, H. Itoh, H. Seki, Image filtering, edge detection, and edge tracing using fuzzy reasoning, IEEE Trans. Pattern Analysis Mach. Intell. 18 (1996) 481–491.

[6] 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.

[7] Dalal N, T riggs B, “Histograms of oriented gradients for human detection”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol 1,pp. 886-893, 2005

[8] 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.

[9] K. Fukunaga, L.D. Hostetler, “The estimation of the gradient of a density function, with applications in pattern recognition,” IEEE Trans. Inform. Theo. 21 (1) (1975) 32–40.

[10] Montabone S, Soto A, “Human detection using a mobile platform and novel features derived from a visual saliency mechanism”, Image and Vision Computing Volume. 28, Issue.3, pp. 391-402, 2010

[11] 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.

[12] 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.

[13] P. Viola, M. Jones, and D. Snow, “Detecting Pedestrians Using Patterns of Motion and Appearance,” Int’l J. Computer Vision, vol. 63, no. 2, pp. 153-161,

2005.

[14] Dimitrijevic M, Lepetit V, Fua P, "Human body pose detection using Bayesian spatio-temporal templates", Computer Vision and Image Understanding, Vol.104, issue.2-3, pp.127-139,2006 .

[15] Plamen P, Ognian B and Krasimir M, “Face Detection and Tracking with an Active Camera”, International IEEE Conference Intelligent Systems, 2008 4th.

[16] Smith, A. R., ”Color Gamut Transform Pairs,” Computer Graphics, Vol. 12(3), pp. 12-19, 1978

[17] http://en.wikipedia.org/wiki/HSV_color_space#Conversion_from_RGB_to_HSL_or_HSV

[18] http://www.mathworks.com/access/helpdesk/help/toolbox/images/f8-20792.html

[19] Comaniciu, D., Ramesh, V., and Meer, P. “Kernel-based object tracking.” IEEE Transactions on Pattern Analysis and Machine Intelligence, 25, 5 (2003), 564—577.

[20] D. Freedman and P. Kisilev, " Fast mean shift by compact density representation." IEEE Conference on Computer Vision and Pattern Recognition Pages: 1818-1825 Published: 2009.

[21] Wang F.L., Yu S.Y. and Yang J., "Robust and efficient fragments-based tracking using mean shift." Aeu-International Journal of Electronics and Communications Volume: 64 Issue: 7 Pages: 614-623 Published: 2010

[22] F. Porikli, O. Tuzel, “Multi-kernel object tracking,” IEEE Int. Conf. Multimedia Expo (2005) 1234–1237.

[23] C. Yang, R. Duraiswami, L. Davis, “Efficient mean-shift tracking via a new Similarity measure.” in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 176–183.

[24] P. Pe´rez, C. Hue, J. Vermaak, M. Gangnet, Color-based probabilistic tracking, in: Proceedings of European Conference on Computer Vision, Copenhagen, Denmark, 2002.

[25] K. Nummiaro, E. Koller-Meier, L. Van Gool, An adaptive color based particle filter, Image Vis. Comput. 21 (1) (2003) 99–110.

[26] O. Williams, A. Blake, R. Cipolla, Sparse bayesian learning for efficient visual tracking, IEEE Trans. Pattern Anal. Mach. Intell. 27(8) (2005) 1292–1304.

[27] A. Agarwal and B. Triggs, “Recovering 3D human pose from monocular images,” IEEE Trans. Pattern Analysis Machine Intelligence, vol. 28, no. 1, pp. 44-58, Jan. 2006.

[28] Bohyung Han and Larry Davis, "Object tracking by adaptive feature extraction", International Conference on Image Processing, Singapore, 2004.

[29] Robert T. Collins, Yanxi Liu and Marius Leordeanu, "Online Selection of

Discriminative", IEEE Transactions on Pattern Analysis and Machine intelligence, VOL. 27, NO. 10, 2005.

[30] D. Murray and A. Basu, “Motion tracking with an active camera,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 5, pp. 449–454, May 1994.

[31] Lin C., Wang C., Chang Y., Chen Y. “Real-time object extraction and tracking with an active camera using image mosaics”. Proceedings of the IEEE International Workshop on Multimedia Signal Processing, Dec. (2002)

[32] Collins, R., Amidi, O., Kanade, T.: An active camera system for acquiring multi-view video. In: Proceedings of the 2002 International Conference on Image Processing, Sept. (2002)

[33] L. Fiore, D. Fehr, R. Bodor, A. Drenner, G. Somasundaram and N. Papanikolopoulos, "Multi-camera human activity monitoring", Springer Netherlands, Journal of Intelligent & Robotic Systems, pp.5-43, 2008.

[34] R. Venkatesh Babu a, Patrick Pe´rez b, Patrick Bouthemy, "Robust tracking with motion estimation and local Kernel-based color modeling", ,Image and Vision Computing, pp.1205–1216, 2007.

[35] Feng S., Guan Q., Xu s. and Tan F., "Human tracking based on mean shift and Kalman Filter", International Conference on Artificial Intelligence and Computational Intelligence,2009

[36] Yizong Cheng, "Mean shift, mode seeking, and Clustering", IEEE Transaction on pattern analysis and machine intelligence, VOL.17, NO.8, August 1995.

[37] Robert T. Collins, "Mean-shift Blob Tracking through Scale Space", IEEE computer society conference on computer vision and pattern recognition, Pages: 234-240, Published: 2003

[38] Tsia-Jung Chung and Chin-Teng Lin ,"Human Detection and Tracking Based on Modified Independent Component Analysis" National Chiao Tung University ,publish 2008