

利用陰影去除及重疊偵測之人類行為分析
Human Behavior Analysis with Shadow Cancellation and Occlusion
Detection Techniques

研究生：李卓皓

Student : Zhuo-Hao Lee

指導教授：李素瑛

Advisor : Suh-Yin Lee

國立交通大學
資訊科學與工程研究所
碩士論文



Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年八月


利用陰影去除及重疊偵測之人類行為分析

研究生：李卓皓

指導教授：李素瑛

國立交通大學資訊工程與科學系

摘要



監控系統在很多領域上有很廣泛的應用，如在家庭保全、保安、病人監控等。在這些應用中，一個智慧型的監控系統需要具有即時觀察環境以及即時發送訊息的能力。在這篇論文中，我們提出了一個以追蹤物件以及分析人類不正常行為的監控系統，例如：暈倒。首先，我們使用以背景作為參考之動態物件切割演算法把移動中的物體從背景中分割出來。接著，我們使用陰影去除演算法把陰影消除，以得到較佳的切割物件。經過陰影去除後，追蹤演算法使用了兩個配對涵式來找出人的移動軌跡。最後，我們使用找出來的移動軌跡以及一個事先定義好的有限狀態機來分析人的不正常行為。我們測試陰影去除、追蹤演算法和人類不正常行為分析的效果。我們使用了幾個監控影片來測試系統，獲得了滿意的實驗結果。


Human Behavior Analysis with Shadow Cancellation and Occlusion Detection Techniques

Student: Zhuo-Hao Lee

Advisor: Prof. Suh-Yin Lee

**Institute of Computer Science and Information Engineering
National Chiao Tung University**

Abstract

The logo of National Chiao Tung University is a circular emblem with a blue border. Inside the circle, there is a stylized representation of a building or a ship, with the letters 'ES' and 'A' visible. Below the emblem, the year '1896' is inscribed.

Surveillance is widely used in many fields, such as home care, security, patient monitoring and so on. In such applications, an intelligent system is needed to monitor the situation and give corresponding responses in real time. In this thesis, we propose an object-based video tracking and human abnormal behavior (e.g.faint) surveillance system. First we use a background-registration segmentation algorithm to segment the moving objects. In order to obtain better segmentation, we use a shadow cancellation algorithm to eliminate shadow. After shadow cancellation, two matching algorithms are used in tracking algorithm to extract the human trajectories. Finally, we use the trajectory of human and a pre-decided finite state machine to analyze the abnormal of human behavior. We perform the experiments on effectiveness of shadow cancellation, tracking algorithm and human abnormal behavior analysis. We test our system with several surveillance video sequences and we obtain a satisfactory experimental result.

Acknowledgement

I sincerely appreciate the guidance and the encouragement of my advisor, Prof. Suh-Yin Lee. She encouraged me in exploiting research topics freely and enthusiastically helped me. Without her, I cannot complete this thesis.

Besides, I would like to extend my thanks to the lab mates in the Information System Laboratory, especially Mr. Ming-Ho Hsiao and Mr. Yi-Wen Chen. They gave me a lot of suggestions and shared their experience.

Finally, I want to express my appreciation to my parents for their support. They gave me the opportunity to have good education. This thesis is dedicated to them.



Table of Contents

摘要.....	i
Abstract.....	ii
Acknowledgement.....	iii
Table of Contents.....	iv
Lists of Figures.....	vi
Lists of Tables.....	vii
Chapter 1 Introduction.....	1
1.1 Motivation and Introduction.....	1
1.2 System Overview.....	2
1.3 Organization.....	3
Chapter 2 Object Segmentation and Features Extraction.....	4
2.1 Related Work of Video Object Segmentation.....	4
2.2 Video Object Segmentation.....	6
2.2.1 Inter-frame Differencing.....	7
2.2.2 Dynamic Threshold Decision.....	8
2.2.3 Background Buffer Update.....	9
2.2.4 Background Differencing.....	11
2.2.5 Shadow Elimination.....	12
2.2.6 Morphological Operation.....	15
2.2.7 Connected Component Labeling Algorithm and Size Filtering ...	15
2.3 Video Object Feature Extraction.....	16
Chapter 3 Video Object Tracking Algorithm.....	20
3.1 Related Work of Video Object Tracking.....	20
3.2 Video Object Tracking.....	21
3.2.1 Single Object Matching Algorithm.....	23
3.2.2 Multiple Objects Matching Algorithm.....	29
3.2.3 Objects Tracking Algorithm.....	32
3.3 Problems in Object Tracking Algorithm.....	36
Chapter 4 Human Behavior Analysis.....	38
4.1 Related Work of Human Behavior Analysis.....	39
4.2 Posture Analysis.....	40
4.3 Abnormal Behavior Analysis via Finite State Machine.....	42
Chapter 5 System Architecture and Experimental Results.....	44
5.1 System Architecture Overview.....	44
5.2 Experimental Results of the Video Object Segmentation.....	44

5.3 Experimental Results of the Video Object Tracking	47
5.4 Experimental Results of the Human Behavior Analysis	53
Chapter 6 Conclusion and Future Work	54
Reference	56



Lists of Figures

Fig. 1. System architecture overview.....	2
Fig. 2. Segmentation process diagram	7
Fig. 3. The histogram of a difference image	9
Fig. 4. The cumulative histogram and the interpolated lines	9
Fig. 5. Shadow Elimination Algorithm	14
Fig. 6. The increase of width of bounding box is 200%	18
Fig. 7. The increase of width of best-fit-ellipse is about 150%	19
Fig. 8. The flow chart of tracking algorithm.....	23
Fig. 9. A matching example for three object O_1 , O_2 and O_3	26
Fig. 10. The perfect matching	27
Fig. 11. Illustration of N-best algorithm.....	28
Fig. 12. The N-best algorithm and the single object matching algorithm ...	28
Fig. 13. The relationship between possible collision and possible occlusio	31
Fig. 14 The multiple objects matching algorithm.....	32
Fig. 15. The flow chart of tracking algorithm.....	34
Fig. 16. Tracking Algorithm.....	35
Fig. 17. Split and merge occur in same time.....	36
Fig. 18. The problem occurs in object trajectories updating.....	37
Fig. 19. Three posture and its x,y variance	40
Fig. 20. The experiment of postures versus logarithm of P.....	41
Fig. 21. A finite state machine for the analysis of abnormal behavior.....	43
Fig. 22. An adaptive finite state machine for the analysis of abnormal behavior.....	43
Fig. 23. Surveillance system architecture overview	44
Fig. 24. The shadow thresholds $t_1=0.88$ and $t_2=0.01$	45
Fig. 25. The shadow thresholds $t_1=0.88$ and $t_2=0.01$	45
Fig. 26. The shadow thresholds $t_1=0.88$ and $t_2=0.01$	46
Fig. 27. The shadow thresholds $t_1=0.94$ and $t_2=0.01$	46
Fig. 28. After Shadow Elimination.....	47
Fig. 29. Tracking results of the speedway sequence.....	49
Fig. 30. Tracking results of our Lab. Test Seq. 1	50
Fig. 31. Tracking results of the ETRI_C Seq.....	52

Lists of Tables

Table.1. Statistics of tracking and detecting of occlusion events.....	52
Table.2. The recognition rates for three posture.....	53
Table.3. The recognition results for abnormal behavior.....	53



Chapter 1

Introduction

1.1 Motivation and Introduction

In recent years, automated visual surveillance becomes more and more important, mainly due to the advantages it provides in numerous applications such as crime prevention, security, patient monitoring and so on. Thus, a system is needed to monitor the situation and give corresponding responses in real time. In such applications, the analysis of human behaviors is essential and should involve high level understanding of the human actions.

In this thesis, we present a home care surveillance system. The goal of such a kind system is to monitor the moving objects, say human, in the surveillance environment and to analyze the unusually behavior of human, such as faint and falling down, in real time. Thus, video object segmentation is essential to extract the moving objects. After video object segmentation, video object features are extracted to describe video objects. After video object feature extraction, video object tracking is required to track the object trajectory. Finally, object tracking can be used to analyze behavior of human.

In this proposed system, we develop a method to eliminate the shadow and two matching algorithms to track objects. Besides, we design a finite state machine to analyze the abnormal behavior of human. From the experiments, we obtain satisfactory results.

1.2 System Overview

Our proposed system as depicted in Fig.1 contains four modules, which are the video object segmentation module, the video object feature extraction module, the video object tracking module and the human behavior analysis module. The surveillance video data are first captured and input to the video object segmentation module. The object segmentation module segments the moving objects from the background and generates the object masks which indicate the position and the shape of the moving objects. The segmented objects are then input to the video object feature extraction module. In video object feature extraction module, feature such as color histogram, center of objects and the shape of objects are extracted to describe the characteristic of moving objects. Those extracted features are then input to the video object tracking module. The purpose of object tracking module is to match the object features of current input to those features appearing in previous input. Also, the occlusion and the splitting of the objects are detected and handled in the object tracking stage. In the human behavior analysis module, posture of objects being tracked will be analyzed first. Finally, a finite state machine (FSM) and posture analyzed before will be used to analyze the behavior of human.

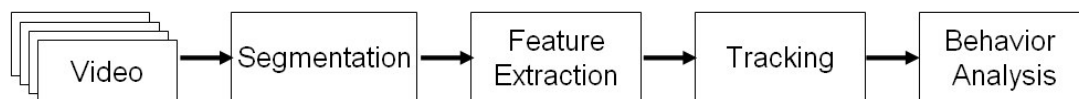


Fig. 1. System architecture overview

1.3 Organization

The rest of the paper is organized as follows. Chapter 2 presents the video object segmentation and our proposed shadow elimination algorithm. Chapter 3 presents the method of video object tracking including two matching algorithms. Chapter 4 presents the method to analyze the abnormal behavior of human. Chapter 5 shows the architecture of the system and the experimental results. We make a conclusion in Chapter 6.




Chapter 2

Object Segmentation and Features Extraction

The first module in the proposed surveillance system is video object segmentation. The task of video object segmentation is to find a mask indicating the shape and the position of the moving objects. In this chapter, we introduce the related work of video object segmentation in Section 2.1. In Section 2.2, we present the proposed method of video object segmentation including the shadow elimination. In Section 2.3, we introduce the features extracted in our system.

2.1 Related Work of Video Object Segmentation



Segmentation algorithms can be classified into two categories, the homogeneity based methods and the change detection based methods. The homogeneity based algorithms [1-4] segment moving objects based on the homogeneity of their color, texture or motion information. Pixels or blocks with similar features are first grouped into small regions, and these regions are then grouped into objects with some other features. Though this kind [1-4] of algorithms can provide precise object masks, however the watershed algorithm for the boundary decision is a computational expensive process. Also, the motion estimation process to compute the precise motion vectors for clustering small regions also takes a lot of time. Thus, this kind of algorithm is not a good choice for a real-time system.

The other category of segmentation algorithms is the change detection based algorithms. This kind of algorithms [5-7] segment objects by taking difference between the current input frame and a reference image, and then a threshold is chosen

to decide a difference mask indicating the shape and the position of the moving objects. Traditionally, the previous input frame is chosen as the reference image. However, there are some well-known drawbacks [8].

First, when the speed of the moving object is not consistent, it becomes impossible to indicate the position using the difference image and thus miss or false alarm in segmentation is unavoidable. Second, the uncovered background is another problem in traditional change-detection algorithms because the uncovered background regions covered by objects in previous frame may be considered as changed. Although uncovered background can be detected and removed when the motion information is taken into consideration, the computation of motion estimation is expensive and greatly lowers the efficiency of the change-detection algorithms.

Recently, some change detection algorithms [8-13, 19] use a reference background image to segment moving objects. The reference background image, which contains the still background without any objects, is acquired beforehand or by some means to be updated dynamically. The change-detection algorithms with registered background effectively solve the problems of uncovered background and inconsistent object speed. Besides, they are efficient and can meet the real-time requirement.

For the change-detection based algorithm, shadow is always extracted accompanying with object [12-14]. The main idea to remove shadow tries to classify the object pixels into real moving objects or shadows. Because of the real time requirement, a simple but efficient algorithm to remove shadows is expected. Based on the shadow elimination method in [12], we develop a new shadow elimination algorithm and present in Section 2.2.5. In our proposed system, we adopt the change-detection based algorithm with registered background to segment moving objects.

2.2 Video Object Segmentation

In our object-based tracking and human behavior analysis system, the first step is to segment the moving objects as precisely as possible. The object segmentation algorithm directly takes the raw video data as input to segment the moving objects in the surveillance video sequences and extracts the object masks to indicate the presence of the moving objects. In the surveillance video, since the position of the camera is always fixed and the background is stationary, so the simplest way to segment moving objects is to use the change detection based method. When comparing a frame to a background image, it is straightforward to consider the regions that change significantly as moving objects. Therefore, selecting background image as reference for the change detection based algorithm can effectively achieve our goal. However, this simple method may fail when the background changes. For reasons above, we use a more complex method to update the background and then to extract moving objects.

However, besides the moving objects that we are interested in, there are other types of changing regions that may be miss-classified during the segmentation process. We classify these miss-classified regions into two types. The first type is the camera noises which are the white noise of the camera and usually small. The second type is 'ghost' which is the changing region that appears and then disappears quickly without steady motion and is usually bigger than the camera noise. The ghost effect is usually resulted from the waving of tree leaves and regional lighting effect. In order to obtain accurate object masks, these annoying changing regions should be filtered out.

In our segmentation algorithm, as shown in Fig.2, we use the change detection based algorithm with background registration and use a size filter to remove the noises and ghosts.

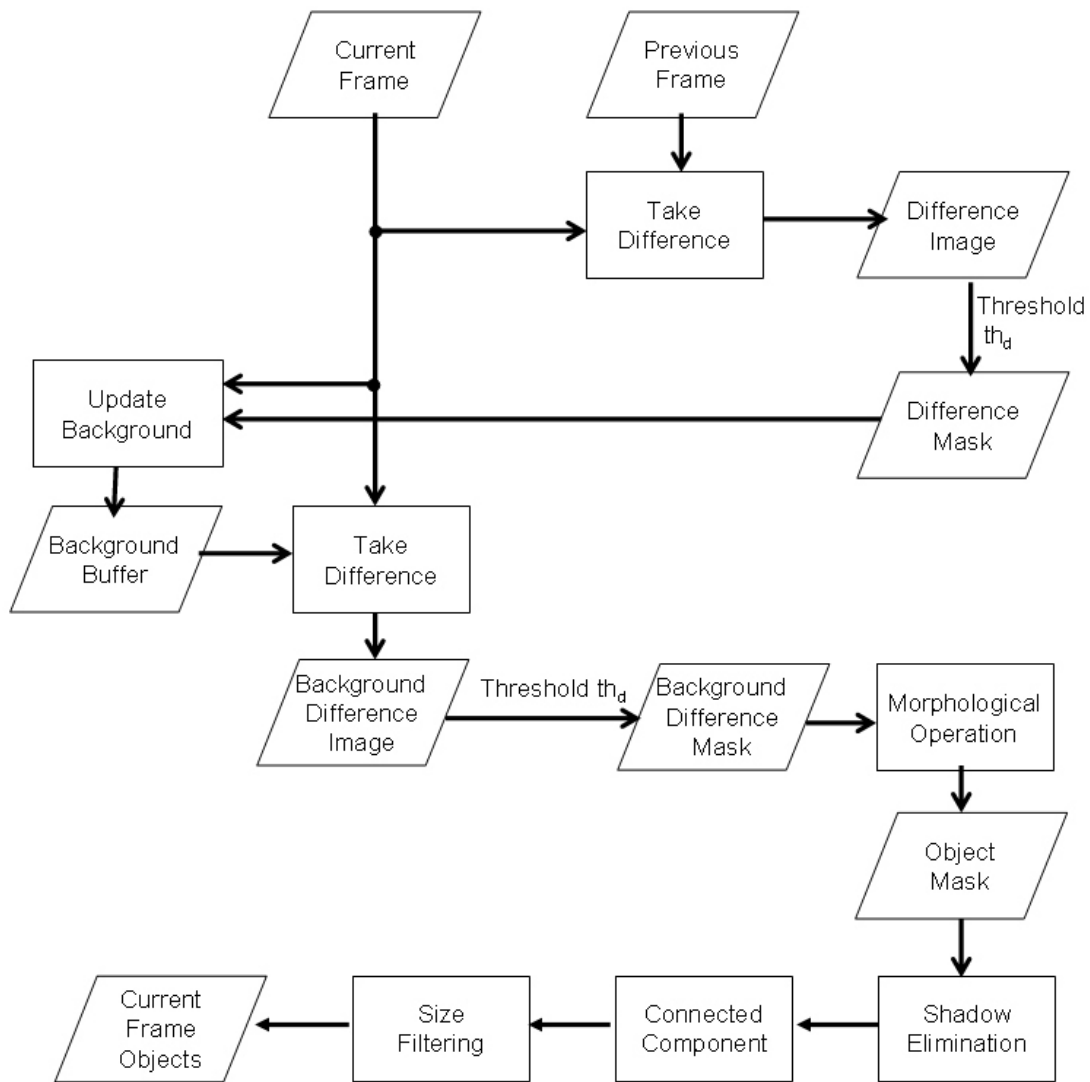


Fig. 2. Segmentation process diagram

2.2.1 Inter-frame Differencing

In the segmentation algorithm, we first compute the inter-frame difference image (DI) between the current input frame and the previous frame. The difference value in the DI shows how strong a pixel change in two consecutive frames and the possibility that a pixel will be considered as changing. Because the human eyes are more sensitive to luminance than to chrominance, we only take difference value on the luminance channel. After taking threshold TH_d on the difference image, we obtain a difference mask (DM) that indicates the possible changing regions between two

consecutive frames. The computations of DI and DM are shown in Eq.(1) and Eq.(2), where the $I^Y(i, j, t)$ and $I^Y(i, j, t-1)$ denotes the pixel value at (i, j) in current frame and previous frame in the luminance channel, respectively. The DM is used to update the background image in the next step.

$$DI(i, j, t) = |I^Y(i, j, t) - I^Y(i, j, t-1)| \quad (1)$$

$$DM(i, j, t) = \begin{cases} 1 & \text{if } DI(i, j, t) > TH_d \\ 0 & \text{if } DI(i, j, t) \leq TH_d \end{cases} \quad (2)$$

2.2.2 Dynamic Threshold Decision

In order to make the segmentation adapt to various kinds of environments and video contents, the threshold TH_d for deciding the DM cannot be fixed and should be selected adaptively. In many researches [8, 10, 43, 44] the values in the difference image can be modeled by a mixture of two distributions. Thus, finding the threshold corresponds to finding the two distribution functions that approximate the histogram of the difference values. Traditionally, the valley between two peaks is found and is chosen as the threshold dividing two distributions. However, in the real case as shown in Fig.3, the histogram fluctuates heavily and it is difficult to find a threshold just by finding a valley. In [43], Wu et al. suggest that the histogram can be converted to a monotonic increasing histogram by accumulating the original histogram values, as shown in Fig.4. In the cumulative histogram, the problem of finding a threshold can be simplified to finding an intermediate point such that two straight lines which are interpolated by the start point, end point and the intermediate point can best approximate the cumulative histogram. Instead of using the ratio histogram in [43], we simply use the difference since the computational cost is much expensive for the ratio histogram.

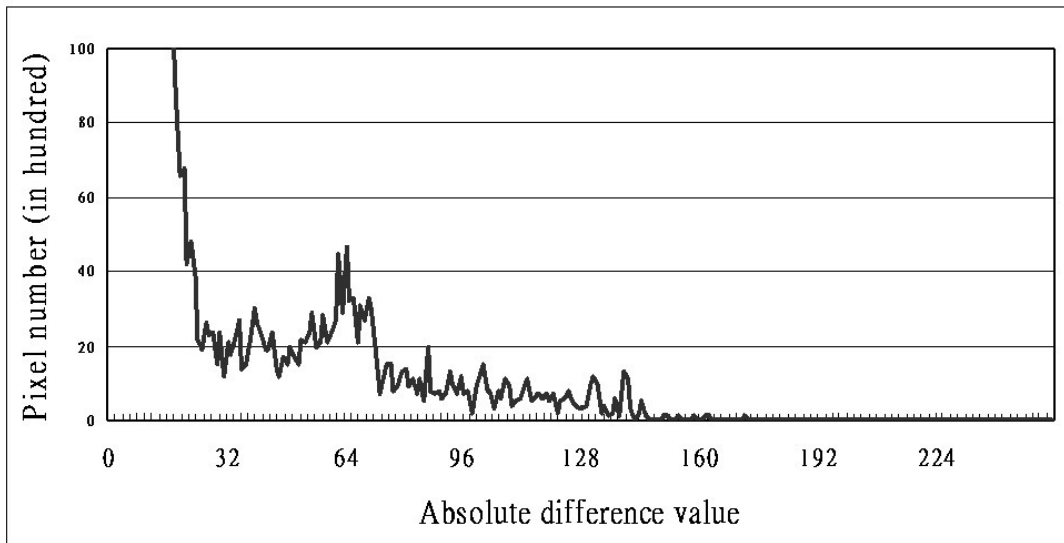


Fig. 3. The histogram of a difference image

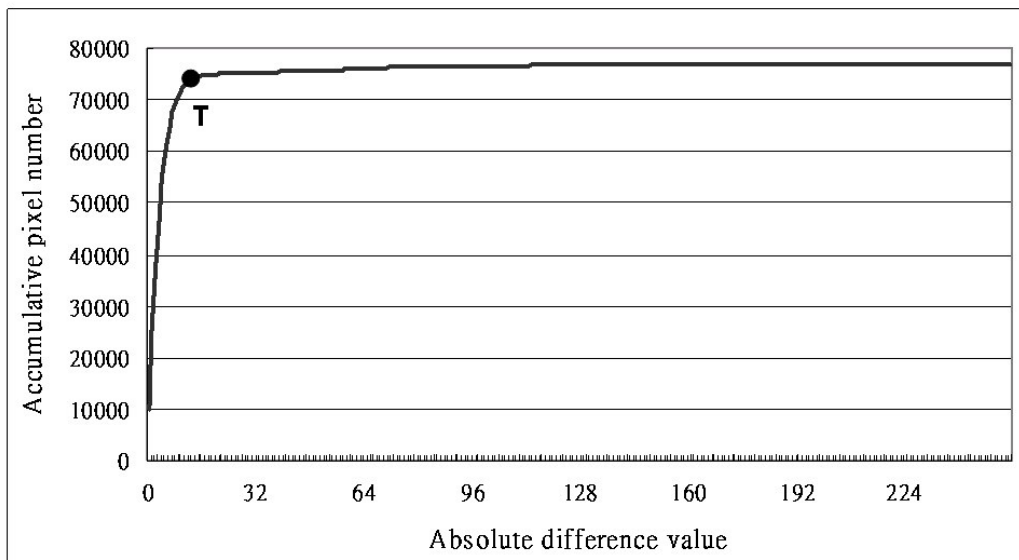


Fig. 4. The cumulative histogram and the interpolated lines

2.2.3 Background Buffer Update

The next step in the segmentation algorithm is to update the background image (BI). In the background registration method [8-13, 19, 44] because the performance of the segmentation result relies on the correctness of the background dramatically, we need a robust method to retrieve and maintain the background image. The simplest way to obtain the background image is to capture the background beforehand.

However, the background image may change slightly and gradually because the luminance may vary with time. In our algorithm, we dynamically update the background buffer using the stationary index (SI). The stationary index is used to record the possibility if a pixel is in the background region. Consider Eq.(2), the value of $DM = 0$ can be considered as background pixel and the value of $DM = 1$ can be considered as moving pixel. Hence, the relationship between DM and SI can be stated in Eq.(3). If a pixel is not moving for many consecutive frames, the possibility to be a background pixel will be high. Therefore, we can use the SI value to update the background image (BI). The background update method is stated in Eq.(4), where $BI^C(i, j, t)$ denotes the value of color channel C of background image at position (i, j) at time t , $I^C(i, j, t)$ denotes the value of color channel C of video image at position (i, j) at time t . The value th_m , which is pre-decided, means a pixel that is not moving for consecutive th_m frames is considered to be a background pixel.

$$SI(i, j, t) = \begin{cases} SI(i, j, t-1) + 1 & \text{if } DM(i, j, t) = 0 \\ 0 & \text{if } DM(i, j, t) = 1 \end{cases} \quad (3)$$

$$BI^C(i, j, t) = \begin{cases} I^C(i, j, t) & \text{if } SI \geq th_m \\ BI^C(i, j, t-1) & \text{if } SI < th_m \end{cases} \quad (4)$$

where C can be Red, Green or Blue.

When the system starts up, the first video frame is set as background buffer to speed up the convergence of background image. After the setting step, we update the background buffer if SI is greater or equal to th_m . The gradual variation can be updated to background buffer in th_m frames. Even if there is a sudden lighting variation when the clouds are dispersed and the sun is revealed, the update equation can also catch up the variation in th_m frames. Hence, the threshold th_m determines the updating rate of background buffer. So, the threshold th_m depend on the environment.

2.2.4 Background Differencing

After we obtain a background buffer, we can segment the moving objects from the background. Unlike the way adopted in computing the difference image, we use the luminance and chrominance channels together instead of using luminance only. Because some moving objects look quite different compared to the background in their color, but the difference between the current frame and the background is almost zero when the chrominance information is discarded. In order to extract accurate object masks and not to miss any important moving objects, the chrominance information must be also considered. Because the importance of the chrominance channels depends on the intensity of the luminance channel, we design a difference score function to evaluate the difference in YUV color space. The different score, denoted as DS, is computed using the equations Eq.(5) through Eq.(7). For a pixel (i,j), we first get the strongest luminance intensity among the current frame and the background image. Then, we decide the weighting factor of the chrominance based on the luminance intensity. Because the valid range of the luminance channel after conversion is from 16 to 235, we can subtract the luminance value from 16 and divide it into 11 levels, which are from 0.0 to 1.0. After that, we can use the weighting sum equation Eq.(7) to compute the color distance in the YUV color space.

$$M(i, j, t) = \max(I^Y(i, j, t), BI_t^Y(i, j, t)) \quad (5)$$

$$w = \text{floor}\left(\frac{M(i, j, t) - 16}{20}\right) / 10 \quad (6)$$

$$DS(I(i, j, t), BI(i, j, t)) = \frac{|I^Y(i, j, t) - BI^Y(i, j, t)|}{1 + (2 \cdot w)} + \frac{w \cdot |I^U(i, j, t) - BI^U(i, j, t)|}{1 + (2 \cdot w)} + \frac{w \cdot |I^V(i, j, t) - BI^V(i, j, t)|}{1 + (2 \cdot w)} \quad (7)$$

$$BDI(i, j, t) = DS(I(i, j, t), BI(i, j, t)) \quad (8)$$

$$BDM(i, j, t) = \begin{cases} 0 & \text{if } BDI(i, j, t) > TH_b \\ 1 & \text{if } BDI(i, j, t) \leq TH_b \end{cases} \quad (9)$$

Sometimes an object enters a frame and then keeps staying in the same position on the xy-plane. We call such a kind of objects as ‘stopped object’ since the motions in both the x and y direction are almost zero. Because the algorithm updates the input frame to the background for the unchanged regions, the color of the stopped objects will be updated to background buffer when they stop too long. In this case, object regions will be false alarmed because the background has been incorrectly updated. Although this problem can be solved by lengthening the interval of background updating, the time needed to adapt to the luminance variations is also lengthened. Thus it is a tradeoff and both of the cases must be taken into consideration.

After we finish computing the background difference image using the difference score function, another dynamically selected threshold TH_b is applied to get the background difference mask (BDM), as shown in Eq.(9). The background difference mask extracted here indicates the moving object regions relative to the reference background image. However, the shadow is always included in the background difference masks. Thus, further filtering of shadow is required.

2.2.5 Shadow Elimination

Shadow is always extracted accompanying objects in change detection based algorithm. In order to extract a more precise object, we need to remove the shadow from object mask. The main idea to remove shadow tries to classify the object pixels into real moving objects or shadow [12-14]. Based on the mathematics analysis model in [12], we develop a new shadow elimination algorithm to remove shadow. Before introducing our shadow elimination algorithm, we need to explain the principle of color reflection. The principle of color reflection can be modeled as the multiplication of light energy and reflectance of an object. This principle can be expressed as the

following equation.

$$I^C(i, j, t) = E^C(i, j, t) * R^C(i, j, t), \quad (10)$$

where $I^C(i, j, t)$ is the value of color C of the pixel (i, j) at time t ,

$E^C(i, j, t)$ is the C light energy, $R^C(i, j, t)$ is the reflectance of color C ,

and C is red, green or blue channel.

For frame objects, we use the change detection based algorithm to obtain moving and static objects. However, shadow is always extracted accompanying real moving objects. Thus, we can classify the current frame objects into three classes, real moving object, shadow of moving object and static background object. Since the color of real moving object is unknown, we neglect analyzing the color of moving object. Now, we use the Eq.(10) to model the background objects and shadows in current frame as shown in Eq.(11) and Eq.(12), respectively. The subscript B and S are used to distinguish background object and shadow. From the Eq.(4), we generate a background image. Thus, we also can use the Eq.(10) to model background image as in Eq.(13).

$$I_B^C(i, j, t) = E_B^C(i, j, t) * R_B^C(i, j, t) \quad (11)$$

$$I_S^C(i, j, t) = E_S^C(i, j, t) * R_S^C(i, j, t) \quad (12)$$

$$BI^C(i, j, t) = BE^C(i, j, t) * BR^C(i, j, t) \quad (13)$$

Now, we consider the relationship among Eq.(11), (12), (13). We can assume the background object is stationary, so the reflectance of background object and shadow in current frame is equal to the reflectance of object in background frame. Moreover, if there is no light changing, the light energy of the background object in current frame is equal to the light energy of the object in background frame. Since the light

energy of shadow is covered by moving object, the energy of the shadow is decreased. Therefore, the difference of light energy between background object and shadow is a value K . Thus, we have the following relationship.

$$R_B^C(i, j, t) = R_S^C(i, j, t) = BR^C(i, j, t) \quad (14)$$

$$E_B^C(i, j, t) = E_S^C(i, j, t) + K = BE^C(i, j, t), \quad 0 < K \leq BE^C(i, j, t) \quad (15)$$

Based on the observation from the Eq.(11) to Eq.(15), if we substitute Eq.(14), (15) into Eq.(11), (12), (13), we have,

$$\frac{I_B^C(i, j, t)}{BI^C(i, j, t)} = 1 \quad (16)$$

$$\frac{I_S^C(i, j, t)}{BI^C(i, j, t)} = 1 - \frac{K}{BE^C(i, j, t)} \quad (17)$$

From Eq.(17), we know that if a pixel belongs to shadow, and then the value of Eq.(17) will be lower than 1. According to the Eq.(16) and Eq.(17), we develop a shadow elimination algorithm as shown in Fig.5 to remove shadow where th_1 and th_2 are the predecided thresholds.

Input: The pre-extracted object masks.

Output: Object masks without shadow.

Procedure:

1. For each pixel(i, j) of object mask at time t ;
2. Do
 - i. If $th_1 \leq BI^C(i, j, t)/I^C(i, j, t) \leq 1 + th_2$, then remove pixel (i, j) from current object mask
 - ii. Else do nothing

Fig. 5. Shadow Elimination Algorithm

After shadow elimination, shadow is removed from object mask. However, the

object mask still contains a lot of noises and the object boundaries are not smooth. Thus, further filtering of noise is required.

2.2.6 Morphological Operation

To smooth the object boundaries and remove the noises, two kinds of morphological operations are frequently used [8, 17]. The closing operation is first used to fill the black holes inside the object masks and the opening operation is then used to remove the small noises that do not belong to a moving object. In our algorithm, the structure element of size 7×7 and 5×5 are selected for closing and opening operations, respectively. In most of the cases, the smaller camera noises can be successfully filtered. However, larger regions caused by ghost effect are hard to remove out. Although larger structuring element may help, the computation cost is too expensive. Thus, instead of using larger structuring element, we will filter out these ghost regions in the video object tracking algorithm with temporal filtering.

After the morphological operations, the object mask is smoothed and indicates the shapes and the positions of all the moving objects in the current frame. The individual object in the object mask is then extracted in the next process.

2.2.7 Connected Component Labeling Algorithm and Size

Filtering

Since object mask just simply indicates the positions and the shapes of all the moving regions without separate information, thus, each individual object in the object mask must be extracted and assigned an identifying label. The connected component algorithm is a frequently used algorithm [8] to achieve this work. For every pixel, it first examines the neighboring pixels and assigns that pixel a label.

Then, pixels with the same label or equivalent labels are clustered together to form an isolated object.

Because some large noise and ghost regions are hard to be completely removed out, the size filtering must be performed after the labeling process. The size filtering process filters out those regions which are smaller than a pre-defined threshold. The objects not filtered out are called the object-of-interests and are tracked in the video tracking module.

2.3 Video Object Feature Extraction

From the previous module, we obtain the current objects which are the objects in current frame. However, we do not know any content included in these object. Thus, we need to extract the current object features. From the past research, there are many features, such as, motion vector, texture, shape, color histogram and so on, can be used to describe objects. Actually, the more meaningful features are extracted; the object description will be more accurate. Due to real time requirement, we can only choose a few of features to describe objects. Through the strict screening, we require those features can be used to describe (a) the position of object, (b) the color of object and (c) the shape of object. As the reasons mentioned above, we select the following four features: (i) center of object, (ii) color histogram, (iii) variance of object and (iv) major and minor axis of best-fit-ellipse.

For the first feature, we use the center of object to indicate the position of an object. We calculate the center (C_x, C_y) of a moving object by using Eq.(18), where R is the region of moving object and, N is the pixel number of the region R .

$$\begin{aligned}
C_x &= \frac{1}{N} \sum_{(x,y) \in R} \sum x \\
C_y &= \frac{1}{N} \sum_{(x,y) \in R} \sum y
\end{aligned} \tag{18}$$

For the second feature of color histogram, we divide each color channel into 16 bins instead of using a full color. Such a decision is based on two reasons. The first reason is full color information takes too much memory. For example, we need to use $256*256*256*4$ (Byte/object) = 64 MB/object to describe an object by using full color feature. The second reason is slightly light changing cause the object color changing. We use Eq.(19) to calculate the color bin of a moving object where r_i is the value of i^{th} red color bin and initialize to zero; $I^R(i, j, t)$ is the red color value of the pixel (i, j) at time t . Similarly, we use the same method to calculate green color bin (g_i) and blue color bin (b_i).

$$r_i = \begin{cases} \text{No Change} & \text{if } \text{floor}(I^R(i, j, t)/16) \neq i \\ r_i \text{ increase by one} & \text{if } \text{floor}(I^R(i, j, t)/16) = i \end{cases} \tag{19}$$

Similarly, for g_i and $b_i, 1 \leq i \leq 16$.

For the third feature of object variance, we use the x,y variance of an object to describe the spatial distribution of an object. The Eq.(20) is used to calculate the variance (V_x, V_y) of a moving object where R is the region of the moving object, N is the pixel number of the region R and (X_c, Y_c) is the center of object.

$$\begin{aligned}
X_V &= \frac{1}{N} \sum_{(x,y) \in R} \sum (X_c - x)^2 \\
Y_V &= \frac{1}{N} \sum_{(x,y) \in R} \sum (Y_c - y)^2
\end{aligned} \tag{20}$$

Since the third feature is lack of shape description, we select the fourth feature. We use a best-fit-ellipse [15] to approximate the shape of the object instead of using a bounding box. This is because the object bounding box will change rapidly while the posture of an object just changes a little bit. For example, the comparison between bounding box and best-fit-ellipse is shown in Fig.6 and Fig.7. The increasing width of bounding box is about 200% and too much space is not belongs to object. Reversely, the increasing width of best-fit-ellipse is about 150% and the non-belongs space is smaller than bounding box. Due to the reason, we use best-fit-ellipse instead of using bounding box. From the Fig.7, we can know that, a best-fit-ellipse is not fully wrapped up the moving object. The reason is that the best-fit-ellipse tries to find an ellipse whose second geometrical moment is equal to that of the object.

As the above reasons, we select the best-fit-ellipse as the fourth feature. For the fourth feature, the calculation is more complex. From the [15], we obtain the value J_{min} and J_{max} where J_{min} and J_{max} are the least and greatest moments of inertia for the ellipse. After that, we substitute J_{min} and J_{max} in Eq.(21). In Eq.(21), a and b are the length of ellipse axes. At last, Eq.(22) is used to assign the value of major and minor axis.

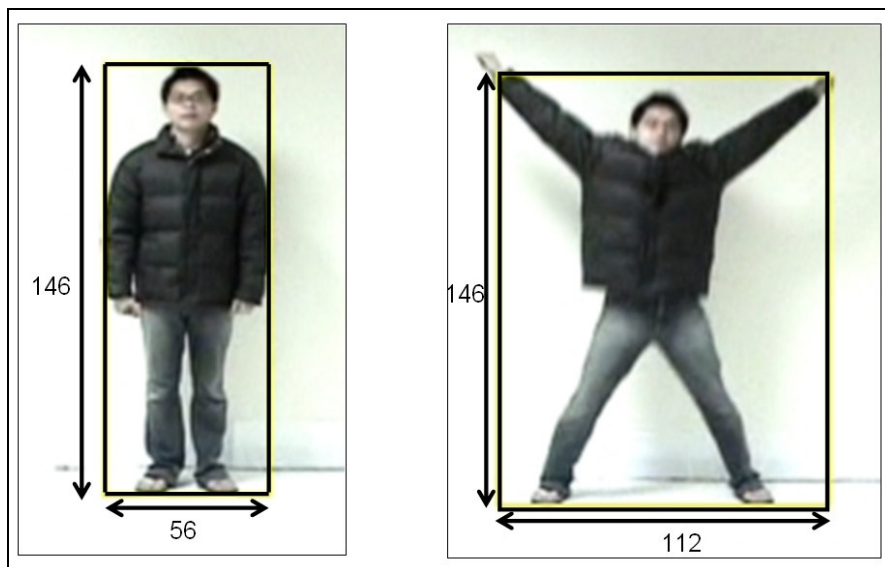


Fig. 6. The increase of width of bounding box is 200%

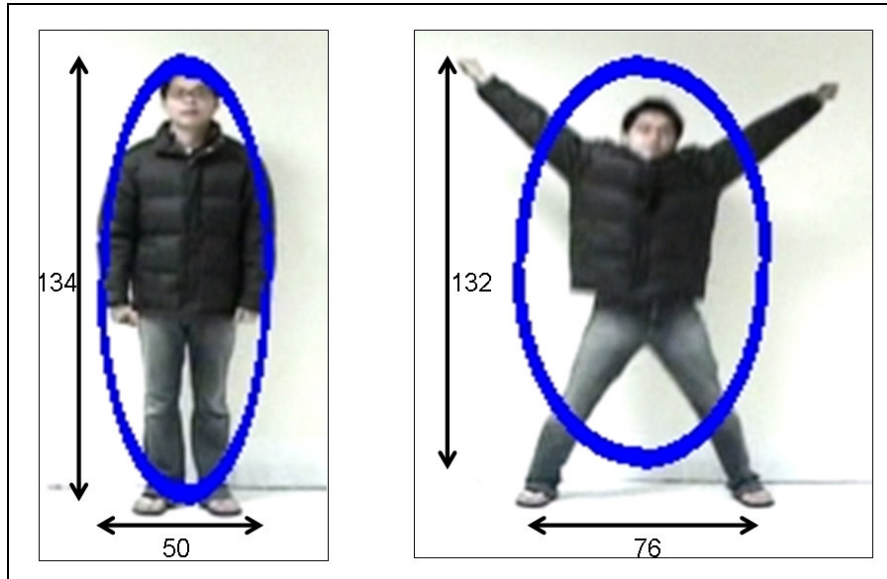


Fig. 7. The increase of width of best-fit-ellipse is about 150%

$$a = 2 * \left(\frac{4}{\pi}\right)^{\frac{1}{4}} \left[\frac{(J_{max})^3}{J_{min}}\right]^{\frac{1}{8}}$$

$$b = 2 * \left(\frac{4}{\pi}\right)^{\frac{1}{4}} \left[\frac{(J_{min})^3}{J_{max}}\right]^{\frac{1}{8}}$$



(21)

$$major = \begin{cases} a & \text{if } a \geq b \\ b & \text{if } a < b \end{cases}$$

$$minor = \begin{cases} a & \text{if } a < b \\ b & \text{if } a \geq b \end{cases}$$

(22)

Chapter 3

Video Object Tracking Algorithm

Video object tracking is an important module in our proposed system. We utilize the tracking information to analyze the behavior of human. In this chapter, we introduce the related work of video object tracking in Section 3.1 and the detailed method of tracking algorithm is proposed in Section 3.2. In Section 3.3, we point out some drawbacks in our proposed tracking algorithm.

3.1 Related Work of Video Object Tracking

Video object tracking is an important and frequently discussed research topic. Its objective is to match the detected objects in the current frame to the corresponding objects detected previously. The tracked position and the shape of objects can be used to form the object trajectories for later human behavior analysis.

The object tracking algorithms first take the detected object masks from the segmentation algorithms as input data, and try to match the objects detected earlier using features such as position, shape and color. For example [16], Oberti et al. use the shape of the object corners to track video objects. Some tracking algorithms also take motion information into consideration. For example, Kim et al. uses the direction of the motion and the variation of the speed to compute smoothness feature as the matching criteria [17]. Chen uses the motion as the constraint to find matching objects [18]. Some other algorithms [19-22] adopt Kalman Filtering. It is a linear estimation process that estimates the current value and updates the prediction recursively to estimate and track the position of the objects.

The precision of the prediction involves two errors: the processing error and the measurement error. Because sometimes there are errors in the segmentation process due to the cluttered scene, the object masks would not be very accurate and hence the measurement errors would be large. Besides, some abrupt movements of the objects such as waving of hands will make the processing error large. The prediction error may not converge quickly if both the processing error and measurement errors are large. Thus, it may be difficult to match correctly due to the uncertainty of the prediction. In this thesis, instead of using Kalman Filtering, our algorithm uses the color and shape as features. The occlusion and the split of objects can also be handled in our tracking algorithm.

3.2 Video Object Tracking

Tracking is a difficult problem in video surveillance system due to tracking objects might be occluded. To solve this problem, many ideas have been proposed [16-22]. Generally, tracking methods can be classified into two categories. One category [19-21] estimates the motion of the objects and minimizes the error function to track the objects. The other category finds the similarity of current objects and previous objects and maximizes the similarity measure to track the objects. We choose the similarity measure method to track the objects in this system. The detailed reasons will be explained in Section 3.2.1.

From the observation, the object occlusion can happen inside the camera view or outside the camera view. For the first condition, we can observe the occlusion of the objects. For the second condition, we can not observe the occlusion of objects except we can segment the occluded objects or the occlusion objects will split in camera view. Due to the two different conditions, Jung [19] defines the first condition as EXPLICIT

OCCLUSION and the second condition as IMPLICIT OCCLUSION. In the cases of explicit occlusion, the occlusion of objects can be detected and its trajectories can be reconstructed. However, in the cases of implicit occlusion, the splitting of objects can be detected but the trajectory of objects is hard to reconstruct. We explain the problem in Section 3.3.

Although the object-level information, such as color, shape, can be extracted via the segmentation of video objects, the higher level object semantics can be extracted from the object trajectories. Thus the object tracking process is the key role toward the human behavior analysis. Our tracking module gets the extracted object features as the input and tracks all the objects to get the object trajectories. The flow chart of tracking algorithm is shown in Fig.8. In Fig.8, we use the single object matching algorithm to find all the matches of the single objects from previous segmented object to current segmented object. If we find a match, we update both the current and previous object trajectory. The details of single object matching algorithm are presented in Section 3.2.1.

After single object matching algorithm, we classify the remaining unmatched objects into four categories. The first class is the objects occlusion in the current frame. We define these objects as the current merge objects. The second class is the objects split in the current frame. We define these objects as current split objects. The third class is the objects disappear in previous frame or in current frame. The fourth class is the new objects appear in current frame. Due to the unmatched condition, we develop a multiple objects matching algorithm to detect current merge and split objects.

In the multiple objects matching algorithm, we generate all the combination of objects and call these temporal object virtual objects. If we find a match in these virtual objects, then we conclude the detection of a current merge object or current

split object. For example, if we want to detect the current merge objects, we generate all the combination of the possible candidates in previous frame. For each combination object, or virtual object, if we find a match between this virtual object and current object, then we conclude the detection of a current merge object. The details of multiple objects matching algorithm are presented in Section 3.2.2. If we can not find a match under multiple objects matching algorithm, then we believe the remaining objects could be disappeared or new objects. The complete flow chart of tracking algorithm is presented in Section 3.2.3.

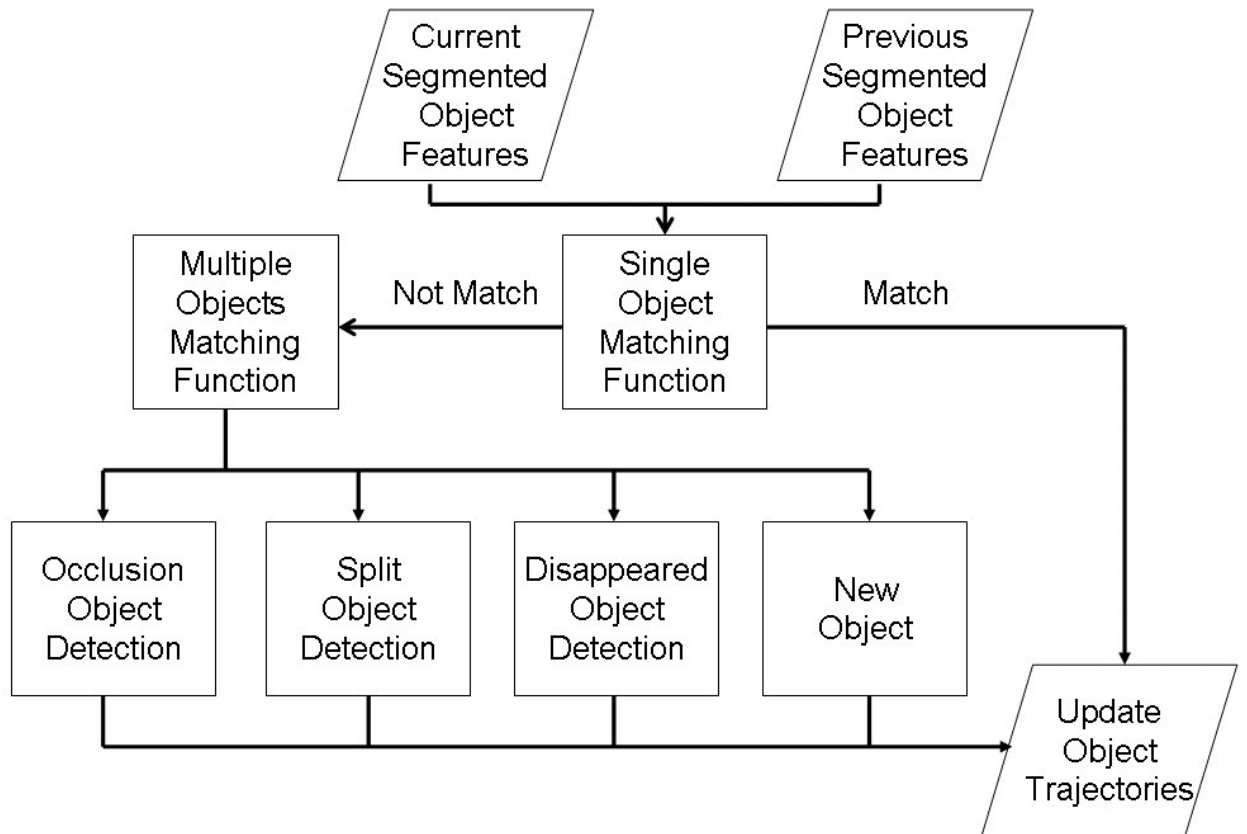
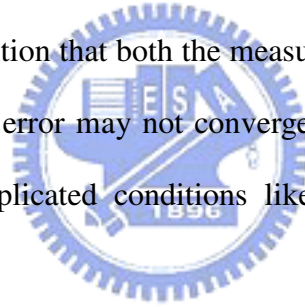


Fig. 8. The flow chart of tracking algorithm

3.2.1 Single Object Matching Algorithm

The object trajectories can be obtained by matching the current video objects

with the previously tracked video objects. In the literature of object tracking, some algorithms [19-21] adopt the Kalman Filtering to estimate and track the objects. It is appealing because it recursively estimates the object states and updates the predictions. In the ideal situation, when the moving paths are very smooth and the object masks are very accurate, the prediction error converges quickly because both the measurement error and the processing error are small. However, the detected object boundaries may contain some errors due to the clutter scenes in the real environments, and the measurement errors thus become large. In addition, the path of a moving object may not be as smooth as expected. For example, if we connect the mass centers of a walking person, the connected path looks like zigzag rather than a straight line because all the actions such as waving of hands and striding affect the mass centers significantly. Under this condition that both the measurement error and the processing error are high, the prediction error may not converge quickly. Thus, it is difficult to track and handle some complicated conditions like object occlusions due to the uncertainty of the estimation.



In our single object matching algorithm, we match an object in previous frame to an object in current frame by using a score function. The score function is stated in Eq.(23). In the score function, we use the color histogram, object center and object major, minor axis extracted in previous module to calculate the similarity between objects. Since the color information is a very important feature during matching, we assign 70% weight to color similarity. The rest of weight is assigned to shape similarity.

The *ColorSimilarity* function to calculate the color similarity between two Objects (O_1 and O_2) is shown in Eq.(24) where r_i^1 and r_i^2 are the value of i^{th} red color bin for O_1 and O_2 respectively and similarly for g_i^1, g_i^2, b_i^1 and b_i^2, N_1 and N_2 are the number of pixels for O_1 and O_2 . The value of *ColorSimilarity* function lies between 0

and 1.

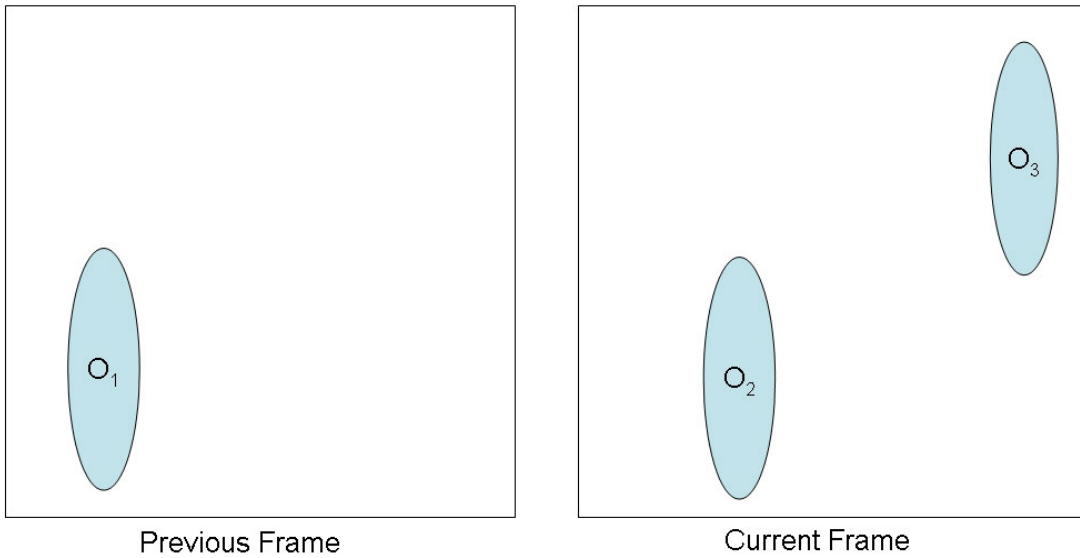
The *ShapeSimilarity* function to calculate the shape similarity between two Objects (O_1 and O_2) is shown in Eq.(25) where $major_1$ and $major_2$ are the length of major axis for O_1 and O_2 , respectively, similarly, $minor_1$ and $minor_2$ are the length of minor axis for O_1 and O_2 . The value of *ShapeSimilarity* function lies between 0 and 1.

$$Score(O_1, O_2) = 0.7 * ColorSimilarity(O_1, O_2) + 0.3 * ShapeSimilarity(O_1, O_2) \quad (23)$$

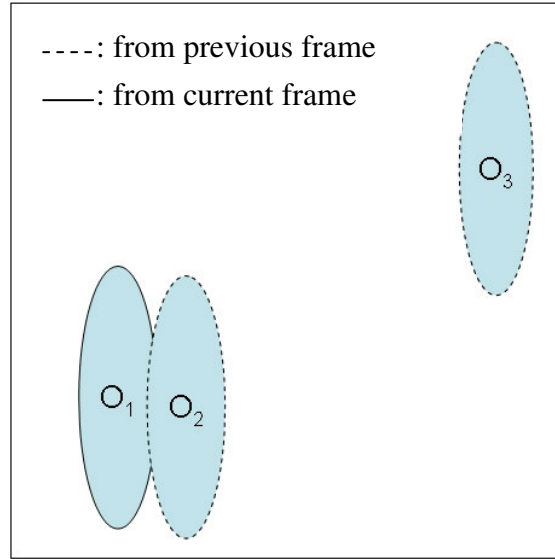
$$ColorSimilarity(O_1, O_2) = 1 - \frac{1}{3 * (N_1 + N_2)} * \sum_{i=0}^{15} (|r_i^1 - r_i^2| + |g_i^1 - g_i^2| + |b_i^1 - b_i^2|) \quad (24)$$

$$ShapeSimilarity(O_1, O_2) = 1 - \frac{1}{2} * \left(\frac{|Major_1 - Major_2|}{\max(Major_1, Major_2)} + \frac{|Minor_1 - Minor_2|}{\max(Minor_1, Minor_2)} \right) \quad (25)$$

Since the score function in Eq.(23) only calculates the similarity of color and shape, the matching might be incorrect. For example in Fig.9(a), the match O_1 to O_3 is not correct although the color and shape is similar. Since the object O_1 is closer to O_2 than O_3 as shown in Fig.9(b), thus, O_1 match with O_2 is a better match than O_3 . Based on the reason mentioned above, we use a distance threshold to eliminate the impossible candidate.



(a). Three object O_1 , O_2 and O_3 in current frame and previous frame



(b). Merge current and previous frame into one frame

Fig. 9. A matching example for three object O_1 , O_2 and O_3

Before introducing our single object matching algorithm, we must point out a critical problem in scoring. Consider the case in Fig.10 where P_i and C_j are the previous i^{th} object and current j^{th} object, respectively, and the numeric value are the matching score between P_i and C_j . We use greedy algorithm to find the matching and the answer is shown in Fig.10(a). We assume that the optimal solution is the largest summing score. The answer in Fig.10(a) is not an optimal solution while the optimal solution is shown in Fig.10(b). However, to find the optimal solution is a NP-Complete problem.

Since the greedy algorithm always miss to find an optimal solution, we develop an N-best algorithm to find a better answer in an $N \times N$ scoring table. The main idea of our N-best algorithm finds the first N-best candidates in each matching. We illustrate the idea of N-best algorithm in Fig.11 by using the example in Fig.10. In Fig.11, we maintain a 3×1 and a 3×3 array for recording the 3-best summing score and recording the traced path, respectively. The first iteration shown in Fig.11(a), finds 3-best matching from P_1 , P_2 and P_3 to C_1 . During the second iteration, shown in Fig.11(b),

we check the 3x3 matrix to prevent selecting a same path from P_1 , P_2 and P_3 to C_2 and for each possible path sum all the score. Since we only maintain 3 high scores, we delete other scores and corresponding traced path from Fig.11(b) and the result is shown in Fig.11(c). Analogously, we find a non-repeated path from P_1 , P_2 and P_3 to C_3 and sum the score of path in last iteration. The highest score in Fig.11(d) is the final answer whose corresponding path is P_3, P_2, P_1 meaning that P_3, P_2, P_1 match with C_1, C_2, C_3 respectively. The complete algorithm to calculate the single object matching algorithm and the N-best algorithm are stated in Fig.12(a) and Fig.12(b), respectively.

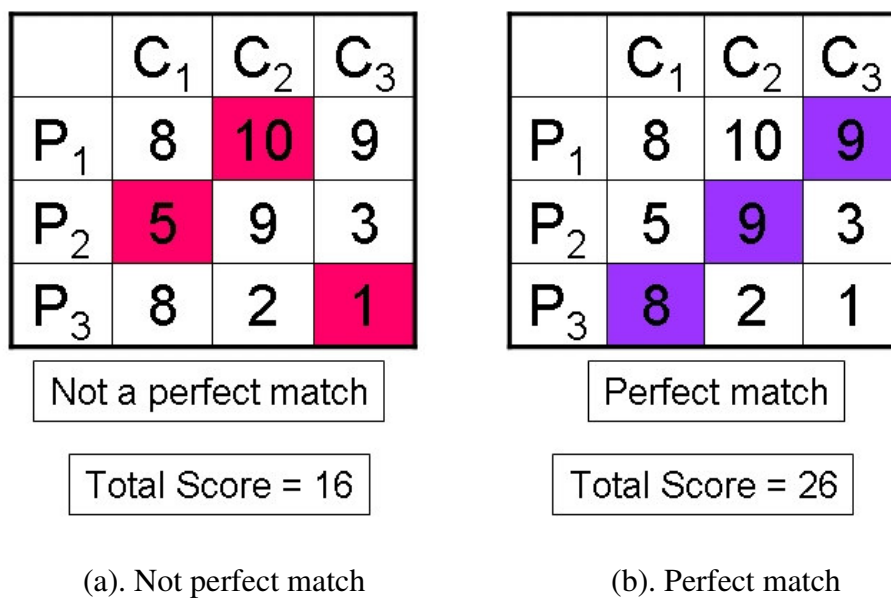
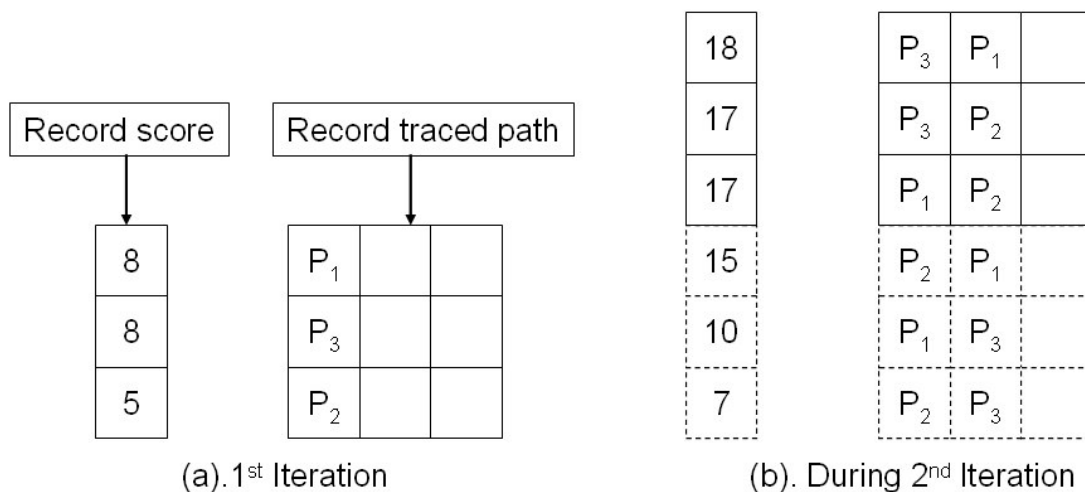


Fig.10. The perfect matching



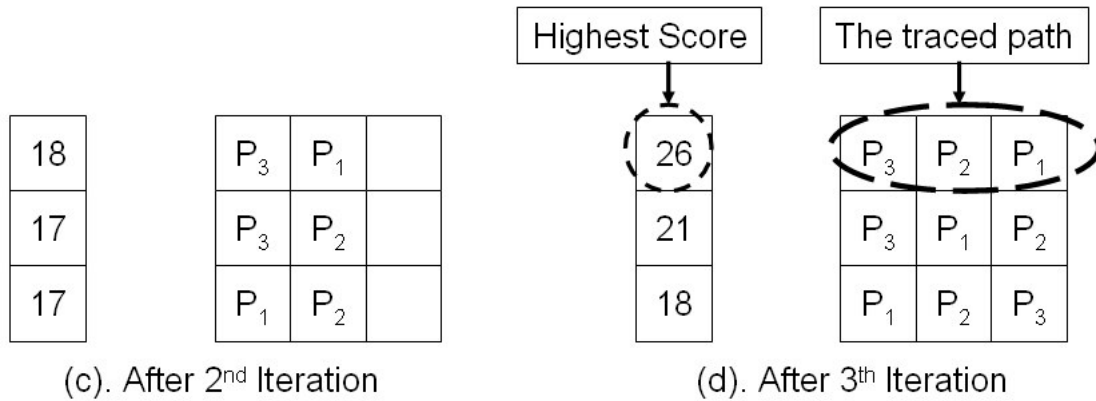


Fig.11. Illustration of N-best algorithm

Input: The $N \times N$ score matrix M .

Output: The matching for current frame objects C_1, C_2, \dots, C_N

Procedure:

S: An $N \times 1$ matrix, used to record the summing score

T: An $N \times N$ matrix, used to record the traced path

U: An $N \times N$ matrix, a temporally buffer for sorting

V: An $N \times 1$ matrix, a temporally buffer for sorting

Initial S, T to 0

For $i = 1$ to N

 Initialize U, V to 0

 For $j = 1$ to N

 For $k = 1$ to N

 //calculate the summing score

$x = S[j] + M[k][i]$

 //if the path haven't traced

 If $T[j][k] = 0$

 //sorting the summing score and corresponding traced path

 For $m = 1$ to N

 If $x > V[m]$

 Insert x before $V[m]$ and insert a row before $U[m]$

 Delete $V[N+1]$ and delete the row $U[N+1]$

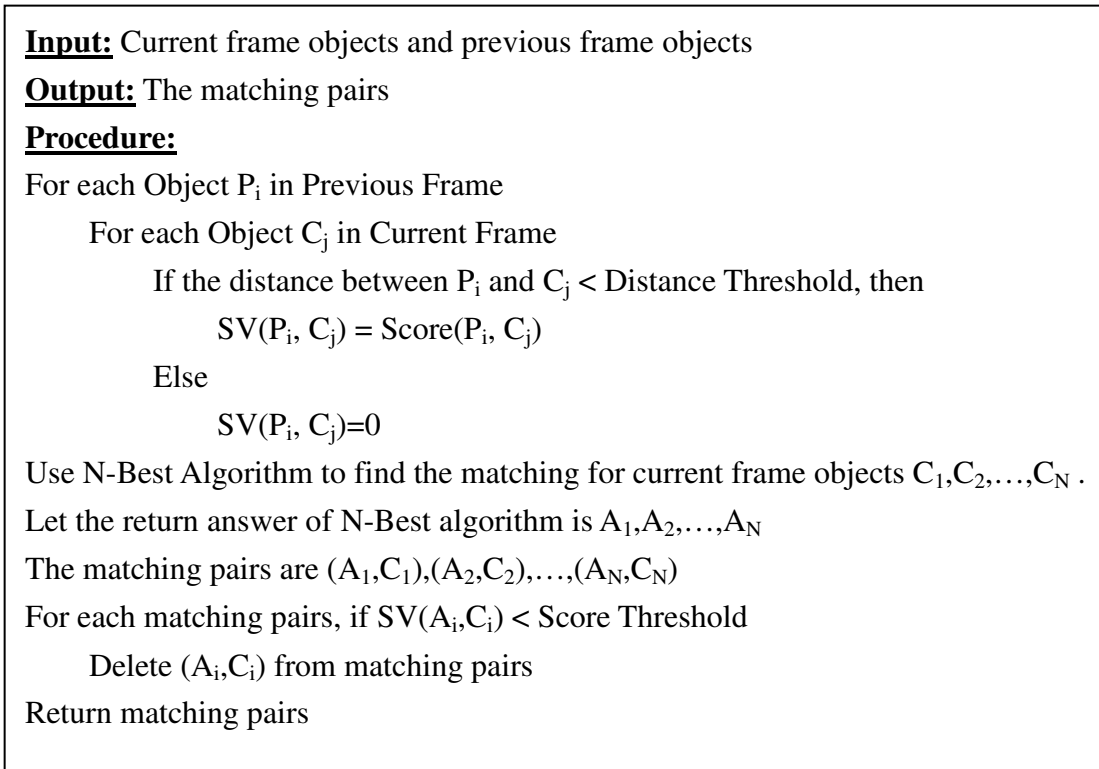
 For $n = 1$ to N

$U[m][n] = T[j][n]$

$U[m][k] = i$

 break

(a). N-best algorithm.



(b). The Single Object Matching Algorithm

Fig. 12. The N-best algorithm and the single object matching algorithm

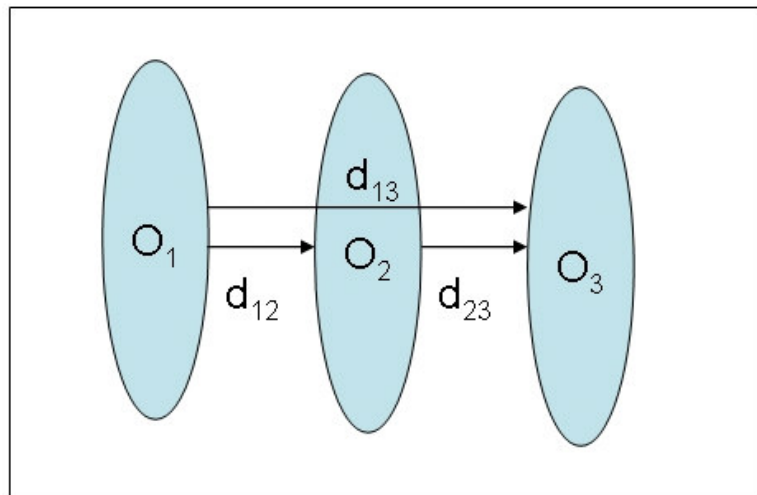
3.2.2 Multiple Objects Matching Algorithm

The single object matching algorithm only matches the single objects in previous frame with the single objects in current frame. Since the current merge objects and current split objects are different to single objects, thus the single object matching algorithm fail to find a match. Due to the reason, we develop a multiple objects matching algorithm to find the current merge object and current split object. The main idea is to guess the possible candidate of current merge object and current split object. Before introducing the multiple objects matching algorithm, we explain the concept of transitive closure to eliminate the impossible candidates. First, we define a relation $R_i(O_1, O_2)$ to indicate the possibility of collision between objects O_1 and O_2 at i^{th} frame. If the distance between O_1 and O_2 is less than a *collision_threshold*, then we set $R_i(O_1, O_2)$ to be 1. Otherwise, we set $R(O_1, O_2)$ to be 0. The relation $R_i(O_1, O_2)$ can be

expressed as in Eq.(26).

$$R_i(O_1, O_2) = \begin{cases} 1 & \text{if } distance(O_1, O_2) \leq collision_threshold \\ 0 & \text{if } distance(O_1, O_2) > collision_threshold \end{cases} \quad (26)$$

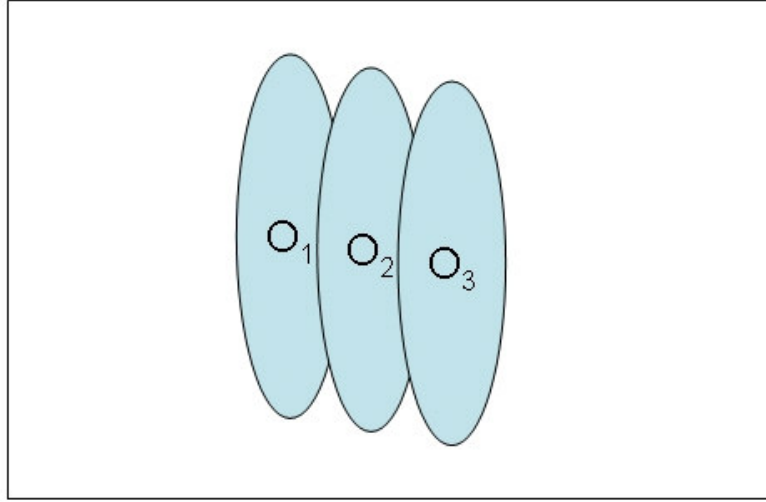
Second, consider the three objects in Fig.13(a), and the tabular form shown in Fig.13(b). If object O_1 and O_3 move toward O_2 , and then we have the result in figure Fig.13(c). Now, consider the relationship between possibility collision relation R_i and the possibility occlusion relation T_i in the i^{th} frame. For example in Fig.13, the transitive closure of the possibility collision relation is the possible occlusion relation. It is not difficult to find that the possible occlusion relation T_i is equal to transitive closure of relation R_i . Thus, the relation T_i is equivalence relation. The elements in an equivalence class of T_i are the possible occlusion set. Therefore, we can use the relation R_i to reduce the possible occlusion candidate.



(a). Previous frame

R	1	2	3
1	1	1	0
2	1	1	1
3	0	1	1

(b). The relation in tabular form (Assuming $d_{12}, d_{23} \leq d, d_{13} > d$)



(c). Current frame

Fig. 13. The relationship between possible collision and possible occlusion

After finding the possible occlusion candidates, we use these candidates to generate all possible occlusion objects as virtual objects and accumulate these objects color bin value. After generating the virtual objects and its color, we use the score function in Eq.(27) to calculate the similarity.

$$Score(O_1, O_2) = ColorSimilarity(O_1, O_2) \quad (27)$$

Since it is hard to predict the shape of occlusion objects, we only use the color

information to calculate the score. Finally, we state the multiple objects matching algorithm in Fig.14.

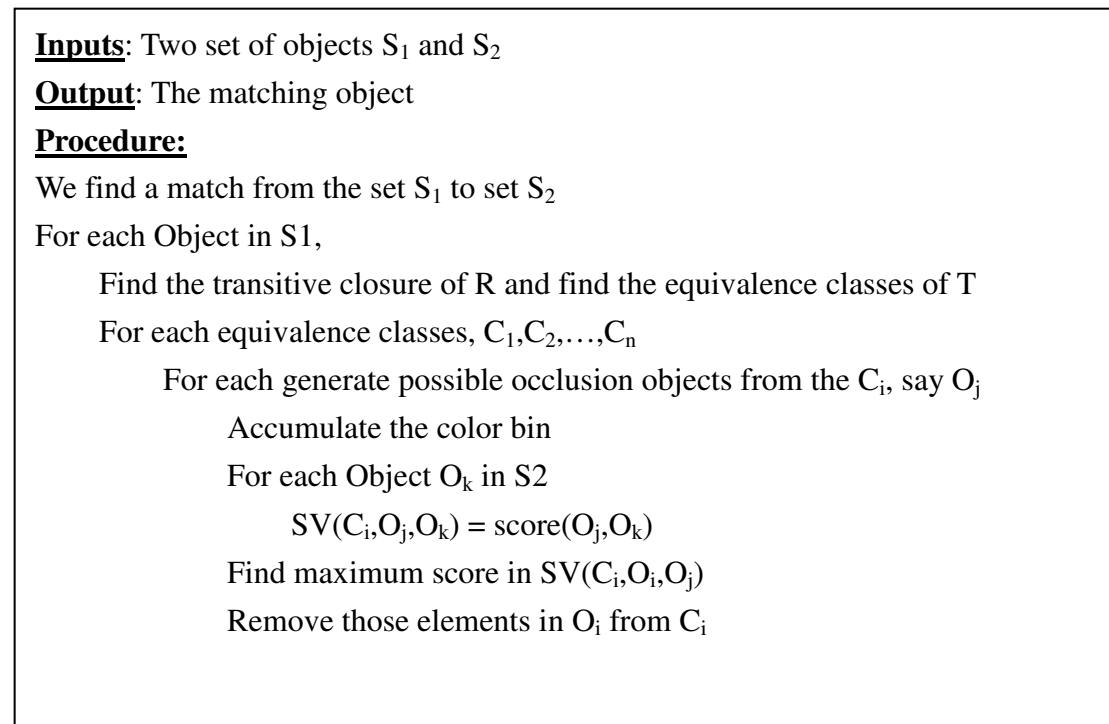


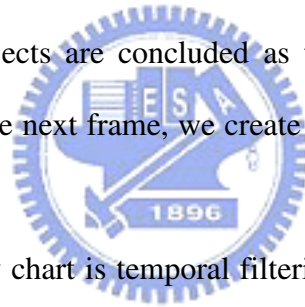
Fig. 14. The multiple objects matching algorithm

Similarly, to find a best matching is an NP-Complete problem. We use greedy approach in multiple objects matching algorithm instead of using N-best algorithm because the N-best algorithm is too complex to solve this problem.

3.2.3 Objects Tracking Algorithm

In this section, we briefly explain the tracking module in our system. The flow chart of our tracking algorithm is shown in Fig.15. First, we use the single object matching algorithm to find the match between the previous segmented objects and current segmented objects. If we find a match, we update both the objects trajectories. Otherwise, we pass the remaining unmatched objects to the multiple objects matching module. The first step of multiple objects matching algorithm finds the current merge object. Thus, we generate all the possible virtual objects from previously remaining

objects. If we find a match between those virtual objects and current objects, then we obtain the current merge objects and update both the objects trajectories. Otherwise, we pass the remaining unmatched objects to the next step of multiple objects matching algorithm. In order to find the current split object, we generate all the possible virtual objects from current remaining objects. If we find a match between those virtual objects and previous objects, then we find the current split objects and update both the objects trajectories. After multiple objects matching algorithm, we believe those remaining objects are the disappeared objects or new objects. To distinguish the disappeared objects from new objects, we find the match between the last n frames objects and those remains objects. If we find a match, then we conclude that the object is temporally disappearance and update both the objects trajectories. Otherwise, the remaining objects are concluded as the new objects. For each new object, if it can be found in the next frame, we create a new entry for it. Otherwise, it is treated as a noise.



The last step in the flow chart is temporal filtering to filter out the ghost effect. Because ghost usually appears and disappears very quickly, we can use the temporal filtering to filter out the ghost objects. In our algorithm, we don't think a detected and tracked object valid unless it survives more than a certain time period. In other words, an object that goes to disappear too soon after it appears can not be deemed as an object and is filtered out. At last, we state the proposed tracking algorithm in Fig.16.

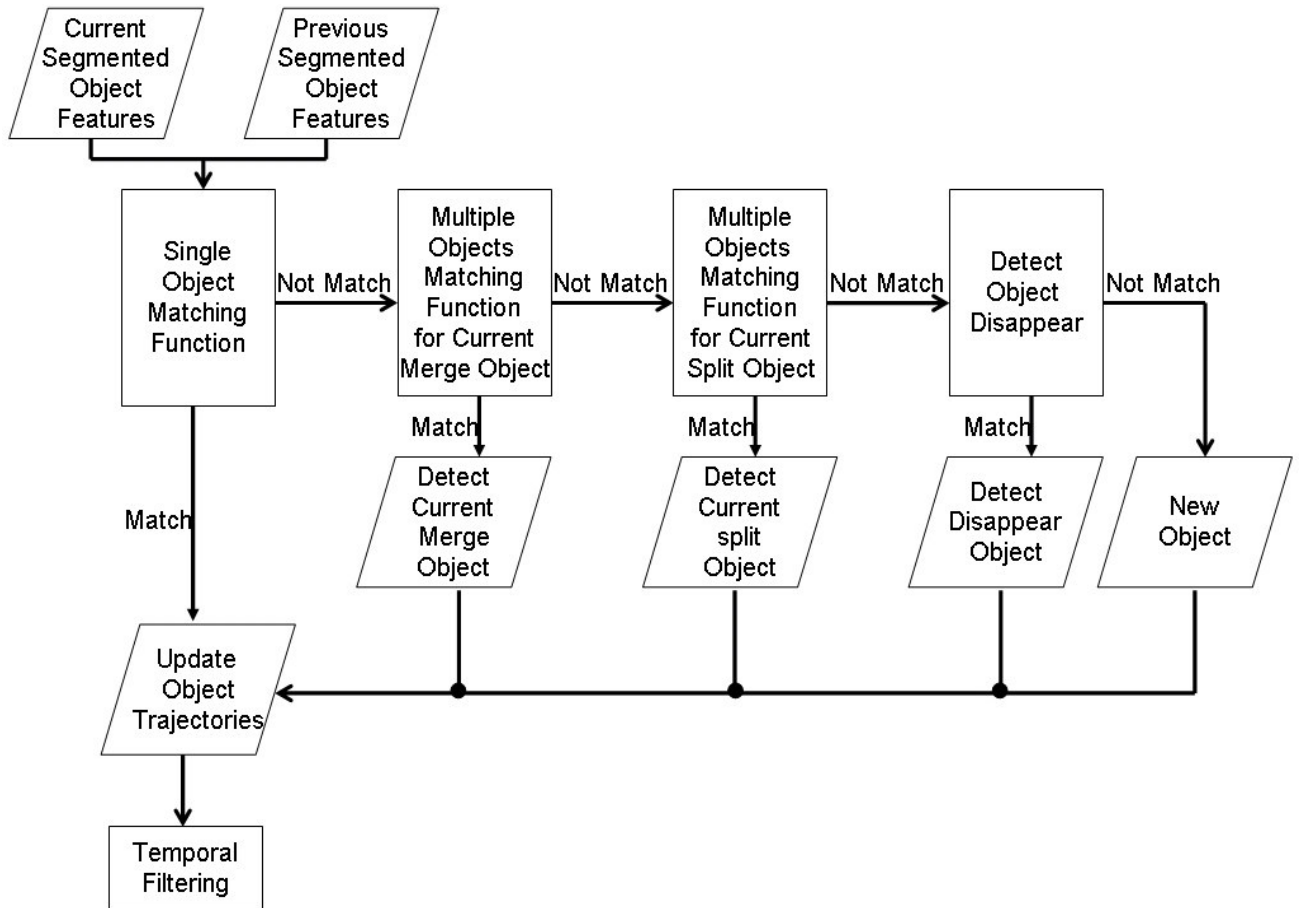


Fig. 15. The flow chart of tracking algorithm

Input: S_1 : The set of current frame objects
 S_2 : The set of previous frame objects
 U : The set of object trajectory

Output: The object trajectory set U

Procedure:

// Match the single object
 Use Single Object Matching Function with input S_1 and S_2 .
 Delete the matching pairs from S_1 and S_2 and form two new sets S_1' and S_2' , respectively
 Update the trajectory set U

// Detect the current merge object
 Use Multiple Object Matching Function with input S_1' and S_2' .
 Delete the matching pairs from S_1' and S_2' and form two new sets S_1'' and S_2'' , respectively
 Update the trajectory set U

// Detect the current split object
 Use Multiple Object Matching Function with input S_1'' and S_2'' .
 Delete the matching pairs from S_1'' and S_2'' and form two new sets S_1''' and S_2''' , respectively
 Update the trajectory set U

// Detect temporally disappear object
 For each element in set S_1''' , if we can find a match using Eq.(27) in the set U ,
 We remove this element from S_1''' a new set S_1'''' and update trajectory set U

// Detect a new object
 For each element in set S_1'''' , If we can find a match in the next frame,
 We add S_1'''' into trajectory set U .

Fig. 16. Tracking Algorithm

3.3 Problems in Object Tracking Algorithm

There are some drawbacks in the so far proposed algorithm and we explain in this section. Consider the situation in Fig.17. Occlusion object O_1 - O_2 splits at time t and the split object O_2 merge with object O_3 at the same time. If we assume occlusion object O_1 - O_2 is never split before time t , then our proposed tracking algorithm fail to track those objects. This problem occurs because the multiple objects matching algorithm can not generate a virtual object to match with remaining objects. To solve this problem, we need to segment objects out from the occlusion object. We leave this issue to be the future work.

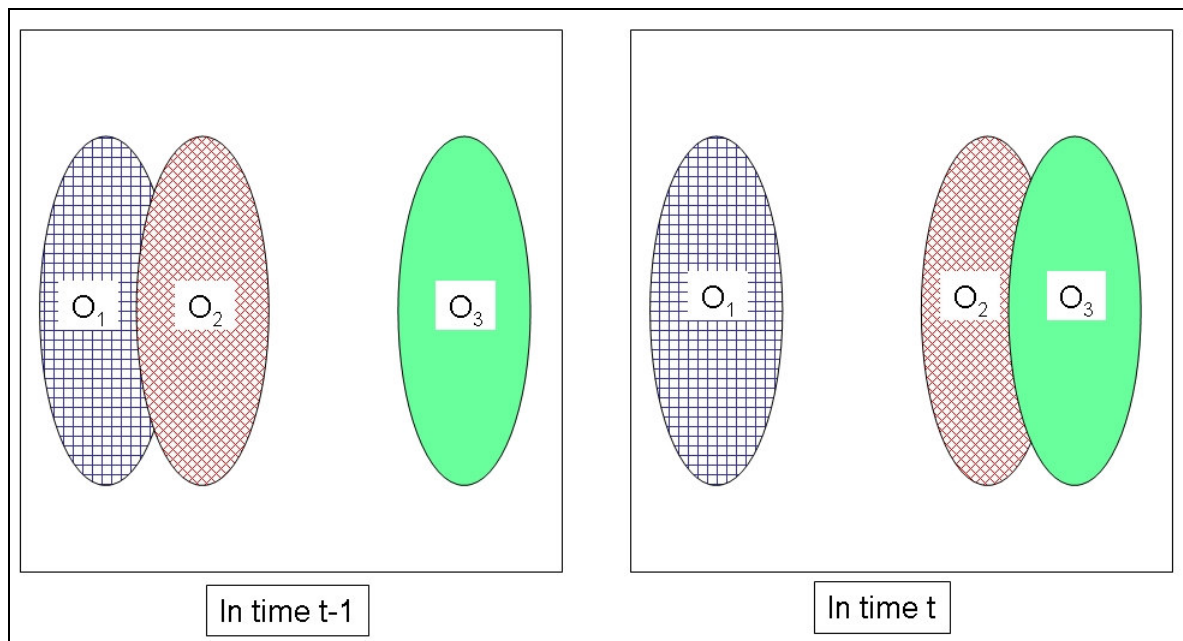


Fig. 17. Split and merge occur in same time

Another problem is depicted in Fig.18. Consider three consecutive frames, f_{t-2} , f_{t-1} , f_t , at time $t-2$, $t-1$ and t , respectively. The occlusion objects O_1 - O_2 and O_3 - O_4 in f_{t-2} merge at time $t-1$. By using our multiple objects matching algorithm, the merging can be detected. After that, the merge object split at time t , and form two split objects O_1 - O_3 and O_2 - O_4 . We also can use the multiple objects matching algorithm to detect

the splitting objects. However, the problem of how to update the object trajectories arises. One solution out of this problem is to segment out the objects from the occlusion object. We also leave this problem to be the future work.

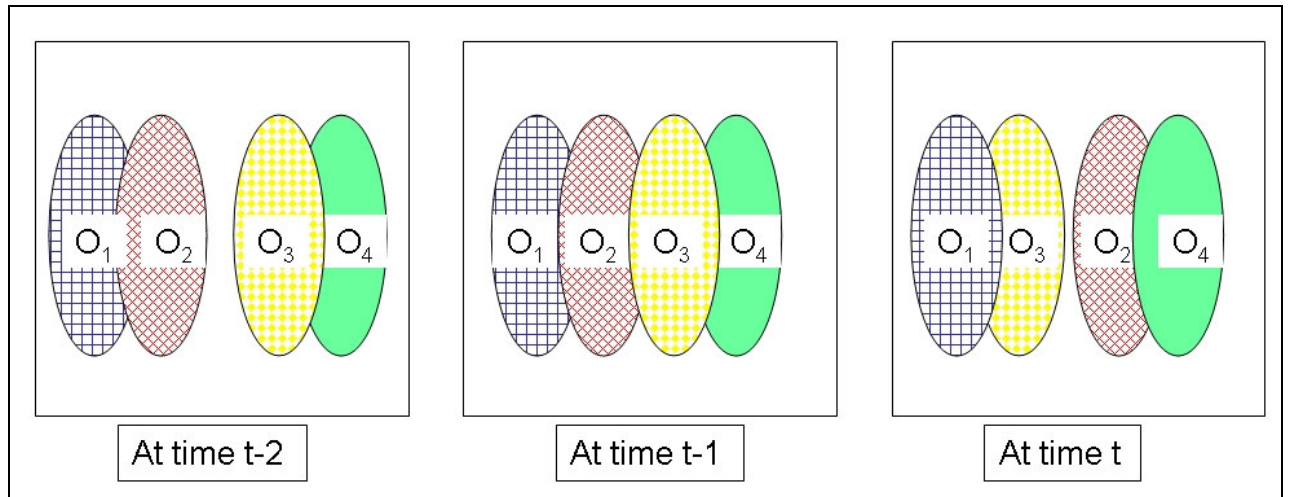


Fig. 18. The problem occurs in object trajectories updating.



Chapter 4

Human Behavior Analysis

The surveillance system performs a trajectory monitoring based on the hypothesis according to which situations at risk imply an abnormal behavior. Thus, to monitor people's behaviors over time trajectory feature is an essential cue. Behavior analysis is a high level understanding which requires the consideration of the activities (walking, sitting) under surrounding context. For examples: "sitting on the sofa" and "watching TV" are two kinds of behaviors which take place on specific location and spatial direction with a sequence of activities. Therefore, designing a general human behavior analysis system to adapt to various kinds of situation seems impossible except using a training model [28, 30, 32, 33, 40].

For the training model, the great drawback is the training data. If the training data can not fit the learning behavior, the accuracy of the system will be degraded. In addition, users must define what behavior he would like to train, which will cause trouble to users. Therefore, we adopt a non-training model to analyze human behavior.

In our system, we are focused on the analysis of abnormal human behavior because abnormal behavior of human is more meaningful in health care monitoring surveillance systems. Some example abnormal human behaviors are falling down, faint, fighting, etc. Among them, faint and falling down is the most important abnormal behavior. Thus, our system focuses on detecting faint and falling down. To distinguish faint, falling down and lying, we give the following definitions. Faint or falling down is the situation that a person changes his posture from standing to lying down quickly and never climbing up for a period of time. On the other hand, lying is

the situation that a person changes his posture from standing to sitting and then to lying down. Therefore, we analyze the three postures: sit pose, stand pose, and lie pose. To analyze faint behavior, we utilize those three postures and a finite state machine (FSM).

In this chapter, we introduce the related work of human behavior analysis in Section 4.1. After that, we introduce the detail method of analyzing the three postures (sit, stand, lie) in Section 4.2. In Section 4.3, we use the extracted postures to analyze the abnormal behavior.

4.1 Related Work of Human Behavior Analysis

There exist several researches on the human behavior analysis. Toshikazu Wada [23] use a state machine to track the human entering and exiting a room. Ismail Haritaoglu and David Harwood [24] develop a system called W4 for multiple object (human in their case) tracking and human body part labeling. The multiple target tracking is performed based on median coordinate and a temporal texture template, followed by the human posture recognition based on the projection histograms of the object. From the recognized posture, the body part of the human are labeled. Polana and Nelson [25] extract the cyclic motions and apply the template matching on their motion vectors to recognize human activity. Jezekiel Ben-Arie [26] uses the maximum likelihood algorithm to match a sequence of model shape images, coded by angles between body parts, for the recognition of the sitting, bending and standing. Among these approaches most are limited to only body posture recognition. The postures that could be recognized are restricted to very specific pictograph patterns. However, for applying human posture analysis in normal situation, the high variation of body part shapes tends to cause the recognition failure. Furthermore, they focus

more on low level to mid level of activity analysis, rather than the high level understanding. However, a high level understanding is related more with the existing context environment than the posture shapes.

4.2 Posture Analysis

Before introducing our posture analysis mechanism scheme, let us observe Fig.19. Fig.19(a), (b), (c) show a standing posture, sitting posture and lying posture, respectively. The x-variance V_x and y-variance V_y of a standing posture in Fig.19(a) are 1.09×10^6 and 1.09×10^7 , respectively. Similarly the V_x and V_y of a sitting posture in Fig.19(b) and of a lying posture in Fig.19(c) are 1.30×10^6 , 3.28×10^6 , 7.92×10^6 and 7.33×10^5 , respectively. From Fig.19, we can conclude that a standing person's y-variance V_y is much greater than x-variance V_x and a lying person's x-variance is much greater than y-variance. The sit pose falls between stand and lie.

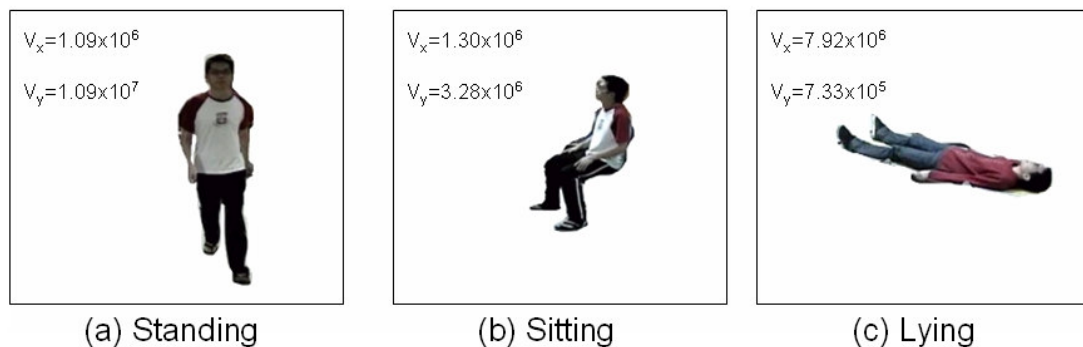


Fig. 19. Three posture and its x,y variance

Thus, we expect the following equation can be used to distinguish the poses sit, stand and lie.

$$P = \frac{V_x}{V_y} \quad (28)$$

For each posture, we use 300 pictures generated by 5 persons and calculate the P value in Eq.(28). The result of experiment is shown in Fig.20. Since the P value of standing concentrates in the range $[0, 0.3]$, we use logarithm value P instead of using value P . From the experiment, we conclude that the P value of stand pose is lower than $10^{-0.53} \approx 0.3$ and P value of lie pose is greater than $10^{0.52} \approx 3.3$. The P value between 0.3 and 3.3 is set to sit pose. We summarize the relationship between value P and three postures in Eq.(29). Thus, we use the equation Eq.(29) to distinguish sit, stand and lie postures.

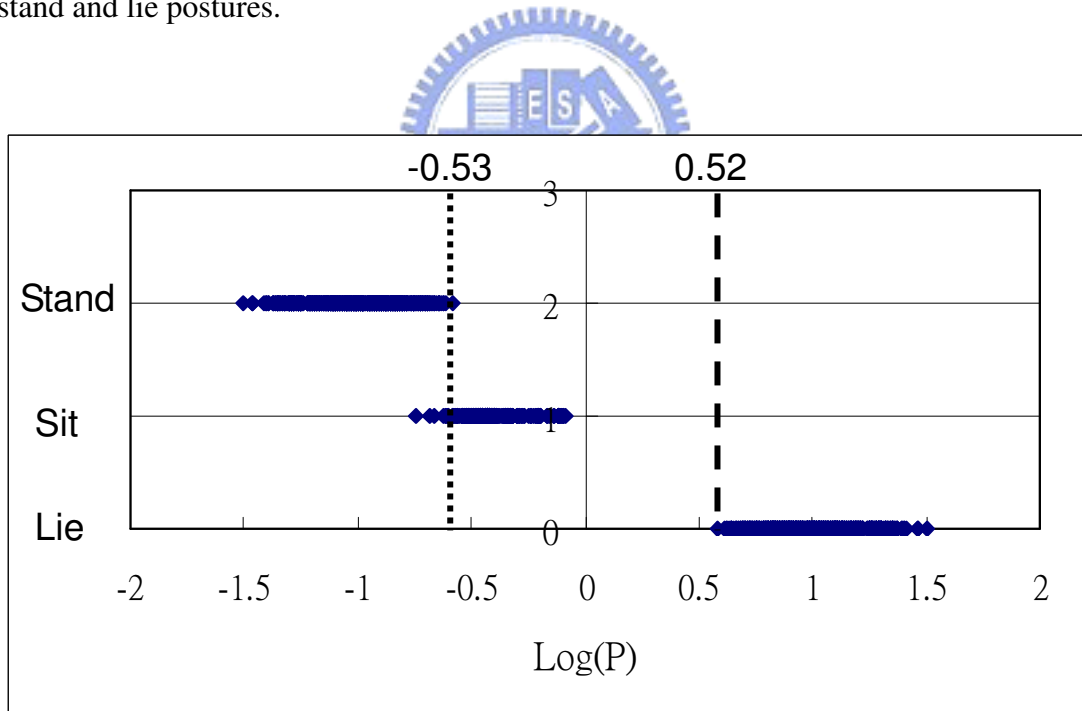


Fig. 20. The experiment of postures versus logarithm of P

$$posture = \begin{cases} stand & \text{if } P \leq 0.3 \\ sit & \text{if } 0.3 < P \leq 3.3 \\ lie & \text{if } P > 3.3 \end{cases} \quad (29)$$

4.3 Abnormal Behavior Analysis via Finite State Machine

In this section, we present the mechanism method to analyze the abnormal behavior. First, we review the definition. We define faint or falling down to be a person changing his posture from stand to lie quickly and never climb up for a period of time. We also define lying to be a person change his posture from stand to sit and then to lie. From the definition, we design a finite state machine depicted in Fig.21 to analyze human abnormal behavior. In Fig.21, the state machine consists of stand state, sit state, lie state, error state and abnormal state. The transition function corresponds to the postures associated with the behaviors. Once the state machine reaches a state, except the error state, the corresponding behavior is detected. For the error state, we believe human can not change the lie state directly to stand state. Thus, if reaching the error state, we restart the state machine.

Since the object segmentation error and ghost effect, the state machine in Fig.21 always go to a wrong state. Thus, we add some unstable states [31] into Fig.21 to tolerance the noise. The new finite state machine is shown in Fig.22. In the new state machine, we add a threshold t and six states into Fig.21. These six states, used to tolerance the noise, are sit to stand, stand to sit, sit to lie, lie to sit, abnormal unstable and error unstable. On the other hand, the noises may appear continuously within a period of time. Thus, we add a loop with threshold t at all unstable states to tolerance this kind of noise.

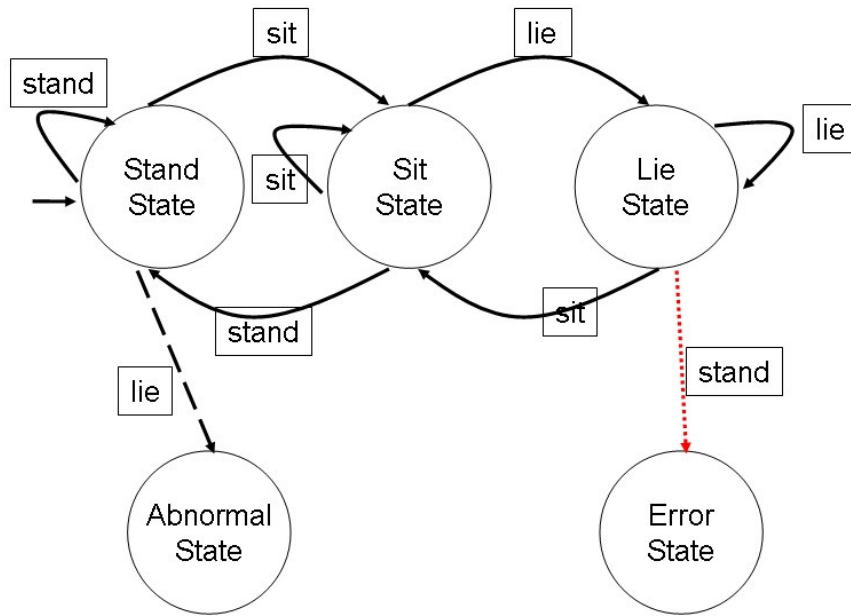


Fig. 21. A finite state machine for the analysis of abnormal behavior

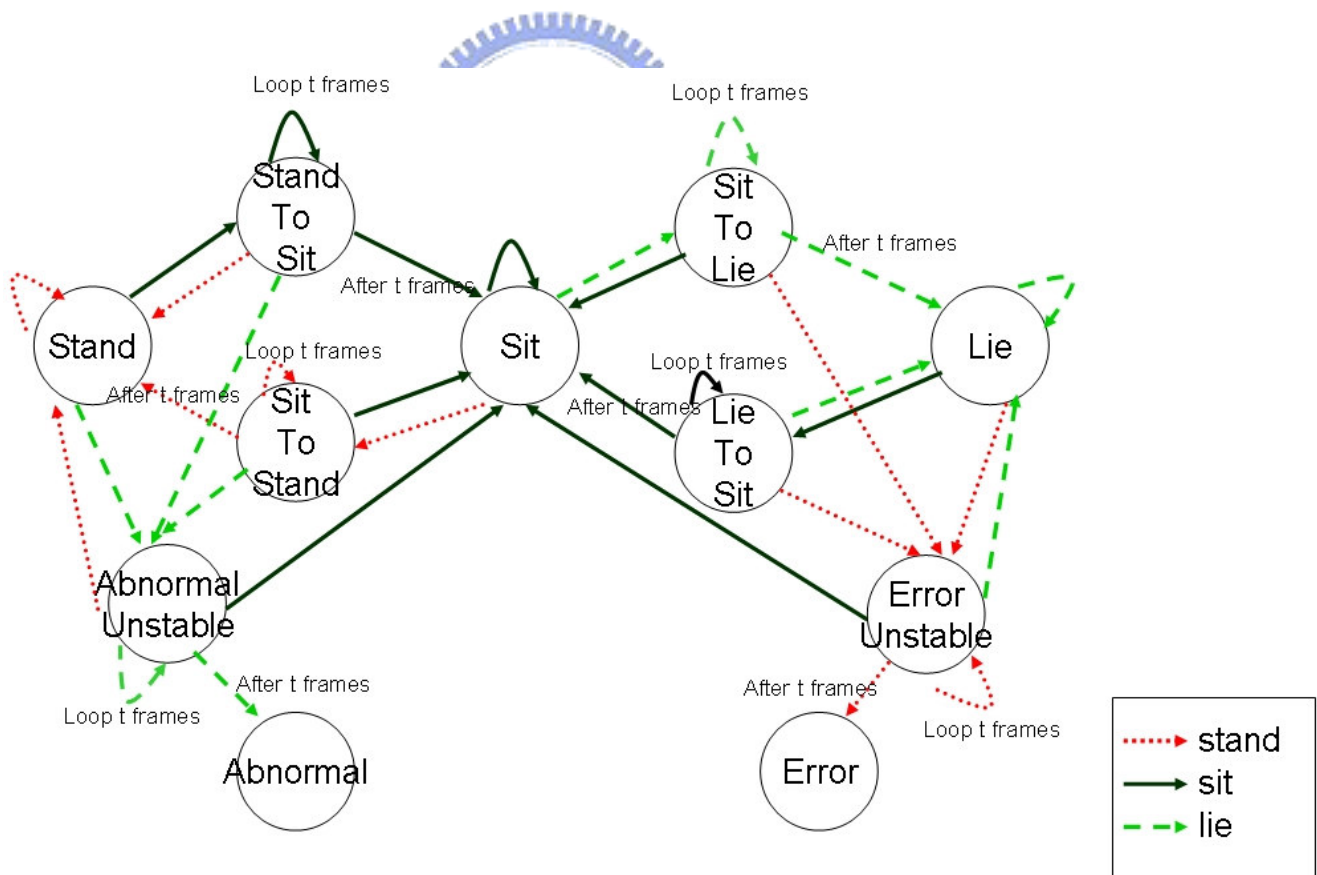


Fig. 22. An adaptive finite state machine for the analysis of abnormal behavior

Chapter 5

System Architecture and Experimental Results

In this chapter, we present the system for object-based video tracking and human behavior analysis. In the Section 5.1, we first introduce an overview of the system architecture. In the Section 5.2, we show the experimental results of video object segmentation. In the Section 5.3, the experimental results of video object segmentation are shown. At last, the experimental results of analyzing abnormal behavior of human are shown in Section 5.4.

5.1 System Architecture Overview

In this thesis, we implement an object-based tracking and human abnormal behavior analysis system on surveillance videos. The raw video data captured lively are input to our system. After deciding some pre-deciding thresholds, all the initializations are done automatically without manual interactions. Fig.23 shows the overview of the system.

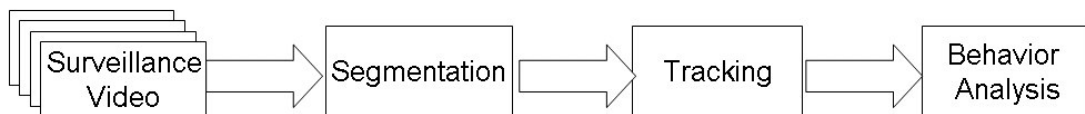


Fig. 23. Surveillance system architecture overview

5.2 Experimental Results of the Video Object Segmentation

In the segmentation algorithm, the first video frame is set as the background buffer to speed up the convergence of background image. In the morphological

operations, the size of structuring element for closing operation is 9 by 9 and the size for opening operation is 5 by 5. The size threshold used for object size filtering is 450 pixels.

Fig.24 through Fig.28 show some results of the segmentation before eliminating shadow and the result after eliminating shadow. Fig.24(a), Fig.25(a), Fig.26(a) and Fig.27(a) show the segmentation without shadow elimination. Fig.24(b), Fig.25(b), Fig.26(b) and Fig.27(b) show the segmentation after shadow elimination. From the figure, our shadow elimination algorithm successfully eliminates the most shadow.

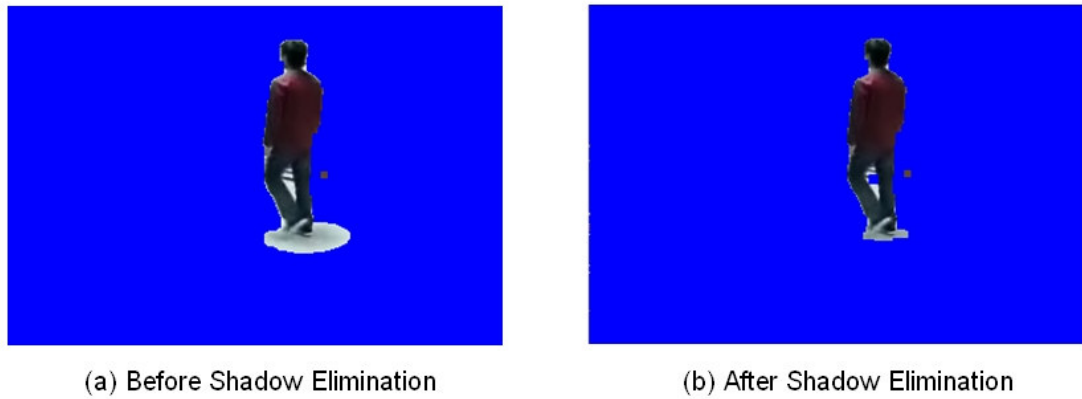


Fig. 24. The shadow thresholds $t_1=0.88$ and $t_2=0.01$

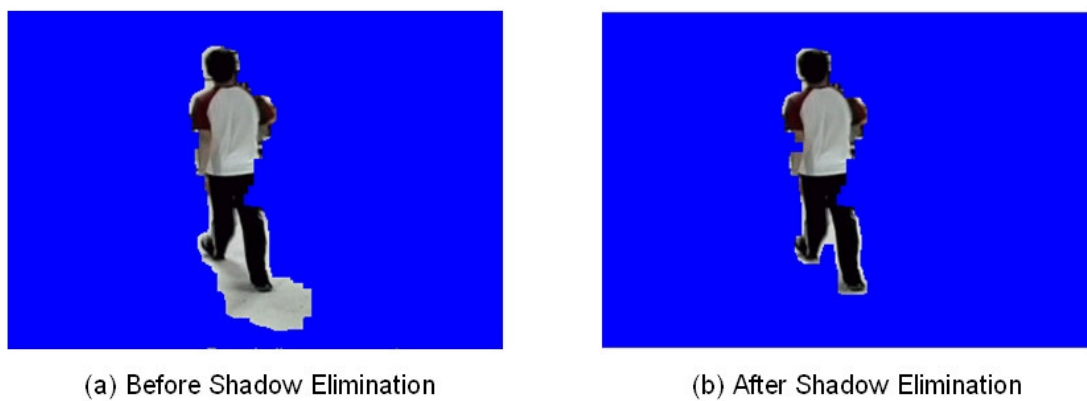


Fig. 25. The shadow thresholds $t_1=0.88$ and $t_2=0.01$

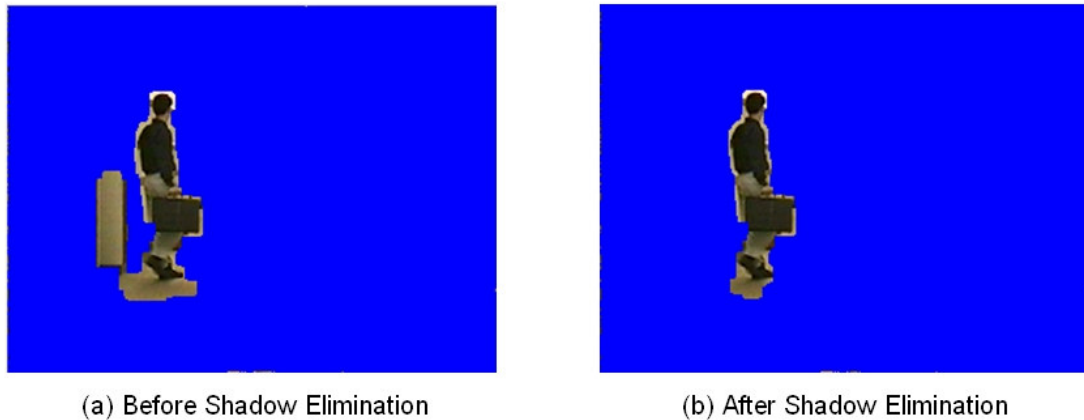


Fig. 26. The shadow thresholds $t_1=0.88$ and $t_2=0.01$

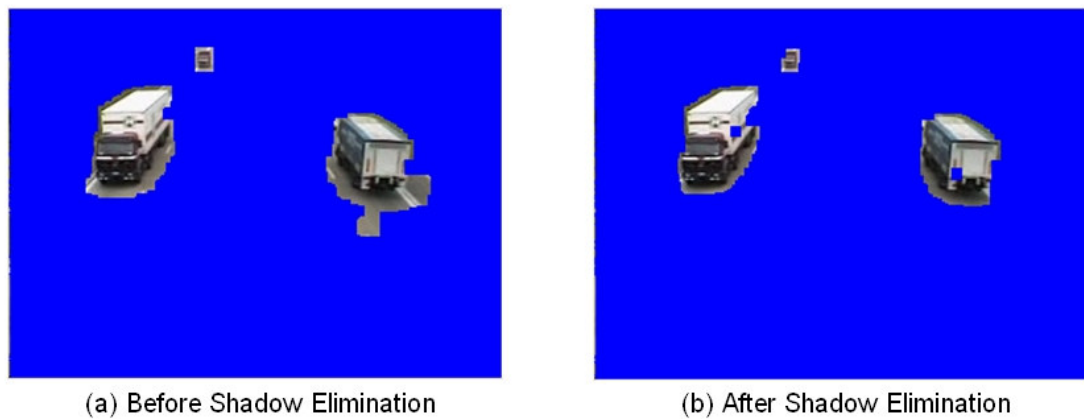


Fig. 27. The shadow thresholds $t_1=0.94$ and $t_2=0.01$

If we change the shadow thresholds t_1 and t_2 in Fig.27(b) to 0.80 and 0.01, respectively, we obtain the result shown in Fig.28. The result in Fig.28 is not good because of the shadow threshold setting. To explore the reason, we review our shadow elimination in Section 2.2.5. We use the principle of color reflection to analyze the color of shadow without analyzing the color of moving object. Thus, the shadow elimination algorithm will fail when we set unsuitable thresholds.

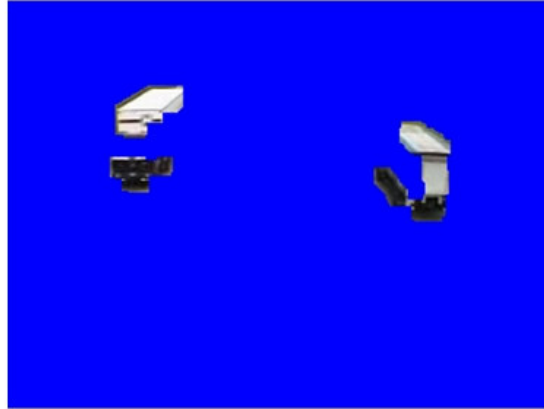


Fig.28 After Shadow Elimination,
The two shadow thresholds t_1 and t_2
set to 0.80 and 0.01, respectively.

5.3 Experimental Results of the Video Object Tracking

In the tracking algorithm, the distance threshold of the single object matching algorithm is heuristically set to 15.

Fig.29 through Fig.31 show the results of video object tracking algorithm. In order to check the result easily, objects which belong to the same trajectory are marked with an identifying label manually. In figures, same labeling is considered as the same object. Fig.29 shows a video clip containing 3 trucks and 1 car without occlusion. Fig.29(a)-(e) shows the tracking result of object 1 and object 2. Fig.29(f), (g) shows the tracking result of new object 3 and disappearance of object 2. Fig.29(h) shows the appearance of object 4. Our single object matching algorithm is successfully to track these 4 objects. Fig.30 shows a video clip containing 3 persons and occlusion twice. Fig.30(a), (b) shows the tracking result of object 1 and object 2. The object 1 and object 2 occlude in Fig.20(c), (d), we use the symbol 1&2 to indicate the occlusion of object 1 and object 2. Fig.30(e), (f) shows the tracking result of object 1, object 2 and object 3. Object 2 and object 3 occlude in Fig.30(g), (h), (i) and use the symbol 2&3 to indicate the occlusion of object 2 and object 3. Occlusion

object 2&3 split in Fig.(j). Our multiple objects matching algorithm is successfully to track the 3 persons. Another example to test our multiple objects matching algorithm is shown in Fig.31. There have 3 persons and also occlusion twice. But the sizes of objects are different. Also, our multiple objects matching algorithm success to track 3 persons.



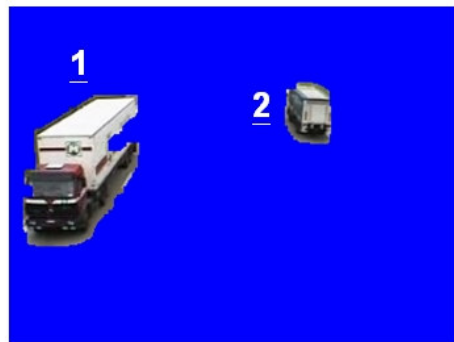
(a) A new object 1



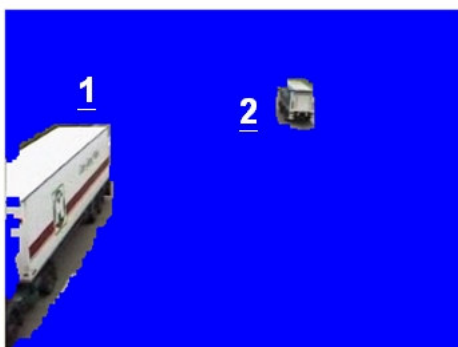
(b) A new object 2



(c) Tracking object 1 and 2



(d) Tracking object 1 and 2



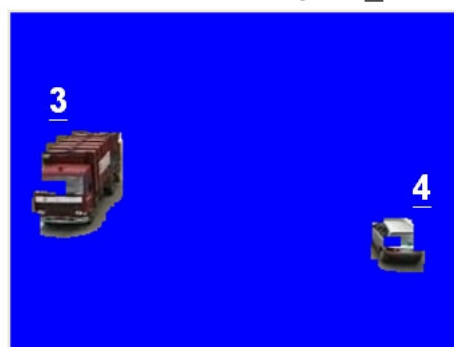
(e) Tracking object 1 and 2



(f) Object 1 disappear, detect a new object 3

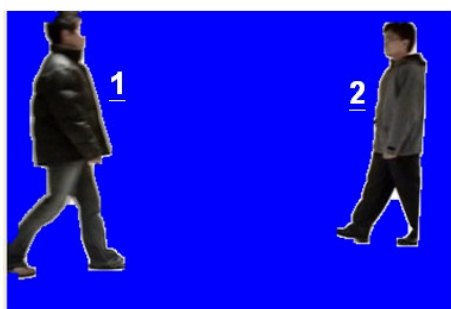


(g) Object 2 disappear



(h) Detect a new object 4

Fig. 29. Tracking results of the speedway sequence



(a) Detect object 1 and 2



(b) Tracking object 1 and 2



(c) Detect occlusion object 1&2



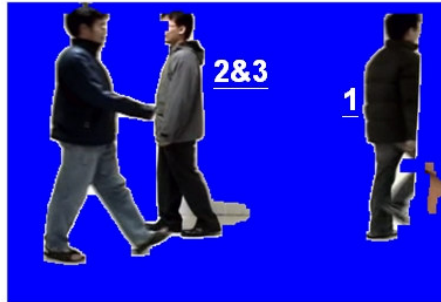
(d) Tracking occlusion object 1&2



(e) Object 1&2 split into object 1 and 2 and a new object 3 appearance



(f) Tracking object 1, 2 and 3



(g) Detect occlusion object 2&3



(h) Tracking occlusion object 2&3

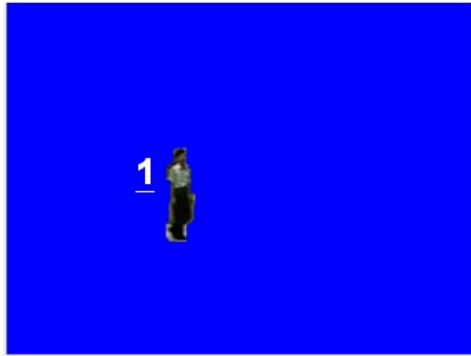


(i) Tracking occlusion object 2&3

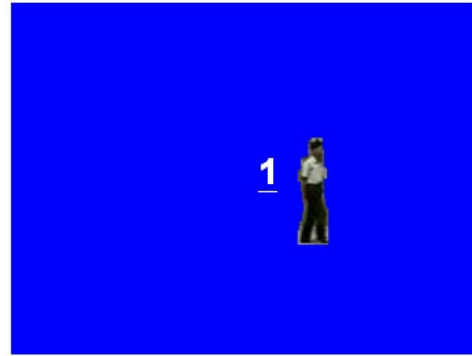


(j) Object 2&3 split into object 1 and 2

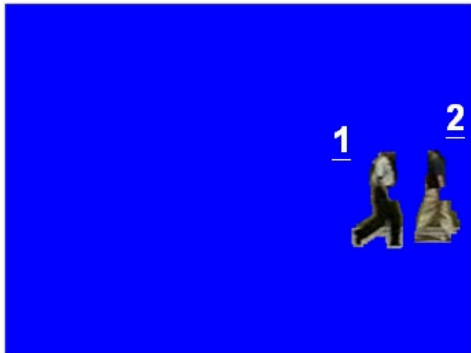
Fig. 30. Tracking results of our Lab. Test Seq. 1



(a) New object 1



(b) Tracking object 1



(c) New object 2



(d) Detect occlusion object 1&2



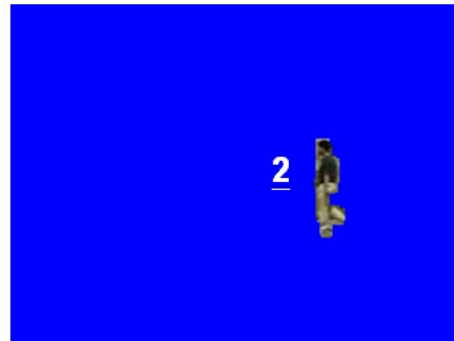
(e) Tracking occlusion object 1&2



(f) Tracking occlusion object 1&2



(g) Occlusion object 1&2 split into object 1 and 2



(h) Object 1 disappearance

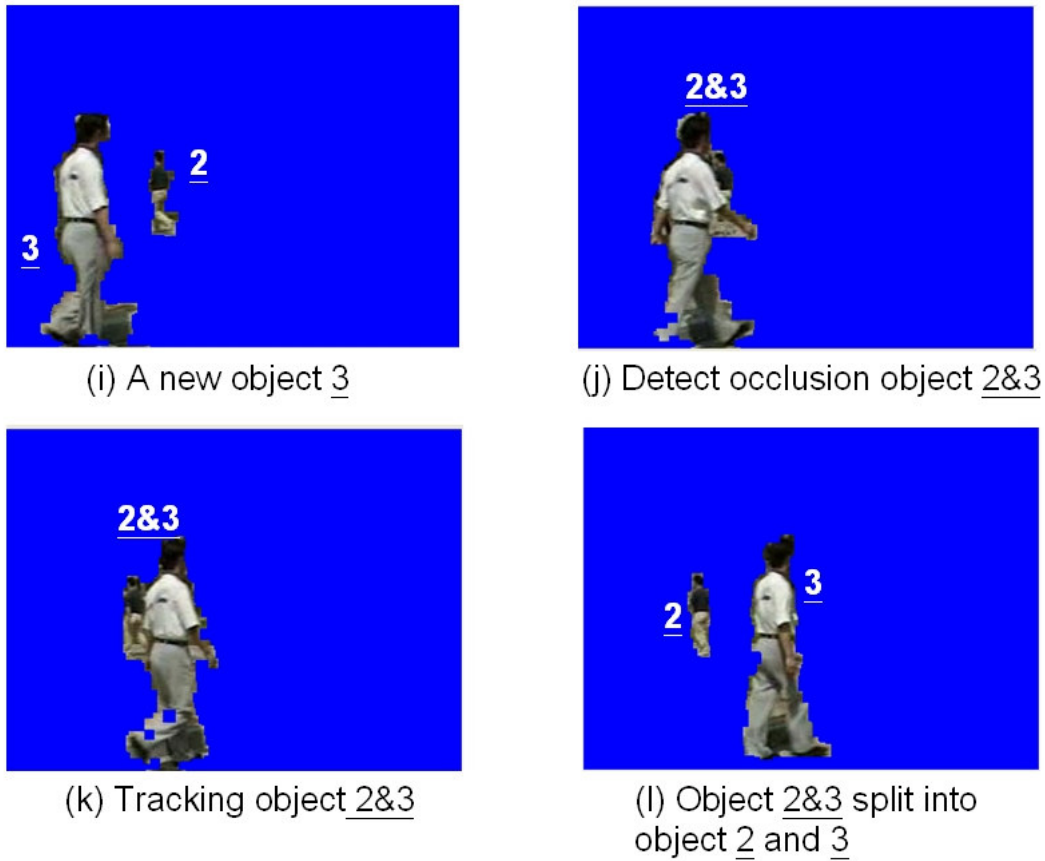


Fig. 31. Tracking results of the ETRI_C Seq.

The results show that the tracking algorithm successfully tracks the video objects and detects the occlusion events. Our algorithm also matches the split objects to the objects before occlusion correctly. The Table.1 shows the statistics of the detecting and tracking of occlusion and split events. The results show that all the objects before and after the occlusions are matched perfectly.

Table.1 Statistics of the tracking and detecting of occlusion events

Sequence name	Number of occlusion events occurred	Number of occlusion events detected	Number of the matching failures after the split
Lab. Test Seq. 1	2	2	0
Lab. Test Seq. 2	3	3	0
ETRI_C Seq.	2	2	0

5.4 Experimental Results of the Human Behavior Analysis

To evaluate the performance of the system in the recognition of human behaviors and postures, we test 30 video sequences containing 2 kinds of abnormal behaviors and 500 testing data for each posture. Table.2 shows the recognition rates of postures. We calculate the ratio of success frame number to the total testing frame number. From the results, we can find that the performance for the sit-pose is the lowest. Since sit-pose is a posture relatively similar to the stand-pose, it is misrecognized as stand-pose. The stand pose shows a lower performance than the lie-pose. This is because the stand-pose is extracted in moving situation and therefore would involve more complicated background situations. On the other hand the lie-pose is extracted on static situation, on a fixed background. Consequently, it presents a higher recognition rate. Table.3 shows the behavior recognition results. The false positive means that a test incorrectly gives a positive result. The true positive means that a test correctly gives a positive result. From Table.3, we know that our method correctly detect the 2 abnormal behavior but wrongly detect 4 video sequence as abnormal behaviors. The reason of wrong detection is caused by the error recognition of posture.

Table.2 The recognition rates for three postures

Behavior Type	Success Frame/Total Frame
Sit pose	74%
Stand pose	83%
Lie pose	90%

Table.3 The recognition results for abnormal behavior

	Number of Abnormal video	Number of true positive	Number of false positive
Abnormal behavior	2	2	4

Chapter 6

Conclusion and Future work

In this thesis, we presented a system for object-based video tracking and human abnormal behavior analysis on surveillance videos. We adopted a simple but effective shadow elimination algorithm to eliminate shadow in object segmentation. From the experimental results, we know that the shadow threshold deciding the shadow elimination result. Besides, we also designed two matching algorithms, using color and shape information of object combine with a score function, to track objects. Especially, the multiple objects matching algorithm successfully detect the occlusion and split objects. Besides, we designed a finite state machine to analyze human abnormal behavior and obtain a satisfactory result. Based on this system structure, we implemented a tracking system and analyzing human abnormal behavior.

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

- (i) Dynamic finding a threshold while eliminating shadow. We use a pre-decided threshold in the shadow elimination module. In the future, we wish to design an algorithm to find the threshold dynamically.
- (ii) Finding a more reliable algorithm in multiple objects matching algorithm. In the multiple objects matching algorithm, we use a greedy approach to find the matching. We wish to find a better method to solve this problem.
- (iii) Extracting more kinds of posture and behavior from video. In our system, we only extract sit, stand, lie three postures and only analyze the faint and falling down two abnormal behavior. In future, we wish to extract more postures and design a complex final state machine to analyze more

abnormal behavior.

- (iv) Summarization and abstraction of human behavior in the video. The object-based abstractions are very valuable and useful. The system can be further extended for the content retrieval and management. To achieve this, we can use the MPEG-7 descriptors to describe the contents with the detected abnormal behaviors and generated abstractions. And thus we can manage a database for surveillance and monitoring videos and important contents can be retrieved efficiently. We believe that the extraction of content will become more and more important and one day such a kind of systems will be widely adopted in the future.



Reference

- [1] Y. Tsaig and A. AverBuch, "Automatic Segmentation of Moving Objects in Video Sequence: A Region Labeling Approach," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.12, NO. 7, pp.597–612, 2002.
- [2] J.C. Choi, S.W. Lee and S.D. Kim, "Spatio-Temporal Video Segmentation Using a Joint Similarity Measure," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.7, NO. 2, pp. 279–286, 1997.
- [3] D. Wang, "Unsupervised Video Segmentation Based on Watersheds and Temporal Tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.8, NO. 5, pp. 539–546, 1998.
- [4] H.T. Nguyen, M. Worring and A. Dev, "Detection of Moving Objects in Video Using a Robust Similarity Measure," *IEEE Transactions on Image Processing*, Vol.9, NO. 1, pp.137–141, 2000.
- [5] T. Aach, A. Kaup and R. Mester, "Statistical Model-Based Change Detection in Moving Video," *Signal Processing*, Vol.31, NO. 2, pp.203–217, 1993.
- [6] A. Neri, S. Colonnese, G. Russo and P. Talone, "Automatic moving object and background separation," *Signal Processing*, Vol.66, pp.219–232, 1998.
- [7] D.D. Giusto, F. Massidda and C. Perra, "A Fast Algorithm for Video Segmentation and Object Tracking," *The 14th International Conference on Digital Signal Processing*, Vol.2, pp.697–700, Cagliari Univ., Italy, 2002.
- [8] S.Y. Chien, S.Y. Ma and L.G. Chen, "Efficient Moving Object Segmentation Algorithm Using Background Registration Technique," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.12, NO. 7, pp. 577–586, 2002.
- [9] E.P. Ong, B.J. Tye, W.S. Lin and M. Etoh, "An Efficient Video Object Segmentation Scheme," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol.4, pp.IV-3361–IV-3364, Singapore, 2002.
- [10] J.H. Pan, C.W. Lin, C. Gu and M.T. Sun, "A Robust Video Object Segmentation Scheme with Prestored Background Information," *IEEE International Symposium on Circuits and Systems*, Vol.3, pp.803 – 806, Seattle, WA, USA, 2002.
- [11] R. Cucchiara, C. Grana, M. Piccardi and A. Prati, "Detecting Moving Objects, Ghosts and Shadows in Video Streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.25, NO. 10, pp.1337–1342, 2003.
- [12] S.Y. Chien, Y.W. Huang, B.Y. Hsieh, S.Y. Ma and L.G. Chen, "Fast video segmentation algorithm with shadow cancellation, global motion compensation, and adaptive threshold techniques" *Multimedia*, *IEEE Transactions on Digital Object Identifier*, Vol.6, NO. 5, pp.732–748, 2004.
- [13] M. Dahmane and J. Meunier, "Real-time video surveillance with self-organizing

maps” Computer and Robot Vision Proceedings. The 2nd Canadian Conference, pp.136–143, May 2005.

[14] A. Leone, C. Distanto; N. Ancona, E. Stella and P. Siciliano, ” Texture analysis for shadow removing in video-surveillance systems” Systems, Man and Cybernetics, 2004 IEEE International Conference, Vol. 7, pp.6325–6330, Oct 2004.

[15] W.M. Lu and Y.P. Tan, ”A vision-based approach to early detection of drowning incidents in swimming pools” Circuits and Systems for Video Technology, IEEE Transactions on Digital Object Identifier, Vol. 14, Issue 2, pp.159 - 178, Feb. 2004.

[16] F. Oberti, S. Calcagno, M. Zara, and C.S. Regazzoni, “Robust Tracking of Human and Vehicles in Cluttered Scenes With Occlusions,” Proceedings of International Conference on Image Processing, Vol.3, pp.629 – 632, Genova, Italy, 2002.

[17] C.G. Kim and J.N. Hwang, “Fast and Automatic Segmentation and Tracking for Content-Based Application,” IEEE Transactions on Circuits and Systems for Video Technology, Vol.12, NO. 2, pp.122–129, 2002.

[18] Y.W. Chen, D.Y. Chen and S.Y. Lee, “Moving Object Tracking for Video Surveillance in Compressed Videos,” The 7th International Conference on Internet and Multimedia Applications and Systems, pp.695–698, 2003.

[19] Y.K. Jung, K.W. Lee and Y.S. Ho, “Content-Based Event Retrieval Using Semantic Scene Interpretation for Automated Traffic Surveillance,” IEEE Transactions on Intelligent Transportation Systems, Vol.2, NO. 3, 2001.

[20] Y. Huang, T.S. Huang and H. Niemann, “Segmentation-Based Object Tracking Using Image Warping and Kalman Filtering,” Proceedings of International Conference on Image Processing, Vol.3, pp.601–604, Urbana, IL, USA, 2002.

[21] G.Z. Cao, J.P. Jiang and J.Q. Chen, “An improved object tracking algorithm based on Image Correlation,” IEEE International Symposium on Industrial Electronics, Vol.1, pp.598–601, Hanzhou, China, 2003.

[22] G.L. Foresti, ” Object recognition and tracking for remote video surveillance” Circuits and Systems for Video Technology, IEEE Transactions, Vol. 9, Issue 7, pp.1045–1062, Oct. 1999.

[23] T. Wada and T. Matsuyama, “Multiobject Behavior Recognition by Event Driven Selective Attention Method” IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.22, NO.8, Aug 2000.

[24] I. Haritaoglu, D. Harwood and L.S. David, “W4Real-Time Surveillance of People and Their Activities.” IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.24,NO.8, Aug 2000.

[25] R. Polana and R. Nelson, “Recognizing Activities,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 815–818, 1994.

- [26] B.A. Jezekiel, Z.Q. Wang, W. Pandif and S. Rajaram. "Human Activity Recognition Using Multidimensional Indexing." IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.24, NO.8, Aug 2002.
- [27] C.D. Liu, P.C. Chuug and Y.N. Chung "Human Home Behavior Interpretation from Video Streams" Networking, Sensing and Control, 2004 IEEE International Conference, Vol. 1, pp.192–197, Mar. 2004.
- [28] P. Peursum, S. Venkatesh, G.A.W. West and H.H. Bui, " Object labelling from human action recognition" Pervasive Computing and Communications. Proceedings of the First IEEE International Conference, pp.399–406, Mar. 2003.
- [29] Y. Chen and D.H. Ballard," Learning to recognize human action sequences" Development and Learning, 2002. Proceedings. The 2nd International Conference, pp.28–33, Jun. 2002.
- [30] J.W. Davis and A. Tyagi, " A reliable-inference framework for recognition of human actions" Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, pp.169–176, Jul. 2003.
- [31] R. Cucchiara, C. Grana, A. Prati and R. Vezzani, " Probabilistic posture classification for Human-behavior analysis" Systems, Man and Cybernetics, Part A, IEEE Transactions, Vol. 35, Issue 1, pp.42–54, Jan. 2005.
- [32] J. Yamato, J. Ohya and K. Ishii, " Recognizing human action in time-sequential images using hidden Markov model" Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference, pp.379–385, Jun. 1992.
- [33] P. Sangho and J.K. Aggarwal, " Semantic-level Understanding of Human Actions and Interactions using Event Hierarchy" Computer Vision and Pattern Recognition Workshop, pp.12–12, Jun. 2004.
- [34] S. Muller-Schneiders, T. Jager, H.S. Loos, W. Niem and R. Bosch, " Performance Evaluation of a Real Time Video Surveillance System" Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2nd Joint IEEE International Workshop, pp.137–144, Oct. 2005.
- [35] X.D. Liu and G.D. Su, " A new network-based intelligent surveillance system" Signal Processing Proceedings, WCCC-ICSP 2000. 5th International Conference, Vol. 2, pp.1187–1192, Aug. 2000.
- [36] M. Valera and S.A. Velastin, " Real-time architecture for a large distributed surveillance system" Intelligent Distributed Surveillance Systems, pp.41–45, Feb. 2004.
- [37] L. Fuentes and S.A. Velastin, " Advanced Surveillance: From Tracking To Event Detection" Latin America Transactions, IEEE (Revista IEEE America Latina), Vol. 2, Issue 3, pp.1–1, Sept.2004.

- [38] M. Valera and S.A. Velastin, "Intelligent distributed surveillance systems: a review" *Vision, Image and Signal Processing, IEE Proceedings*, Vol. 152, Issue 2, pp.192–204, Apr. 2005.
- [39] X. Desurmont, A. Bastide, C. Chaudy, C. Parisot, J.F. Delaigle and B. Macq, "Image analysis architectures and techniques for intelligent surveillance systems" *Vision, Image and Signal Processing, IEE Proceedings*, Vol. 152, Issue 2, pp.224–231, Apr. 2005.
- [40] Y.T. Chien, Y.S. Huang, S.W. Jeng, Y.H. Tasi and H.X. Zhao, "A real-time security surveillance system for personal authentication" *Security Technology, Proceedings. IEEE 37th Annual 2003 International Carnahan Conference*, pp.190–195, Oct. 2003.
- [41] C. Kim and J.N. Hwang, "Object-based video abstraction for video surveillance systems" *Circuits and Systems for Video Technology, IEEE Transactions*, Vol. 12, Issue 12, pp.1128 – 1138, Dec.2002.
- [42] S. Morita, K. Yamazawa and N. Yokoya, "Networked video surveillance using multiple omnidirectional cameras" *Computational Intelligence in Robotics and Automation, IEEE International Symposium*, Vol. 3, pp.1245–1250, Jul.2003.
- [43] Q.Z. Wu, H.Y. Chang and K.C. Fan, "Motion Detection Based on Two-Piece Linear Approximation for Cumulative Histograms of Ratio Image in Intelligent Transportation Systems," *Proceedings of IEEE International Conference on Networking, Sensing & Control*, Vol.1, pp.309–314, 2004.
- [44] J.B. Kim, H.S. Park, M.H. Park and H.J. Kim, "Unsupervised Moving Object Segmentation and Recognition Using Clustering and A Neural Network," *International Joint Conference on Neural Networks*, Vol.2, pp. 1240–1245, Taegu , South Korea, 2002.