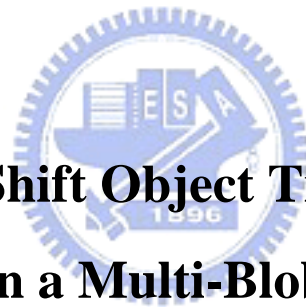


# 國立交通大學

電子工程學系 電子研究所碩士班

碩 士 論 文

基於多團塊模型及平均移動法之物體追蹤技術



## Mean-Shift Object Tracking Based on a Multi-Blob Model

研 究 生：姚文翰

指 導 教 授：王聖智 博士

中 華 民 國 九 十 五 年 六 月

# 基於多團塊模型及平均移動法之物體追蹤技術

## Mean-Shift Object Tracking Based on a Multi-Blob Model

研究生：姚文翰

Student：Wen-Han Yao

指導教授：王聖智博士

Advisor：Dr. Sheng-Jyh Wang

國立交通大學

電子工程學系 電子研究所碩士班



Submitted to Department of Electronics Engineering & Institute of Electronics

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of Master

in

Electronics Engineering

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年六月

# 基於多團塊模型及平均移動法之物體追蹤技術

研究生：姚文翰

指導教授：王聖智 博士

國立交通大學

電子工程學系 電子研究所碩士班

## 摘要



在本文中，我們提出一套能夠自動偵測畫面中移動物體並持續追蹤的演算法，並嘗試解決在物體追蹤問題中常遭遇的遮蔽、場景變化以及光源變化等等問題。我們自行定義一個多團塊模型用以描述移動物體，並基於多團塊模型定義適合的相似性度量，進而發展出以平均移動法為基礎的追蹤系統。我們也針對移動物體在畫面中的大小以及方向性提出一套調整方式。整個系統還包括模型更新、目標丟失等等判斷機制，讓追蹤結果更加合理、強韌。實驗結果顯示我們提出來的演算法在室內、室外等不同場景都能夠正確地追蹤移動物體的運動行為。


# Mean-Shift Object Tracking Based on a Multi-Blob Model

Student : Wen-Han Yao    Advisor : Dr. Sheng-Jyh Wang

Department of Electronics Engineering, Institute of Electronics

National Chiao Tung University

## Abstract



In this thesis, we proposed an object tracking system, which can automatically detect a single moving object in an image sequence and keep tracking of this object. In the proposed system, we deal with the problems of occlusion, scene change and luminance change. A multi-blob model is defined in our approach to represent the moving object. With this multi-blob model, we proposed a new similarity measure and developed a new object tracking algorithm based on the mean-shift method. We also proposed a strategy to update the size and orientation of the bounding ellipse of the moving object. For the sake of robustness, the proposed system contains decision criteria to handle model updating and loss of target. Simulation results demonstrate that the proposed object tracking algorithm can faithfully track the moving object in different scenes.

## 誌謝

特別感謝我的指導教授 王聖智老師，除了在學術上悉心教導外，同時也是我們學習待人處事的典範。感謝實驗室的全體夥伴，因為有你們在生活上的協助及學業上的討論，這篇論文才得以順利完成。最後要感謝我親愛的家人，因為他們的愛，鼓舞遇到瓶頸的我，仍能打起精神，往前邁進。



# Content

摘要 .....	i
Abstract .....	ii
誌謝 .....	iii
Content .....	iv
List of Figures .....	v
Chapter 1. Introduction .....	1
Chapter 2. Backgrounds of Object Tracking .....	2
2.1 Motion Detection .....	2
2.1.1 Background Subtraction .....	2
2.1.2 Temporal Differencing .....	3
2.1.3 Optical Flow .....	4
2.2 Motion Tracking .....	6
2.2.1 Region-Based Tracking .....	6
2.2.2 Active Contour-Based Tracking .....	7
2.2.3 Model-Based Tracking .....	8
2.2.4 Feature-Based Tracking .....	9
Chapter 3. Mean-Shift Tracking and Multi-Blob Model .....	15
3.1 Traditional Mean-Shift Tracking .....	15
3.1.1 Building Models and Measuring Similarity .....	15
3.1.2 Object Tracking Procedure .....	17
3.1.3 Discussion .....	22
3.2 Multi-Blob Model Based Mean-Shift Tracking .....	23
3.2.1 Moving Object Detection .....	23
3.2.2 Multi-Blob Model .....	28
3.2.3 Similarity Measure .....	31
3.2.4 Mean-Shift Tracking Procedure .....	33
3.2.5 Updating Size and Orientation .....	35
3.2.6 Updating Reliability .....	41
3.2.7 Updating Target Model .....	44
3.2.8 Overall Object Tracking Process .....	46
Chapter 4. Experimental Results .....	48
Chapter 5. Conclusions .....	60
Reference .....	61

## List of Figures

Figure 2.1 Effect of the second stage of detection on suppressing false detections. (a) Original image. (b) First stage detection result. (c) Suppressing pixels with high displacement probabilities. (d) Result using component displacement probability constraint. [4] .....	3
Figure 2.2 A human body is considered as a combination of blobs. (a) Original image. (b) A two dimensional representation of the blob statistics. [6].....	7
Figure 2.3 Frames from a sequence of a person who drops an object. [7].....	7
Figure 2.4 Skeletonization of moving targets. The structure and rigidity of the skeleton is significant in analyzing target motion. [8] .....	8
Figure 2.5 (a) The tree hierarchy. (b) Tracking skeletons in 3-D: front and recovered side-views. [10].....	9
Figure 2.6 Example of a model graph which contains boundary cells and internal cells. [11]	10
Figure 2.7 (a) The similarity surface, the initial and final locations of mean-shift iterations. (b) Tracking results. [12] .....	11
Figure 2.8 The spatiogram captures spatial relationships among various colors, whereas the histogram discards all spatial information. (a) Original image. (b) Statistical distribution generated from the histogram of (a). (c) Statistical distribution generated from the spatiogram of (a). [14] .....	12
Figure 2.9 Using the scale-space mode-tracking method. The person is tracked well, both spatially and in scale. [15].....	13
Figure 2.10 Updating both size and orientation through motion tracking. [16].....	13
Figure 2.11 A flow chart in [17], which describes how to obtain the feature score.....	14
Figure 2.12 Example of feature adaptation to avoid distractions. Left column: video frame with the object/background windows overlaid. Right column: weight image from top-ranked tracking features. [17] .....	14
Figure 3.1 Flow chart of building models and measuring similarity. ....	16
Figure 3.2 (a) The moving object to build the target model (Frame 15, sequence <i>Hans</i> ). (b) The red point is the starting point $\bar{y}_0$ of the mean-shift process, and the red bounding box represents the range of consideration corresponding to $\bar{y}_0$ . The blue point $\bar{y}$ is the ideal tracking result (Frame 250, sequence <i>Hans</i> )......	17
Figure 3.3 Mean-shift tracking flow .....	20
Figure 3.4 (a) The moving object to build target model (Frame 15, sequence <i>Hans</i> ). (b) The mean-shift process starts from the red point $\bar{y}_0$ . The blue point $\bar{y}$ is the ideal tracking result. However, the mean-shift tracking result converges to the green	

point $\bar{y}_1$ instead (Frame 250, sequence <i>Hans</i> ).	20
Figure 3.5 (a) The target model. (b) The color histogram and the similarity value with respect to tracking result $\bar{y}_1$ . (c) The color histogram and the similarity value with respect to the starting point $\bar{y}_0$ . (d) The color histogram and the similarity value with respect to the expected tracking result $\bar{y}$ .	21
Figure 3.6 The similarity surface, the initial and final locations of the mean-shift process. $\bar{y}$ is respect to the expected tracking result.	22
Figure 3.7 (a) Frame 75 in the <i>Watson</i> sequence. (b) Frame 85 in the <i>Watson</i> sequence.	23
Figure 3.8 Flow of calculating the statistics. The statistics is used to represent the covariance ellipse.	24
Figure 3.9 Relation between $[\sigma_x \ \sigma_y \ \rho]$ and $[a \ b \ \theta]$ .	25
Figure 3.10 Using a bounding ellipse to define the moving object.	27
Figure 3.11 Two different patterns which a spatiogram can not differentiate.	28
Figure 3.12 Five blobs that appear most frequently are marked with their mean values in the RGB color space.	30
Figure 3.13 (a) The moving object detected in Frame 15 of the <i>Hans</i> sequence. Build a multi-blob model according to the red bounding ellipse. (b) The region with the maximum similarity with the model in Frame 250 of the <i>Hans</i> sequence. (c) The similarity surface where $\bar{y}$ is the center of bounding ellipse shown in (b).	32
Figure 3.14 The foreground region and the background region based on a bounding ellipse.	35
Figure 3.15 Result of size and orientation updating. The blue ellipses represent the 2-sigma contour and the 3-sigma contour with respect to the original covariance matrix. The red ellipse represents the 2-sigma contour with respect to the modified covariance matrix.	40
Figure 3.16 (a) The outer ellipse represents the 3-sigma contour, and the inner ellipse represents the 2-sigma contour with respect to the modified covariance matrix. (b) Color histogram of the background field and foreground field.	41
Figure 3.17 Reliability function, both x-axis and y-axis are in the log scale.	42
Figure 3.18 The reliability map, which can roughly identify the moving object.	43
Figure 3.19 Frames with pixels marked depend on reliability. (a) Frame 50, Sequence <i>Watson</i> . (b) Frame 65, Sequence <i>Watson</i> .	44
Figure 3.20 Flow chart of the proposed object tracking process.	46
Figure 4.1 Experimental results of the sequence “ <i>Hans</i> ”. The orientation of the bounding ellipse and the reliability of blobs can be updated during tracking.	51
Figure 4.2 The reliability of Blobs 147 and 148. They are updated according to the background information.	51
Figure 4.3 The tracking result of sequence <i>Watson</i> . The target model updates at frame 65.	54
Figure 4.4 The variance ratio of tracking result.	55



Figure 4.5 The localization of the traditional mean-shift process is poor when the object's size decreases..... 56

Figure 4.6 Number of iterations when executing the proposed mean-shift process (red) and the traditional mean-shift procedure (blue)..... 57

Figure 4.7 The experimental result, where the red ellipse indicates the target. Due to the target loss, the algorithm detected the object at Frame 75, Frame 185 and Frame 220. We switch the color from red to green at these frames. .... 59



# Chapter 1.

## Introduction

The goal of object tracking is to find a specific target in successive frames of an image sequence. Various algorithms for object tracking have been proposed in recent years. Among these tracking algorithms, the mean-shift method has been popularly used due to its robustness and simplicity. This iterative ‘mean-shift’ process is a simple robust technique for the finding of the local maximum position without knowing the overall distribution. Recently, Comaniciu and Meer [1] successfully applied this mean-shift method to object tracking problems.

So far, in object tracking problems, mean-shift is used to find the position which has the maximal similarity with the target. However, this kind of approach has several serious defects. First, the spatial information of the target is lost and this fact causes the poor localization of the tracking result. Second, the commonly used similarity measures, such as the Bhattacharyya coefficient or the Kullback-Leibler distance, are not very discriminative. Third, the size and orientation of the tracked object cannot be updated in an efficient way.

In this thesis, we propose a new target model to represent the moving object and define a new similarity measurement based on the target model. We call this concept a multi-blob model. We derive a mean-shift process for the multi-blob model and demonstrate the improved tracking results. To make the system more complete, we develop a simple motion detection process to roughly detect the location and size of the moving region. Moreover, we also present a method to update both size and orientation of the bounding ellipse of the tracked object.

The thesis is organized as follows. In Chapter 2, we introduce the background of motion detection and motion tracking. In Chapter 3, we present the mean-shift process and discuss its drawbacks. Furthermore, we present our new target model, the multi-blob model, and our object tracking procedure. Some experimental results are given in Chapter 4. In Chapter 5 we draw our conclusions.

# Chapter 2.

## Backgrounds of Object Tracking

In general, many previously proposed approaches for motion detection and motion tracking are conceptually similar. It is fairly difficult to have a clear cut between these two issues. However, there exist some intrinsic differences between motion detection and motion tracking. In this chapter, motion detection and motion tracking are treated as two individual parts and will be discussed separately in the next two sections.

### 2.1 Motion Detection

Nearly every visual surveillance system starts with motion detection. Motion detection aims at segmenting regions corresponding to moving objects in an image. A good motion detection result usually makes the following motion tracking process much easier. We can roughly divide existing motion detection techniques into three major categories: background subtractions, temporal differencing, and optical flow.

#### 2.1.1 Background Subtraction

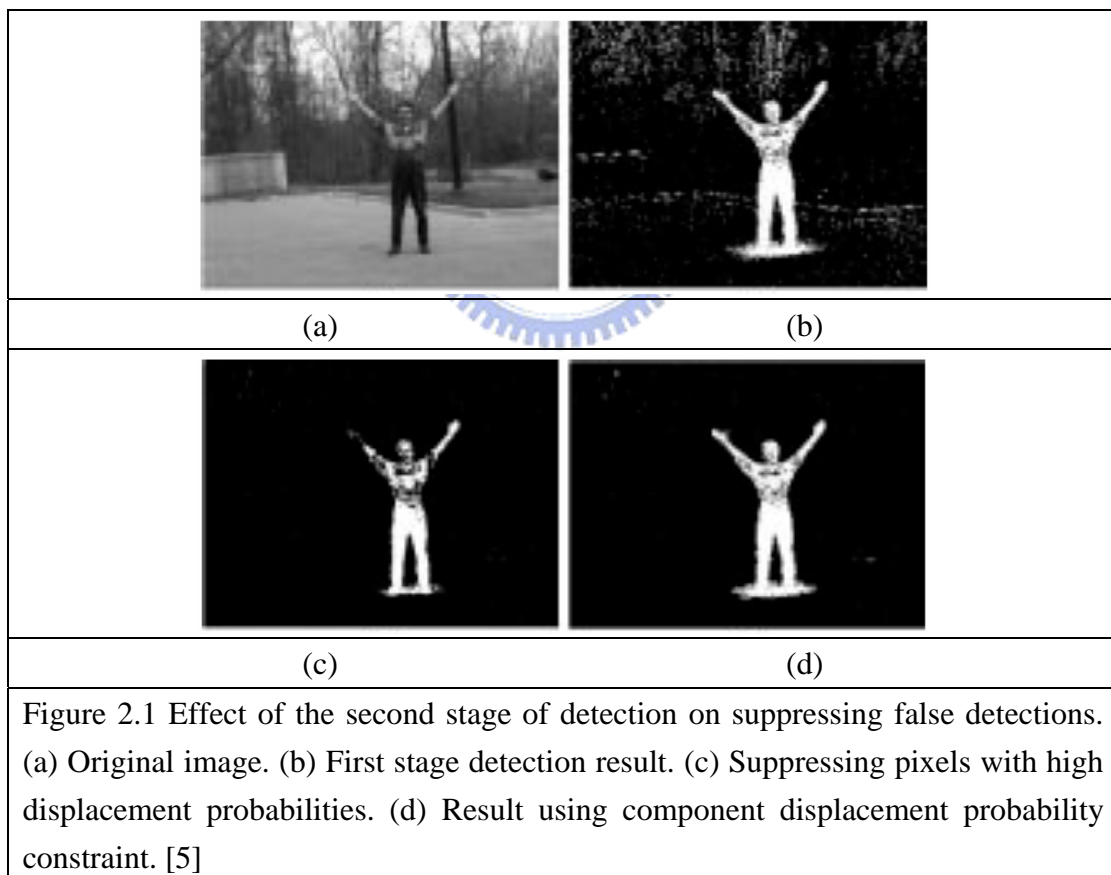
Background subtraction is a popular method for motion detection. It detects moving regions in an image by taking the difference between the current image and the reference background model in a pixel-by-pixel fashion. This method is extremely sensitive to dynamic changes caused by lighting change or shadows in the monitored scenes. Therefore, a reliable background model is in great demand to reduce the influence of these changes. That is, an active construction and updating of the background model are indispensable to visual surveillance.

For fixed cameras, the key problem is to automatically recover and update the background model based on a sequence of dynamic images. Unfavorable factors, such as illumination variance, shadows and shaking branches, bring difficulties to the acquirement and updating of background images. Some simple implementations use the time averages of image data, the adaptive Gaussian estimation, or the Kalman filtering to derive the background model. While these methods can run in real time, they are generally not robust enough. In [1], the authors consider each pixel as an independent statistical process, and record the observed intensity at each pixel over the previous  $n$  frames. The statistical distribution of the observed samples is then optimally fit to the model of a mixture of Gaussian functions. This approach assumes that the temporal behaviors of the intensity/color value at an image pixel are likely to follow

the normal distributions. Moreover, this approach assumes there could be more than one possible states at an image pixel when we do the observation over time.

A PTZ camera is a camera that can pan, tilt, and zoom. The scene captured by an active PTZ camera has non-stationary background. As a PTZ camera moves with respect to a rigid scene, the image content changes over time. In this case, motion compensation is needed to construct temporary background models. In [3], image mosaicing techniques are used to build a panoramic representation of the scene background. Alternatively, in [4], a representation of the scene background in terms of a finite set of images on a virtual polyhedron is used to construct images of the scene background at any pan-tilt-zoom setting.

On the other hand, the kernel density function is proposed in [5] to estimate the ensemble characteristics of sample data to produce the background model. This model keeps a sample of intensity values for each pixel in the image and uses this sample to estimate the density function of the pixel intensity distribution. Therefore, the model can estimate the probability of newly observed intensity. This model is able to handle the scene that is not completely stationary but contain small motions, like wavering tree branches.



## 2.1.2 Temporal Differencing

The pixel-wise differences between contiguous frames in an image sequence are used in the temporal differencing technique. There are many variants on the temporal differencing

method, and the simplest one is to calculate the absolute difference and to use a threshold function to detect the changes [6]. One problem with the temporal differencing technique is that the detection result tends to include undesirable background regions. These regions represent the positions where the target appears in the previous frame. One way to solve this problem is to introduce the knowledge of the target's motion to remove these background regions from the template. This kind of method is usually called "motion cropping". An IIR filter is usually used to update the template to ensure the current template may represent the target accurately.

### 2.1.3 Optical Flow

Optical flow is one of the most common approaches for motion estimation. In this approach, an image sequence is treated as a function  $f(x, y, t)$ , where  $x$  and  $y$  represent the spatial coordinates and  $t$  represents the temporal coordinate. It is assumed that the intensity value projected from a three dimensional point onto the image plane is unchanged all the time, even if the three dimensional point is under movement. This assumption can thus be expressed as

$$\frac{df(x, y, t)}{dt} = 0. \quad \text{Eq. 2-1}$$

Because  $(x, y)$  is also a function of  $t$ , we can apply the chain rule over the above equation. We may then rewrite Eq. 2-1 as

$$\frac{\partial f(x, y, t)}{\partial x} u(x, y, t) + \frac{\partial f(x, y, t)}{\partial y} v(x, y, t) + \frac{\partial f(x, y, t)}{\partial t} = 0. \quad \text{Eq. 2-2}$$

This equation is usually called the optical flow equation or optical flow constraint. If we define  $V = (u, v)$ , which represents the flow vector at each point, we can further reformulate the equation into the vector form:

$$\langle \nabla f(x, y, t), V(x, y, t) \rangle + \frac{\partial f(x, y, t)}{\partial t} = 0, \quad \text{Eq. 2-3}$$

where  $\langle \cdot \rangle$  is the inner product operator and  $\nabla = \left[ \frac{\partial}{\partial x} \quad \frac{\partial}{\partial y} \right]^T$  means the gradient operation.

Due to the fact that the inner product of  $\nabla f(x, y, t)$  and the part of  $V(x, y, t)$  that is perpendicular to  $\nabla f(x, y, t)$  will be zero, the optical flow approach cannot estimate the motion when the moving direction is perpendicular to the intensity gradient of the object. Since the gradient operator is sensitive to noise, some researches also apply the Gaussian filtering along the spatial axis and the temporal axis.

Optical flow detects motions only based on the intensity change. Hence, the detection may be unreliable. Two typical examples are the unobservable motion and the fake motion. When there is no obvious intensity change within the moving object, unobservable motion

happens. On the other hand, as the external illuminating condition changes, the intensity of a stationary object may change over time and fake motion may occur.



## 2.2 Motion Tracking

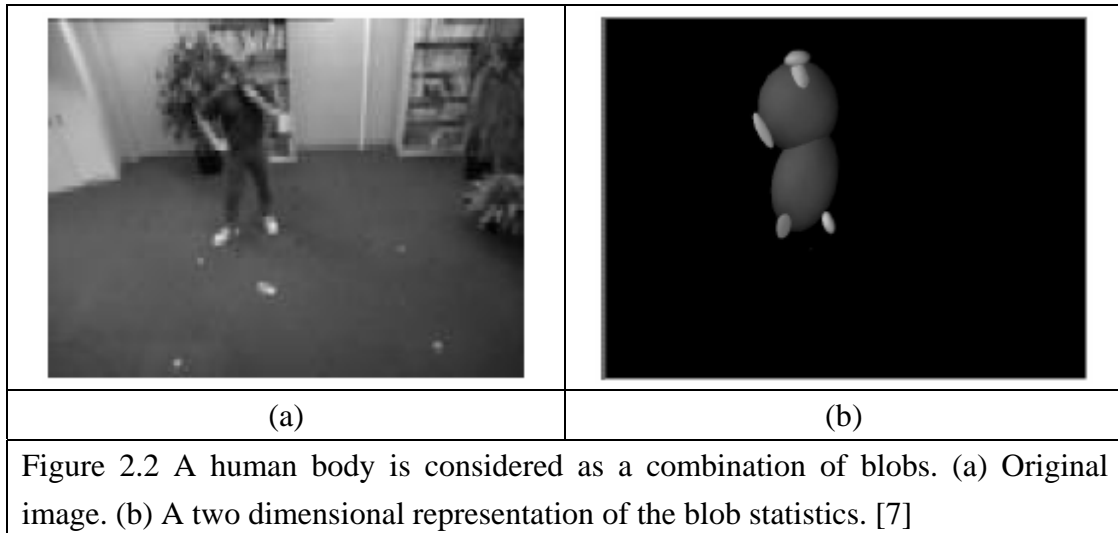
After motion detection, surveillance systems generally track moving objects from one frame to another in an image sequence. Although there are many researches trying to deal with the motion tracking problem, existing techniques are still not robust enough for stable tracking. In general, real-time execution is needed for a practical surveillance system; however, so far it is still hard to achieve high-resolution image quality under the time constraint. There still exist many problems in the motion tracking field.

In order to successfully track objects in an image sequence, various types of information are usually used to match an object in an image with the same object in another image. We can roughly classify the motion tracking techniques into a few categories in accordance with the used information. In the following sections, we'll introduce a few major types of motion tracking techniques. However, it is worth mentioning that a motion tracking process may use more than one kind of information and various kinds of information can be integrated together.

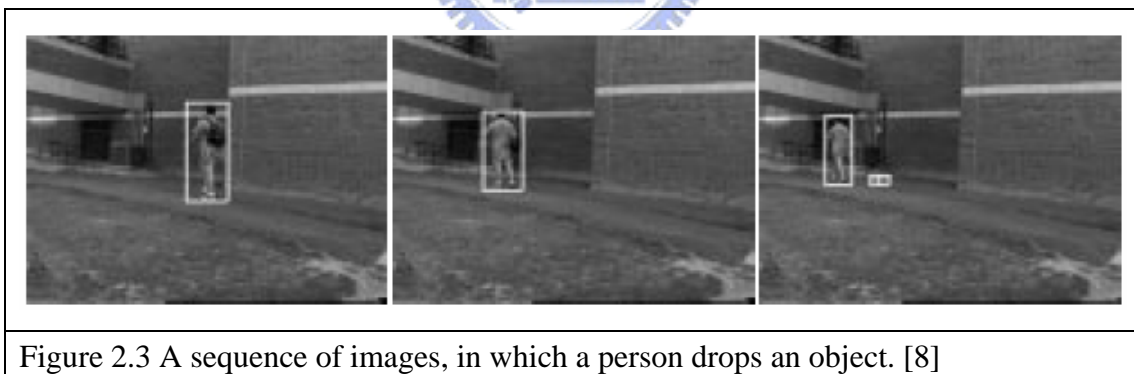
### 2.2.1 Region-Based Tracking

Region-based tracking algorithms track moving objects based on variations of the image regions corresponding to moving objects. For these algorithms, the background model is updated dynamically and motion regions are usually detected by subtracting the background from the current image. Hence, these algorithms use static cameras, instead of active cameras, because of the computational complexity in updating the background model.

In [7], the authors proposed an algorithm, which uses small blob features to track a single human body in an indoor environment. A human body is considered as a combination of several blobs. Each blob represents one body part, such as head, torso, or limb. Moreover, both human body and background are modeled as Gaussian distributions to represent the intensity value of every pixel. Finally, the pixels belonging to the human body are assigned to various blobs using the log-likelihood measure. Therefore, by tracking each small blob, a moving human object can be successfully tracked.



Although region-based algorithms usually use background subtraction to obtain moving regions, the shadow may cause false detection. To avoid false detection, [8] proposed an adaptive background subtraction method, in which color and gradient information are combined to cope with shadows and unreliable color cues in motion detection. Tracking is then performed at three levels of abstraction: regions, people, and groups. Regions can merge and split. A human is composed of one or more regions, which are grouped together under the condition of geometric constraints. On the other hand, a human group consists of one or more people. Therefore, using the region tracker and the individual color appearance model, we can deal with person-to-person and person-to-object interactions.

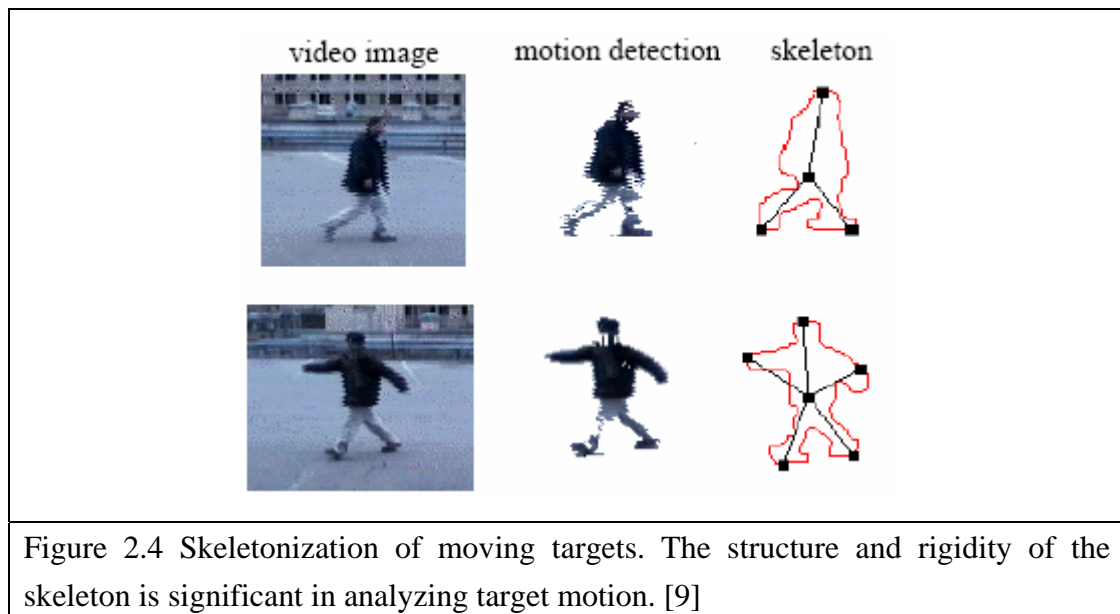


## 2.2.2 Active Contour-Based Tracking

Active contour-based tracking algorithms track objects by representing their outlines as bounding contours and by updating these contours dynamically for every frame. To find the bounding contour of the moving object, background subtraction is often applied. Nevertheless, no motion detection algorithm is perfect. There will be spurious pixels and holes in the detected moving features. In [9], a morphological dilation followed by an erosion operation is used to solve this problem. With the morphological operation, the bounding contour of the



moving object changes. This approach then skeletonizes the boundary to build a star representation for the moving object. By analyzing the torso angle and the star's periodic motion, simple behavior recognition can be achieved [9].

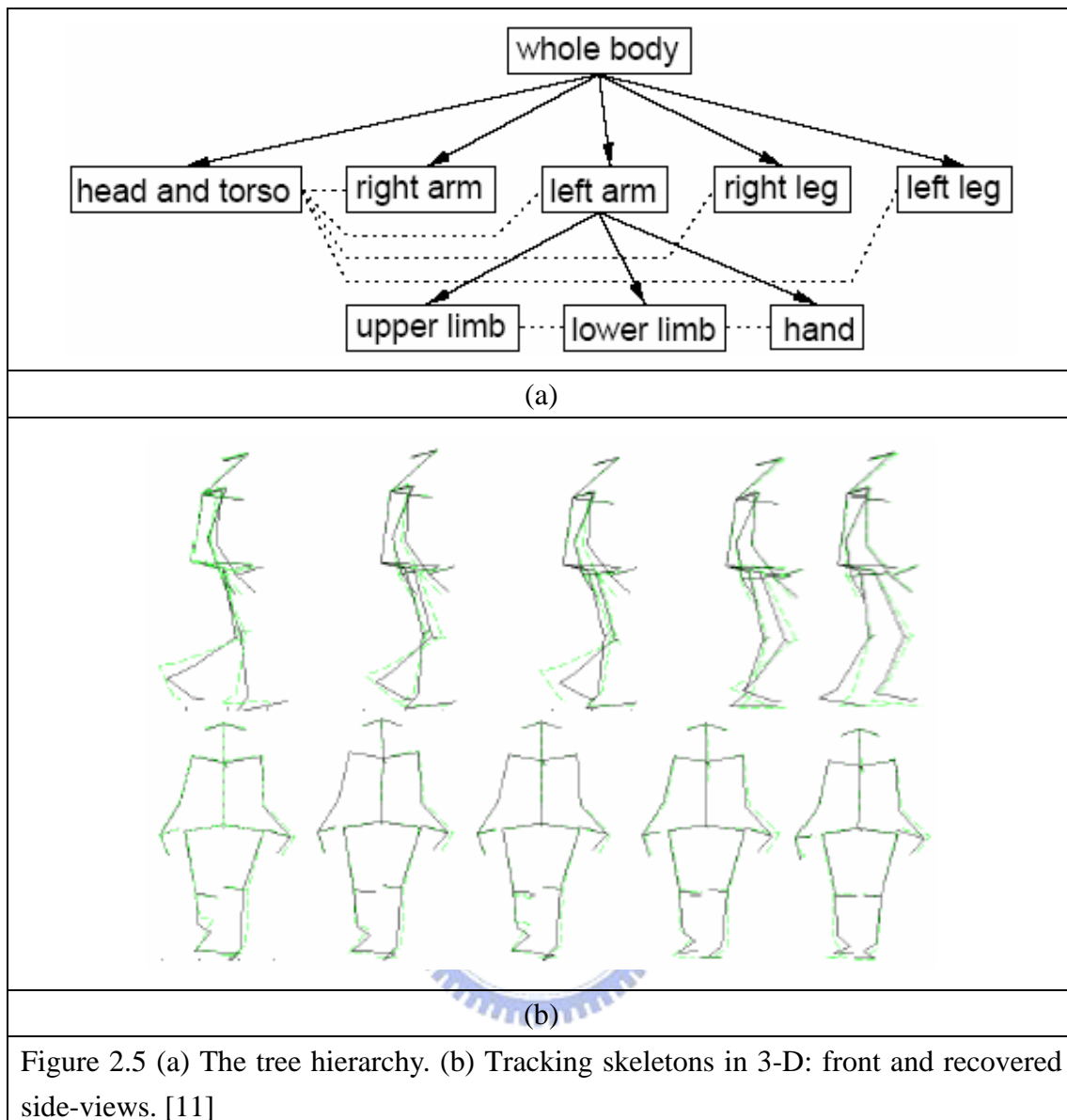


To find the complete contour of moving objects, the active contour approach, which is commonly called “snake”, is widely used. Snakes are deformable contours that move under the influence of image-intensity forces, subject to certain internal deformation constraints. In [10], the authors proposed a kalman snake model in the spatio-velocity space to track non-rigid moving targets.

In contrast to region-based tracking algorithms, these active contour-based algorithms describe objects in a simpler and more effective way and can thus reduce computational complexity.

### 2.2.3 Model-Based Tracking

Model-based tracking methods build a model in advance and match the moving object to the model. A motion model is also established to incorporate with a search strategy. To track different objects, many types of models have been proposed. In [11], a hierarchical model is proposed to describe human dynamics. They regard the transition from one pose to the next as the *dynamics* of the action, and encode this transition using a hidden Markov model (HMM). In this approach, the models, both poses and dynamics, are trained from real data. Then, they describe the model of valid poses, and then move on to describe the HMMs for dynamics. This hierarchical model tracks skeletal poses, this tracking method is largely independent of image modality.



## 2.2.4 Feature-Based Tracking

In contrast to model-based tracking methods, feature-based tracking builds a model according to the moving object's features. There are lots of features that can help us in tracking objects, like edges, corners, color distribution, skin tone and human eyes. An active template which characterizes regional and structural features, such as texture, shape and color, is proposed to track moving objects in [12]. In this approach, the authors design an energy function and adapt the model dynamically by minimizing the energy in order to track non-rigid targets.

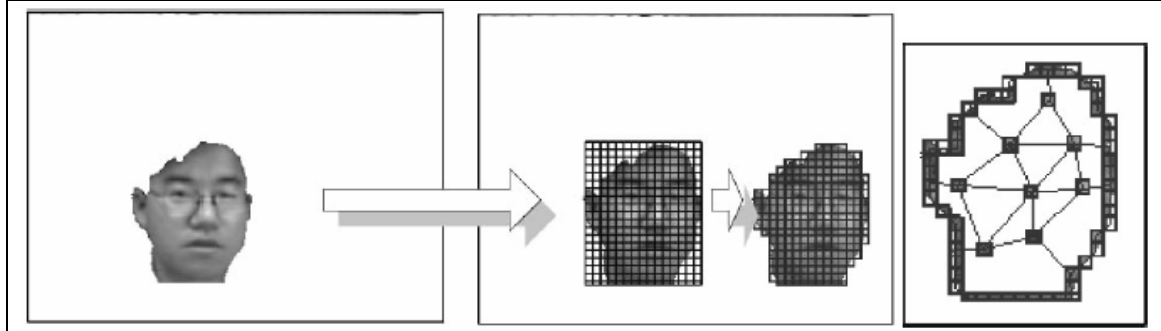


Figure 2.6 Example of a model graph which contains boundary cells and internal cells. [12]

In recent years, an efficient algorithm called “mean-shift” is widely used for motion tracking. In [1], a target model is constructed by calculating moving object’s color histogram. In other words, the target model uses the color distribution status as features. The target model is defined as

$$q_u = C \sum_{i=1}^n k(\|\bar{x}_i\|^2) \delta[b(\bar{x}_i) - u], \quad \text{Eq. 2-4}$$

where  $C$  is the normalization constant to ensure  $\sum_{u=1}^m \hat{q}_u = 1$ , and  $k(\|x\|^2)$  is the kernel profile.

Similarly, a target candidate is defined as

$$p_u(\bar{y}) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{\bar{y} - \bar{x}_i}{h}\right\|^2\right) \delta[b(\bar{x}_i) - u]. \quad \text{Eq. 2-5}$$

The Bhattacharyya coefficient is used to derive the similarity between the target model and the target candidate. The coefficient is defined as

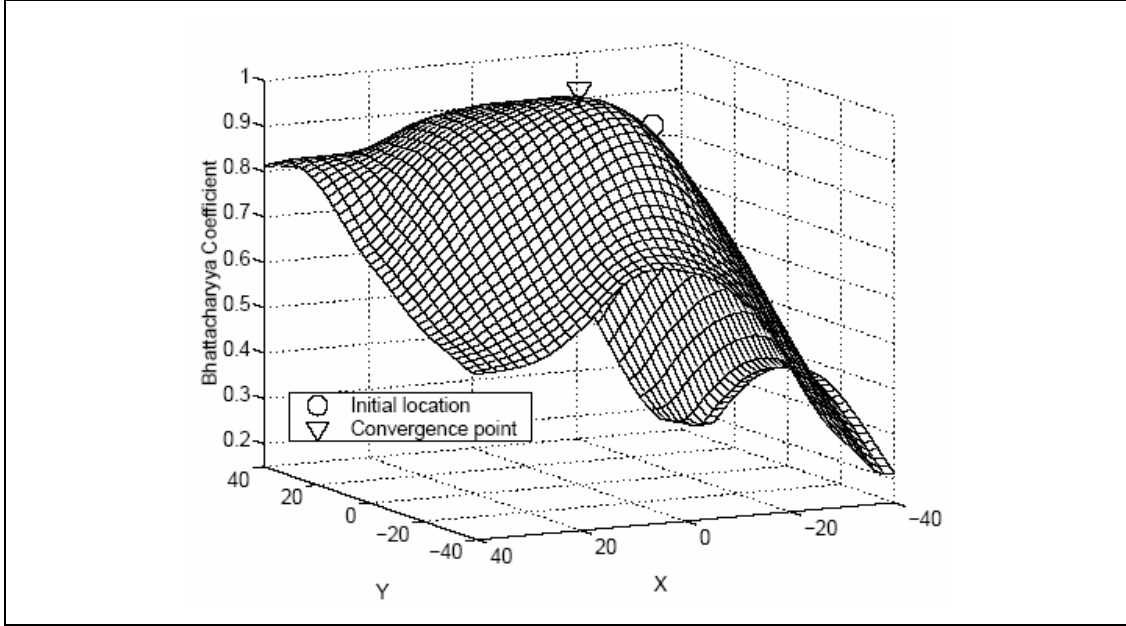
$$\rho(\bar{y}) \equiv \rho[p(\bar{y}), q] = \sum_{u=1}^m \sqrt{p_u(\bar{y}) q_u}. \quad \text{Eq. 2-6}$$

After the maximization of the similarity function, we have

$$\bar{y}_1 = \frac{\sum_{i=1}^{n_h} \bar{x}_i w_i g\left(\left\|\frac{\bar{y}_0 - \bar{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\bar{y}_0 - \bar{x}_i}{h}\right\|^2\right)}, \quad \text{Eq. 2-7}$$

where  $w_i = \sum_{u=1}^m \sqrt{\frac{q_u}{p_u(\bar{y}_0)}} \delta[b(\bar{x}_i) - u]$ , and  $g(x) = -k'(x)$  represents the shadow kernel. In

this procedure, the kernel is iteratively moved from the current location  $\bar{y}_0$  to the new location  $\bar{y}_1$  according to Eq. 2-7.



(a)



(b)

Figure 2.7 (a) The similarity surface, the initial and final locations of mean-shift iterations. (b) Tracking results. [1]

Although the mean-shift approach is relatively an efficient and robust method for motion tracking, there are still some drawbacks. In recent years, many researches try to reform the mean-shift algorithm based on the following issues.

**(a) Accuracy improvement.**

In [13], the authors design some experiments to figure out whether the adopted similarity measure is appropriate or not. The simulations indicate that the Bhattacharyya and K-L distances are inaccurate in higher dimensions and the computations in higher dimensions are instable in the sense that repeated computations using different samples may yield varying results. Hence, they redefine the target model as

$$\hat{q}_x(\bar{x}, \bar{u}) = \frac{1}{N} \sum_{i=1}^N w \left( \left\| \frac{\bar{x} - \bar{x}_i}{\sigma} \right\|^2 \right) k \left( \left\| \frac{\bar{u} - \bar{u}_i}{h} \right\|^2 \right), \quad \text{Eq. 2-8}$$

where  $\bar{x}$  represents the spatial location and  $\bar{u}$  is the corresponding feature vector.

Similarly, the target candidate is redefined as

$$\hat{p}_y(\bar{y}, \bar{v}) = \frac{1}{M} \sum_{j=1}^M w \left( \left\| \frac{\bar{y} - \bar{y}_j}{\sigma} \right\|^2 \right) k \left( \left\| \frac{\bar{v} - \bar{v}_j}{h} \right\|^2 \right). \quad \text{Eq. 2-9}$$

The similarity between the target model and the candidate in the joint feature-spatial space is defined to be

$$J(p_x, q_y) = \frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M w \left( \left\| \frac{\bar{x}_i - \bar{y}_j}{\sigma} \right\|^2 \right) k \left( \left\| \frac{\bar{u}_i - \bar{v}_j}{h} \right\|^2 \right). \quad \text{Eq. 2-10}$$

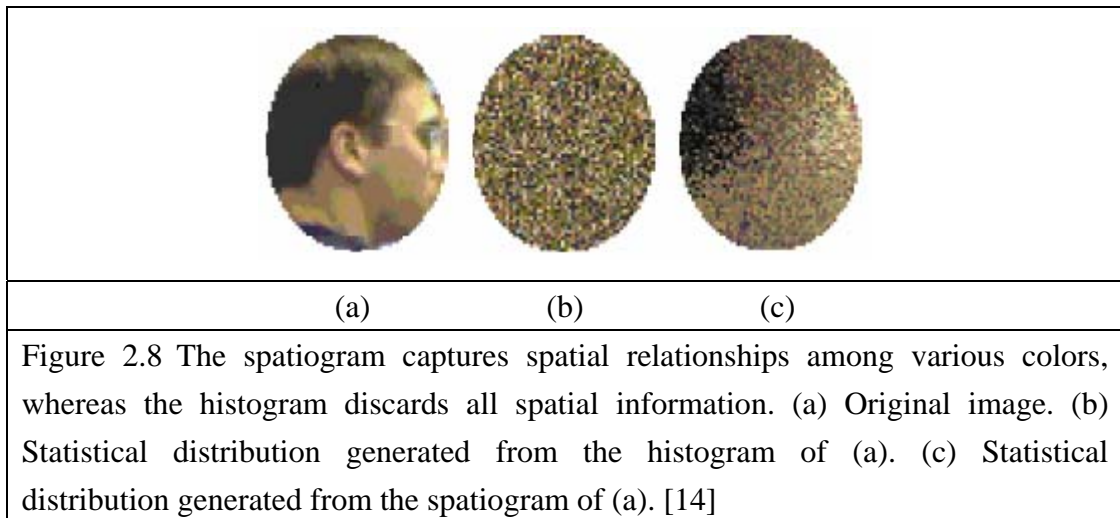
Even though the experimental results show that the tracking accuracy is greatly improved, the computational complexity for the direct evaluation of this similarity measure could be very high.

### (b) Usage of spatial information

All the aforementioned target models lack spatial color distribution information. In order to keep more useful features, [14] proposed a new representation, called spatiogram, for the target model and the candidate. Spatiograms offer a richer representation and may capture not only the values of the pixels but also their spatial relationships. In their approach, a spatiogram is defined as

$$h_p(b) = \langle n_b, \mu_b, \Sigma_b \rangle, \quad b=1, \dots, M, \quad \text{Eq. 2-11}$$

where  $n_b$  is the number of pixels in the  $b$ th bin, and  $\mu_b$  and  $\Sigma_b$  represent the mean vector and covariance matrices, respectively. A mean-shift process is also developed to do motion tracking. Simulations in [13] show that this spatiogram approach offers more robust tracking than the traditional histogram-based approaches.



### (c) Scale and orientation selection

Scale selection is also a problem. If the kernel size is too large, the tracking window will contain many background pixels as well as foreground object pixels. Since in this case the data histogram will get polluted with background data, this large kernel window may cause incorrect tracking. On the contrary, choosing a kernel size that is too small may suffer from poor object localization.

In [1], the “plus or minus 10 percent” scale adaptation method is used to estimate the optimized scale, but the computational complexity is three times than before. In [13] and [15], the authors treat the scale as a variable in the tracking algorithm and update the scale by applying the mean-shift procedure through the scale axis.



Figure 2.9 Scale-space mode-tracking method. The person is tracked well, both spatially and in scale. [15]

On the other hand, the orientation of target is also important. Bad orientation estimation will cause the target information to be polluted by noisy background pixels. Rectangular bounding box can't help in estimating object's orientation. In [16], the authors use a bi-variant Gaussian profile as the kernel in the mean-shift procedure. By calculating the covariance matrix of all pixels belonging to the moving object, we can obtain the orientation and scale at the same time.

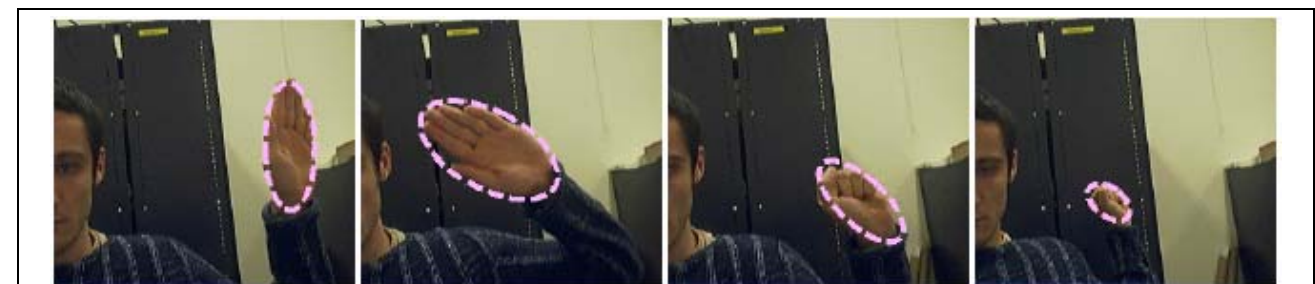


Figure 2.10 Updating both size and orientation in motion tracking. [16]

### (d) Feature selection

Object tracking is cast as a local discrimination problem to distinguish foreground objects from the background. During the tracking process, distractions due to background could easily distract the mean-shift window and cause failure in tracking. In [17], the authors develop a strategy to select features that can best discriminate foreground pixels from background pixels. To quantify the discrimination of a feature, a two-class variance ratio is used as the feature score. Base on the feature score, the most discriminative features can be

chosen for a more robust tracking.

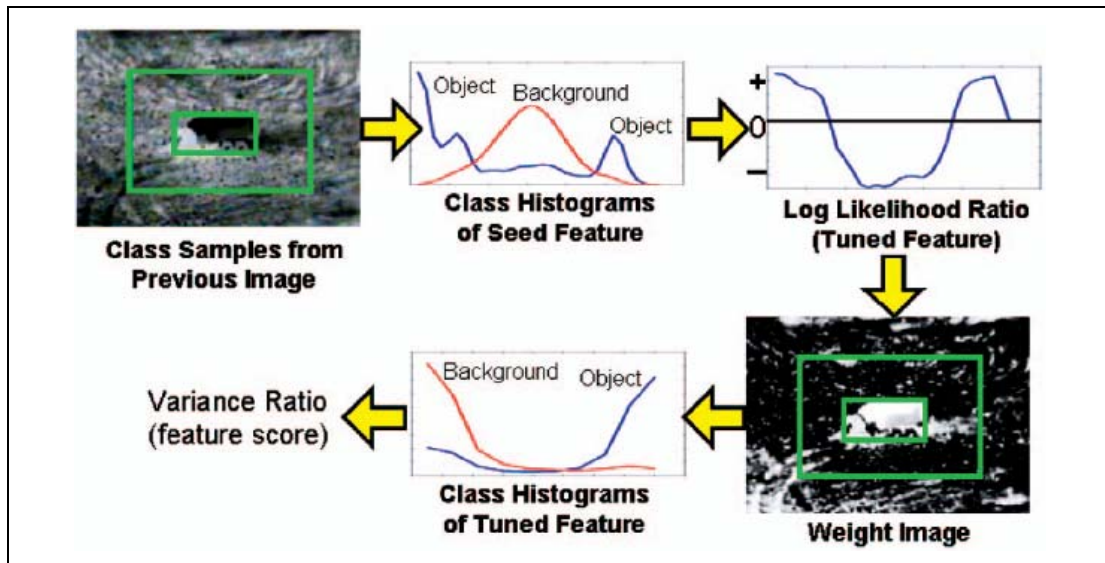


Figure 2.11 A flow chart in [17], which describes how to obtain the feature score.

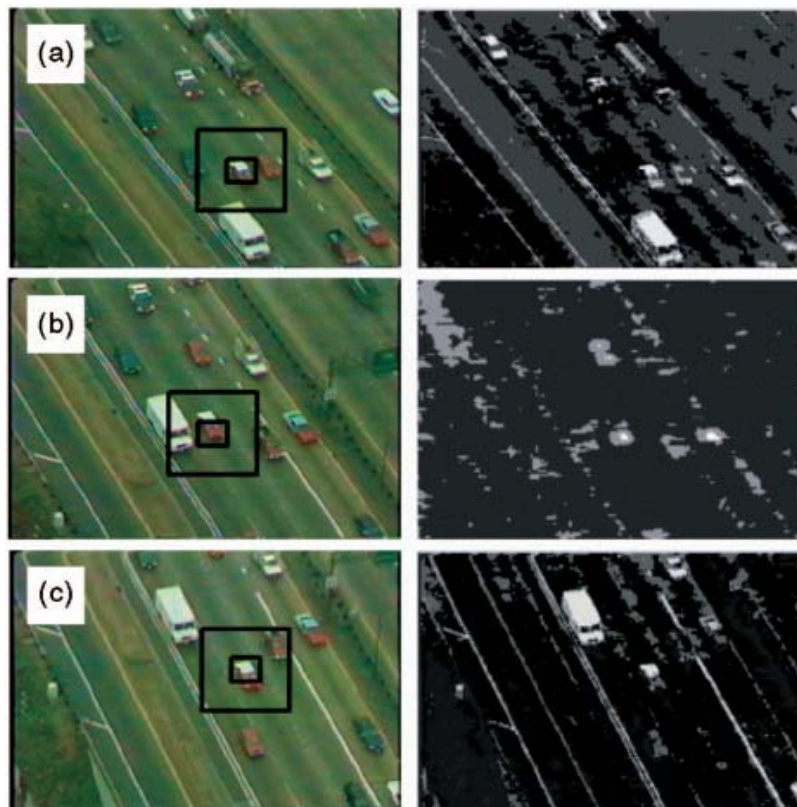


Figure 2.12 Example of feature adaptation to avoid distractions. Left column: video frame with the object/background windows overlaid. Right column: weight image from the top-ranked tracking features. [17]

# Chapter 3.

## Mean-Shift Tracking and Multi-Blob Model

Mean-shift is a powerful mathematical tool for object tracking problem, however, there still exist many drawbacks as mentioned in Section 2.2.4. In this chapter, we propose a new mean-shift-based approach trying to avoid these shortcomings.

### 3.1 Traditional Mean-Shift Tracking

Mean-shift is a technique which can find the location of local maximum without knowing the overall distribution. In object tracking problems, mean-shift is usually used to find the local maximum of the similarity surface.

#### 3.1.1 Building Models and Measuring Similarity

To define the similarity between two objects, we have to define the features of an object first. Recall Eq. 2-4 and Eq. 2-5, which are used to build models. In [1], the object is represented using a  $16 \times 16 \times 16$  histogram in the RGB space and the model becomes

$$q_u = C \sum_{i \in I} k(\|\bar{x}_i\|^2) \delta[b(\bar{x}_i) - u], \quad u = 1, \dots, 16 \times 16 \times 16, \quad \text{Eq. 3-1}$$

where  $I$  represents the set of pixels belonging to the object, and  $k(\cdot)$  is the kernel function for the spatial information.

Based on the model designed to represent features of the object, we may choose an appropriate measurement to evaluate the similarity. In [13], the histogram is considered as a distribution in the feature space. Existing mean-shift trackers use the Kullback-Leibler distance and Bhattacharyya distance to measure the similarity between distributions. The Kullback-Leibler distance between two distributions is define as

$$D(\bar{y}) = \sum_{u=1}^{16 \times 16 \times 16} p_u(\bar{y}) \log \frac{p_u(\bar{y})}{q_u}. \quad \text{Eq. 3-2}$$

On the other hand, the Bhattacharyya distance is



$$B(\bar{y}) = \sqrt{1 - \sum_{u=1}^{16 \times 16 \times 16} \sqrt{p_u(\bar{y})} q_u} . \quad \text{Eq. 3-3}$$

The figure shown below represents the flow of calculating similarity.

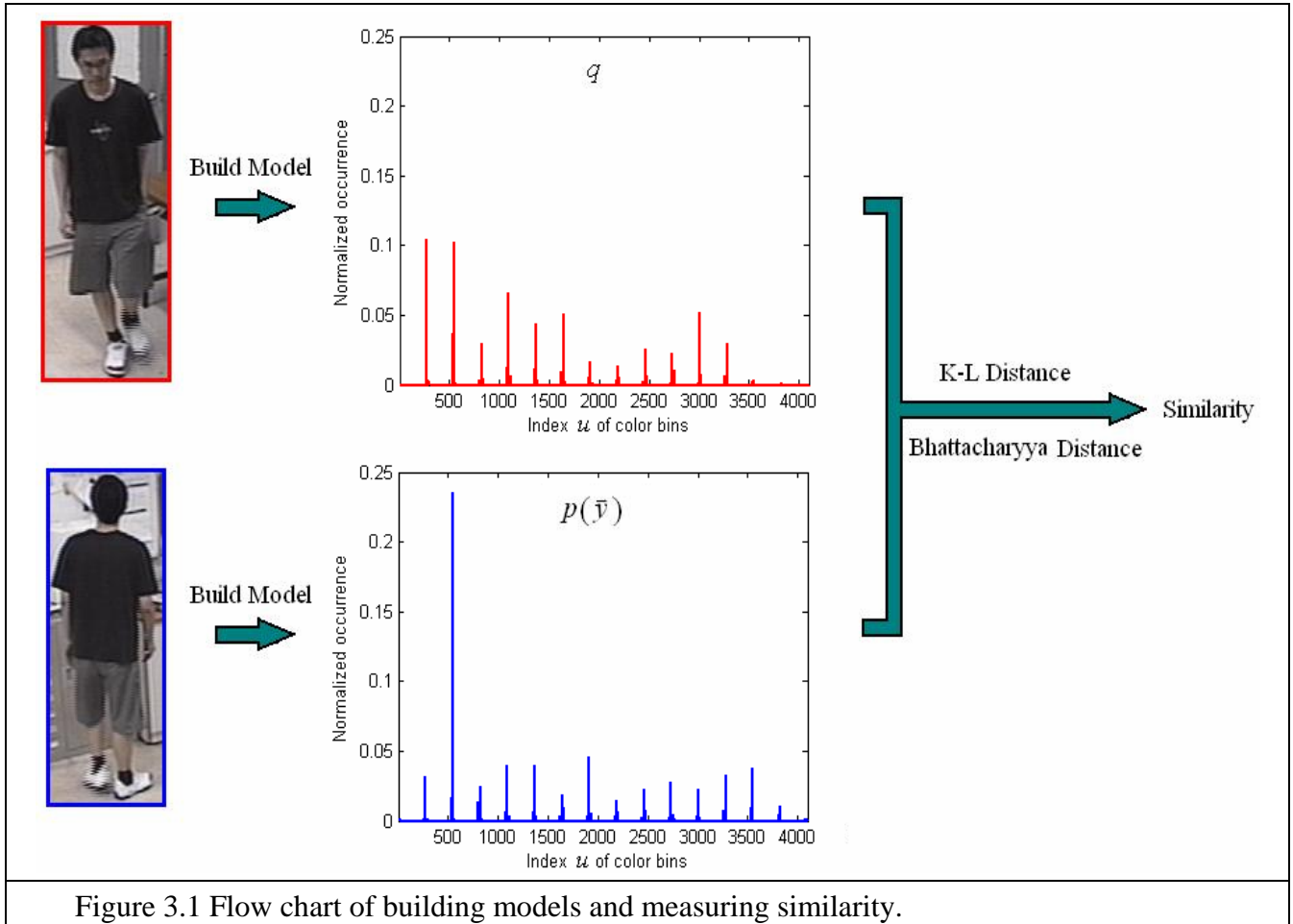
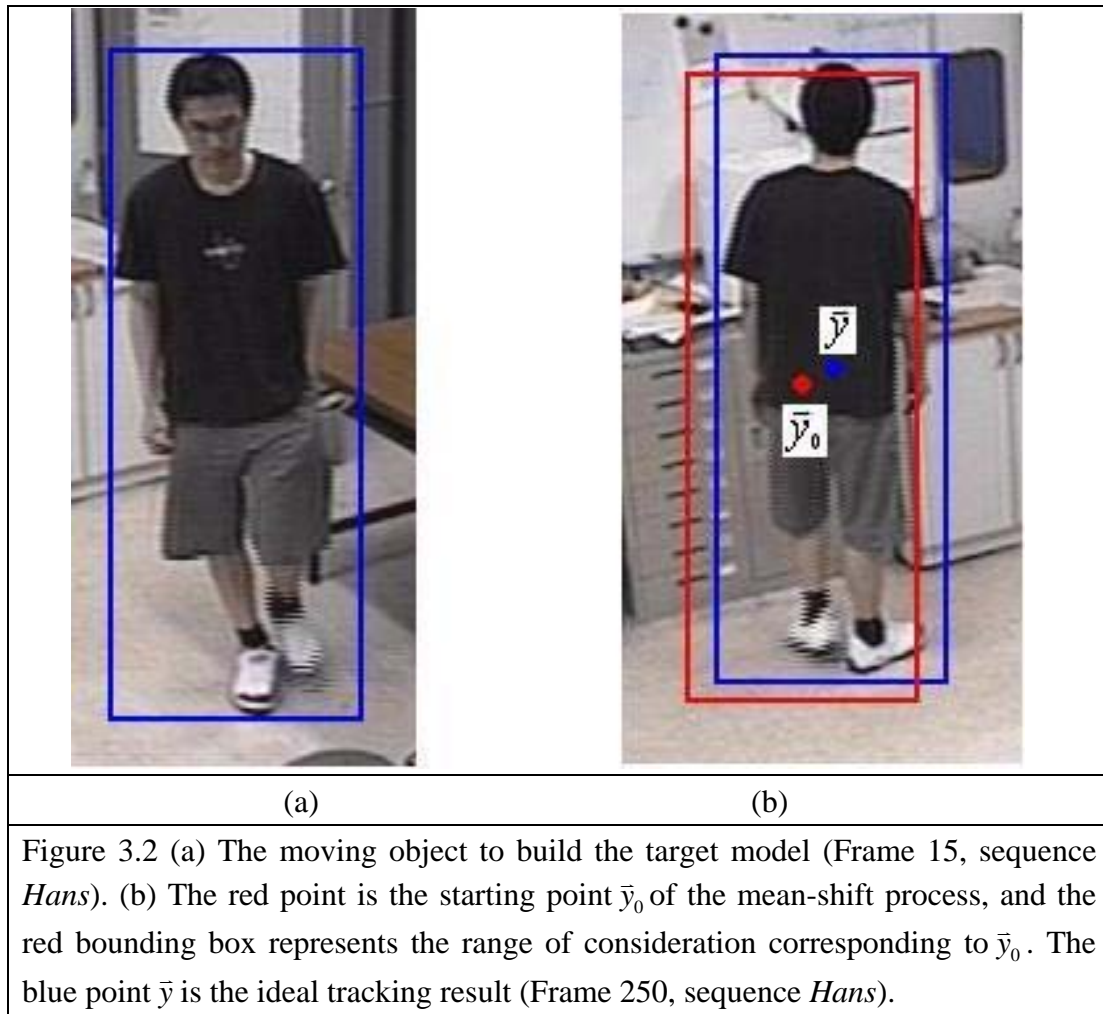


Figure 3.1 Flow chart of building models and measuring similarity.

### 3.1.2 Object Tracking Procedure

[1] uses the Bhattacharyya distance to measure similarity. In Eq. 3-3, minimizing the distance is equivalent to maximizing the Bhattacharyya coefficient, where the coefficient is expressed in Eq. 2-6. The search for the new target location in the current frame starts at the location  $\bar{y}_0$  of the target in the previous frame. As shown in Figure 3.2(b), the location  $\bar{y}$  has the maximum similarity.



Since the motion between two frames is small,  $\bar{y}$  is usually close to  $\bar{y}_0$  but the precise location is unknown. Thus, we may use the first order Taylor expansion to represent the Bhattacharyya coefficient  $\rho(\bar{y})$  in terms of  $\rho(\bar{y}_0)$ .

$$\begin{aligned}
\rho(\bar{y}) &= \sum_{u=1}^m \sqrt{p_u(\bar{y})q_u} \\
&\approx \sum_{u=1}^m \sqrt{p_u(\bar{y}_0)q_u} + \frac{1}{2} \sum_{u=1}^m \sqrt{\frac{q_u}{p_u(\bar{y}_0)}} (p_u(\bar{y}) - p_u(\bar{y}_0)) \\
&= \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(\bar{y}_0)q_u} + \frac{1}{2} \sum_{u=1}^m p_u(\bar{y}) \sqrt{\frac{q_u}{p_u(\bar{y}_0)}} \\
&= \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(\bar{y}_0)q_u} + \frac{C_h}{2} \sum_{i=1}^{n_h} \sum_{u=1}^m k\left(\left\|\frac{\bar{y} - \bar{x}_i}{h}\right\|^2\right) \sqrt{\frac{q_u}{p_u(\bar{y}_0)}} \delta[b(\bar{x}_i) - u] \\
&= \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(\bar{y}_0)q_u} + \frac{C_h}{2} \sum_{i=1}^{n_h} w_i k\left(\left\|\frac{\bar{y} - \bar{x}_i}{h}\right\|^2\right), \tag{Eq. 3-4}
\end{aligned}$$

where  $w_i = \sum_{u=1}^m \sqrt{\frac{q_u}{p_u(\bar{y}_0)}} \delta[b(\bar{x}_i) - u]$ . The first term in Eq. 3-4 is independent of  $\bar{y}$ ; hence, we have to maximize the second term with respect to the vector  $\bar{y}$ . Note that the second term represents the density estimate computed with the kernel profile  $k(\cdot)$  at  $\bar{y}$  in the current frame, with the data being weighted by  $w_i$ . Denote the second term as  $J(\bar{y})$ . The gradient of  $J(\bar{y})$  with respect to  $\bar{y}$  is expressed as

$$\begin{aligned}
\nabla J(\bar{y}) &= \frac{C_h}{2} \sum_{i=1}^{n_h} w_i k'\left(\left\|\frac{\bar{y} - \bar{x}_i}{h}\right\|^2\right) \cdot \frac{2}{h} (\bar{y} - \bar{x}_i) \\
&= \frac{C_h}{h} \bar{y} \sum_{i=1}^{n_h} w_i k'\left(\left\|\frac{\bar{y} - \bar{x}_i}{h}\right\|^2\right) - \frac{C_h}{h} \sum_{i=1}^{n_h} w_i \bar{x}_i k'\left(\left\|\frac{\bar{y} - \bar{x}_i}{h}\right\|^2\right). \tag{Eq. 3-5}
\end{aligned}$$

By letting Eq. 3-5 be 0, we have

$$\bar{y}_1 = \frac{\sum_{i=1}^{n_h} w_i \bar{x}_i g\left(\left\|\frac{\bar{y}_0 - \bar{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\bar{y}_0 - \bar{x}_i}{h}\right\|^2\right)}, \tag{Eq. 3-6}$$

where  $g(\cdot) = -k'(\cdot)$ , named the shadow kernel, is also the profile of the radial basic function,

and  $w_i g\left(\left\|\frac{\bar{y}_0 - \bar{x}_i}{h}\right\|^2\right)$  can be considered as the convolution of  $w_i$  and the shadow kernel.

Based on Eq. 3-6, an iterative procedure is obtained. To achieve better performance, the kernel function should be selected carefully. Kernels with the Gaussian profile or the Epanechnikov profile are recommended. For Gaussian kernel, the derivative of the profile,  $g(\cdot)$ , is still a Gaussian function. For the Epanechnikov kernel, we have

$$k(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2)(1-x) & , \text{ if } x \leq 1 \\ 0 & , \text{ otherwise } \end{cases} \quad \text{Eq. 3-7}$$

where  $c_d$  is the volume of a unit  $d$ -dimension sphere and  $d$  equals to 2 in this case. The derivative of the profile is constant and Eq. 3-6 reduces to

$$\bar{y}_1 = \frac{\sum_{i=1}^{n_h} w_i \bar{x}_i}{\sum_{i=1}^{n_h} w_i} \quad \text{Eq. 3-8}$$



By using the Gaussian kernel in Eq. 3-6, a mean-shift tracking process can be build. Figure 3.3 shows the mean-shift tracking flow, where the detection step is done by hand.

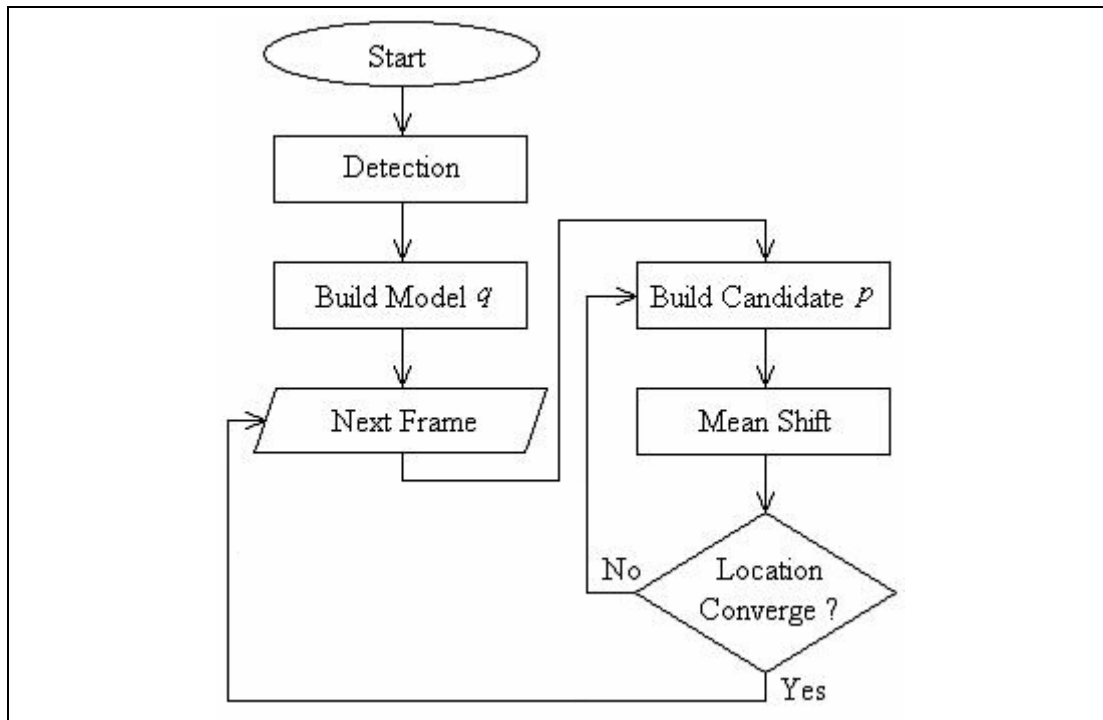
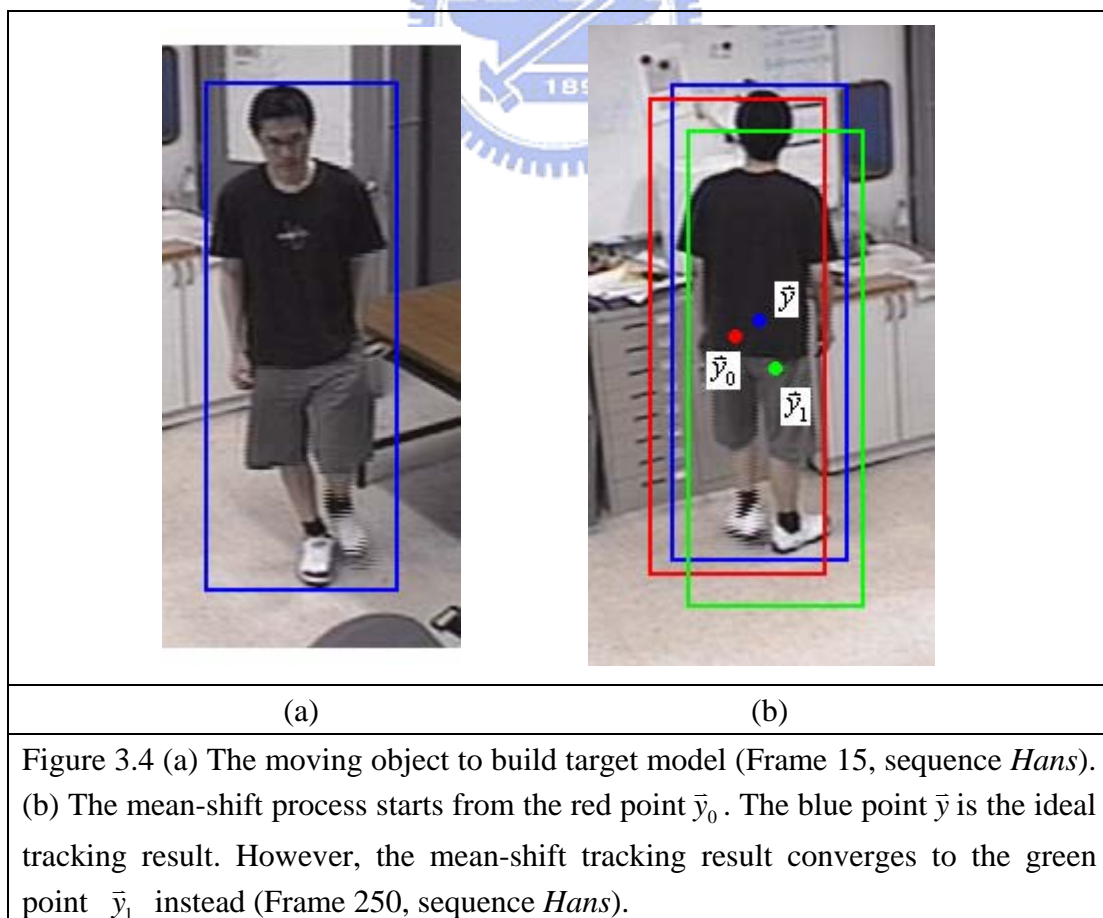


Figure 3.3 Mean-shift tracking flow

Employing the mean-shift tracking algorithm and letting  $\bar{y}_0$  in Figure 3.2 (b) be the start point of the mean-shift process, the tracking result is shown below.



It is observed that the green bounding box in Figure 3.4 (b) which represents the tracking result is not accurate. Theoretically,  $\bar{y}_1$  should have the maximal Bhattacharyya coefficient. The color histogram of the three bounding boxes in Figure 3.4 (b) is shown below. By checking the target model and the computed Bhattacharyya coefficient,  $\bar{y}_1$  indeed has the maximal similarity value and is at the peak of the similarity surface.

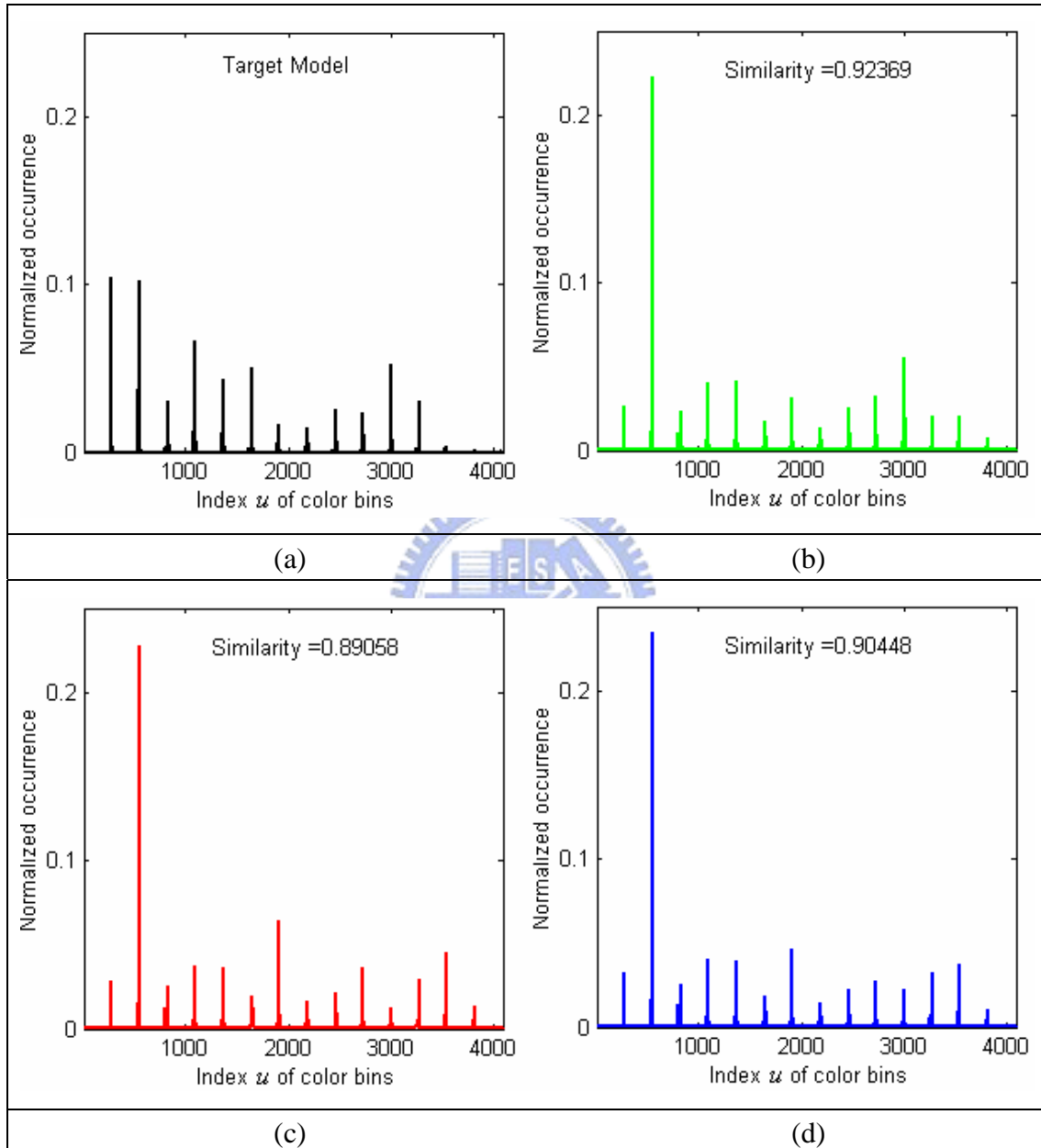


Figure 3.5 (a) The target model. (b) The color histogram and the similarity value with respect to the tracking result  $\bar{y}_1$ . (c) The color histogram and the similarity value with respect to the starting point  $\bar{y}_0$ . (d) The color histogram and the similarity value with respect to the expected tracking result  $\bar{y}$ .

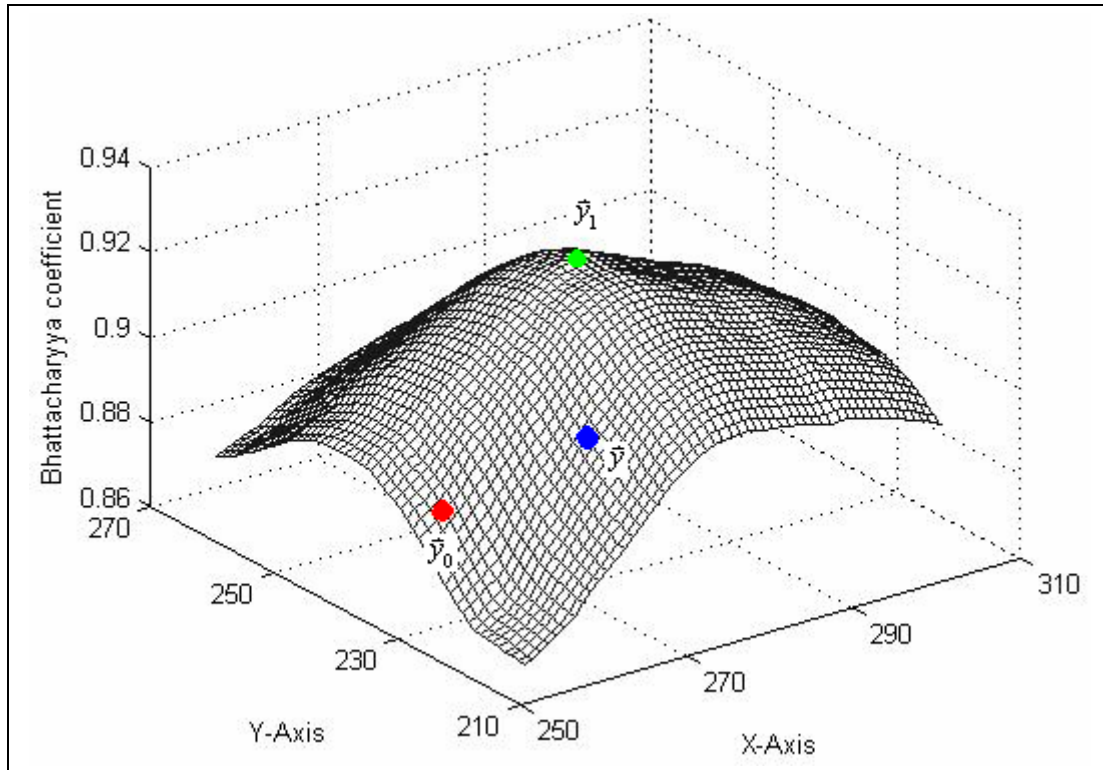


Figure 3.6 The similarity surface, the initial and final locations of the mean-shift process.  $\bar{y}$  is with respect to the expected tracking result.

### 3.1.3 Discussion

To sum up, building an appropriate model is to reduce redundant information but keep useful features for tracking. Based on the chosen model, we then choose an appropriate similarity measure. We can find the maximal similarity value by taking differentiation over the Bhattacharyya coefficient with respect to  $\bar{y}$ . With to the use of the radial basic function, this differentiation operation will deduce the iterative mean-shift formula.

According to the simulation results in Section 3.1.2, color histogram doesn't seem to be an ideal representation of object appearance since the spatial information is discarded. Lack of spatial information, together with the distraction caused by background pixels, causes poor accuracy in tracking. By building a target model to contain more information, both in the spatial domain and the feature domain, and developing an appropriate similarity measure, we can improve the tracking performance.

## 3.2 Multi-Blob Model Based Mean-Shift

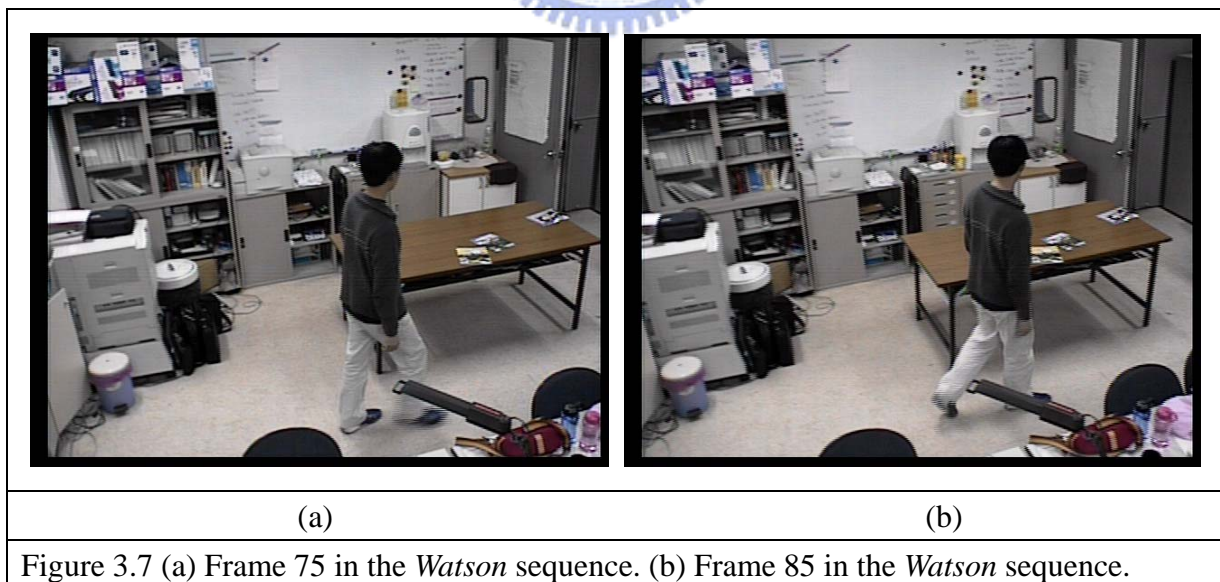
### Tracking

Building a model to represent the features of the target can reduce the redundant data and computational complexity. In this thesis, we propose a new target model, called a multi-blob model, to represent the features of object. Based on the multi-blob model, we design a similarity measure and a corresponding mean-shift tracking algorithm.

#### 3.2.1 Moving Object Detection

To build a target model, we have to know the position of the moving object first. In our tracking system, a pan-tilt-zoom camera is used. Thus, the background may change. Here, we employ a motion compensation technique to detect moving objects in the frame. Our goal is to detect the region of a moving object in a rough but efficient manner.

We choose a small number of blocks, say three or four, and estimate their motion vectors. We then compensate the motion of all blocks based on these motion vectors. The residual between the compensated frame and the reference frame will indicate the area of the moving object. Figure 3.7 shows two frames in the *Watson* sequence, with image size being  $480 \times 640$ . There is only one moving object in these frames and the camera pans during the capture of these two images.



To define the location of the moving object by using a covariance ellipse, we have to gather the statistics of the foreground region. Figure 3.8 shows the flow of calculating mean and covariance matrix of the foreground region.



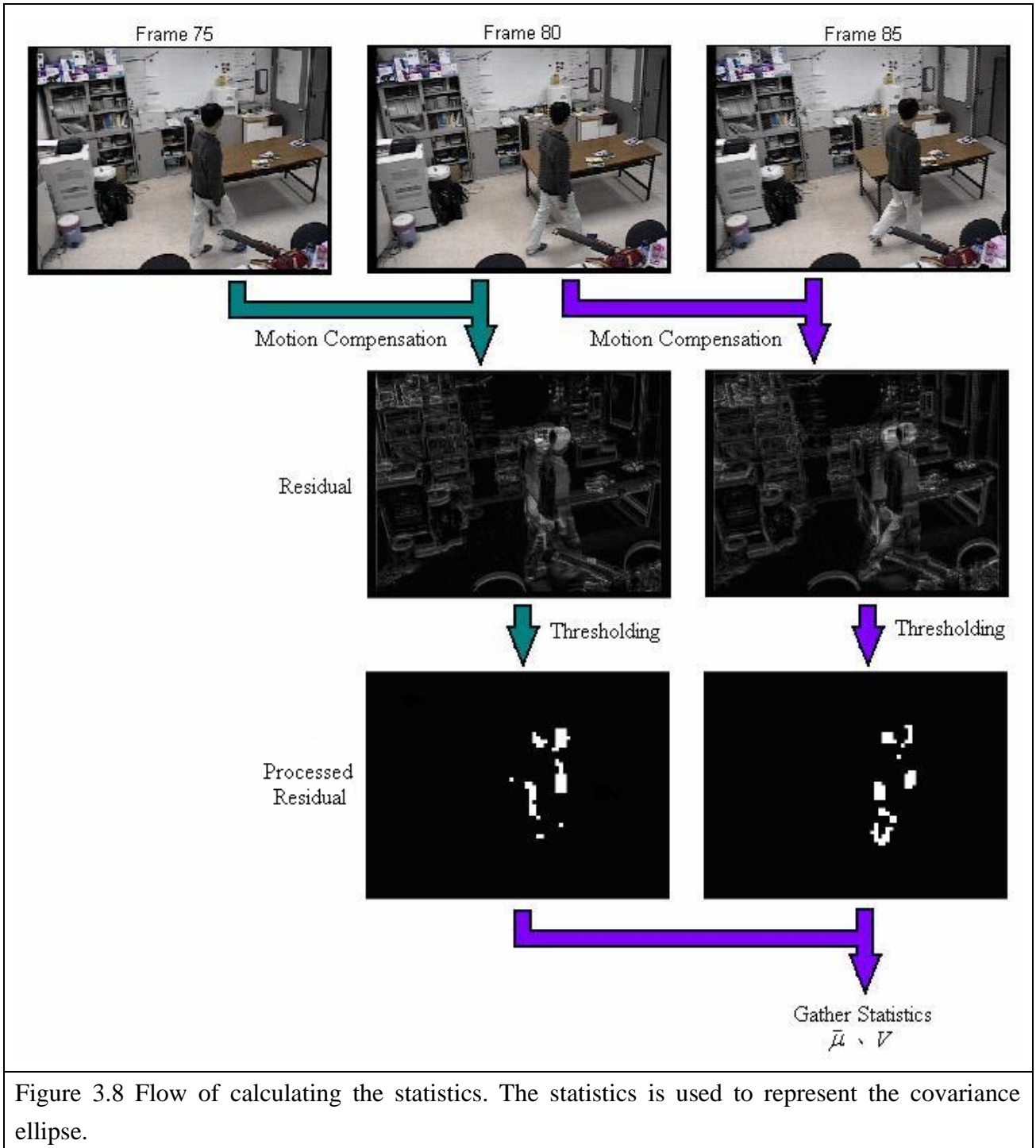


Figure 3.8 Flow of calculating the statistics. The statistics is used to represent the covariance ellipse.

From the flow shown in Figure 3.8, we have  $\bar{\mu} = [408.97 \ 273.04]$  and  $V = \begin{bmatrix} 1217.1 & -1019.7 \\ -1019.7 & 6246.5 \end{bmatrix}$ . The bounding ellipse can be represented by both  $[\sigma_x \ \sigma_y \ \rho]$  and  $[a \ b \ \theta]$ , where  $a$  represents the major axis length,  $b$  represents the minor axis length,  $\theta$  is the offset angle measured clockwise from the y-axis, and the

focal points are  $f_1(c \sin \theta, c \cos \theta)$  and  $f_2(-c \sin \theta, -c \cos \theta)$ . Figure 3.9 shows the relation between these two tuples.

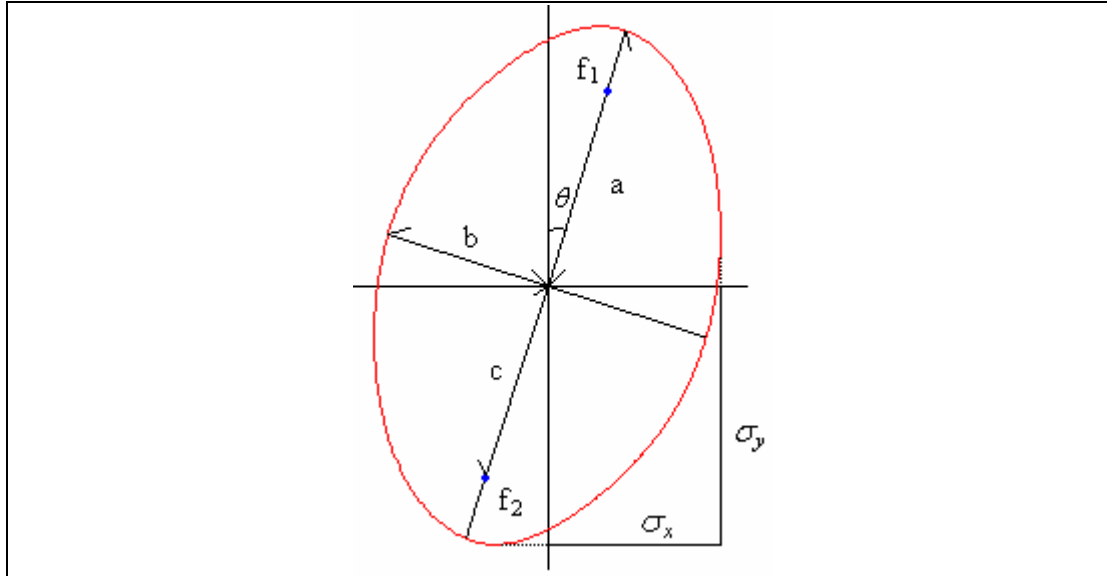


Figure 3.9 Relation between  $[\sigma_x \ \sigma_y \ \rho]$  and  $[a \ b \ \theta]$ .

From the mathematical definition of ellipse, we have

$$\sqrt{(x - c \sin \theta)^2 + (y - c \cos \theta)^2} + \sqrt{(x + c \sin \theta)^2 + (y + c \cos \theta)^2} = 2a$$

$$\Rightarrow -2c \sin \theta x - 2c \cos \theta y = 4a^2 + 4a \sqrt{(x + c \sin \theta)^2 + (y + c \cos \theta)^2} + 2c \sin \theta x + 2c \cos \theta y$$

$$\Rightarrow a^2 \left[ (x + c \sin \theta)^2 + (y + c \cos \theta)^2 \right] = \left[ a^2 + c \sin \theta x + c \cos \theta y \right]^2$$

$$\Rightarrow x^2 + c^2 \sin^2 \theta + y^2 + c^2 \cos^2 \theta = a^2 + \frac{c^2}{a^2} \left[ \sin^2 \theta x^2 + 2 \sin \theta \cos \theta xy + \cos^2 \theta y^2 \right]$$

$$\Rightarrow (a^2 - c^2 \sin^2 \theta) x^2 - 2c^2 \sin \theta \cos \theta xy + (a^2 - c^2 \cos^2 \theta) y^2 = \text{constant A} \quad \text{Eq. 3-9}$$

Regarding the exponent of bi-variance Gaussian function, we have

$$\frac{x^2}{\sigma_x^2} - \frac{2\rho xy}{\sigma_x \sigma_y} + \frac{y^2}{\sigma_y^2} = d^2$$

$$\Rightarrow \sigma_y^2 x^2 - 2\rho \sigma_x \sigma_y xy + \sigma_x^2 y^2 = \text{constant B} \quad \text{Eq.}$$

Comparing Eq. 3-9 and Eq. 3-10, the major axis length, minor axis length, offset angle and eccentricity can be obtained.

$$\sigma_y^2 = (a^2 - c^2 \sin^2 \theta)$$

$$\sigma_x^2 = (a^2 - c^2 \cos^2 \theta)$$

$$\Rightarrow (\sigma_y^2 - \sigma_x^2) = c^2 \cos 2\theta \quad \text{Eq. 3-11}$$

$$2\sigma_{xy} = 2\rho\sigma_x\sigma_y = c^2 \sin 2\theta \quad \text{Eq. 3-12}$$

$$\Rightarrow \theta = \frac{1}{2} \sin^{-1} \frac{2\sigma_{xy}}{c^2} = \frac{1}{2} \cos^{-1} \frac{(\sigma_y^2 - \sigma_x^2)}{c^2} \quad \text{Eq. 3-13}$$

From Eq. 3-11 and Eq. 3-12,

$$c^2 = \sqrt{(\sigma_y^2 - \sigma_x^2)^2 + 4\sigma_{xy}^2} \quad \text{Eq. 3-14}$$

$$\therefore a^2 = \sigma_y^2 + c^2 \sin^2 \theta = \sigma_x^2 + c^2 \cos^2 \theta \quad \text{Eq. 3-15}$$

$$b^2 = a^2 - c^2 = \sigma_y^2 - c^2 \cos^2 \theta = \sigma_x^2 - c^2 \sin^2 \theta \quad \text{Eq. 3-16}$$

Furthermore, the area of the ellipse is  $\pi ab$ . From Eq. 3-15 and Eq. 3-16, we have

$$a^2 b^2 = (\sigma_y^2 + c^2 \sin^2 \theta)(\sigma_x^2 - c^2 \cos^2 \theta)$$

$$= \sigma_y^4 - c^2 \sigma_y^2 \cos 2\theta - \frac{c^4}{4} \sin^2 2\theta$$

$$= \sigma_y^4 - \sigma_y^2 (\sigma_y^2 - \sigma_x^2) - \sigma_{xy}^2$$

$$= \det(V)$$

$$\therefore \pi ab = \pi \sqrt{\det(V)}$$

Eq. 3-17

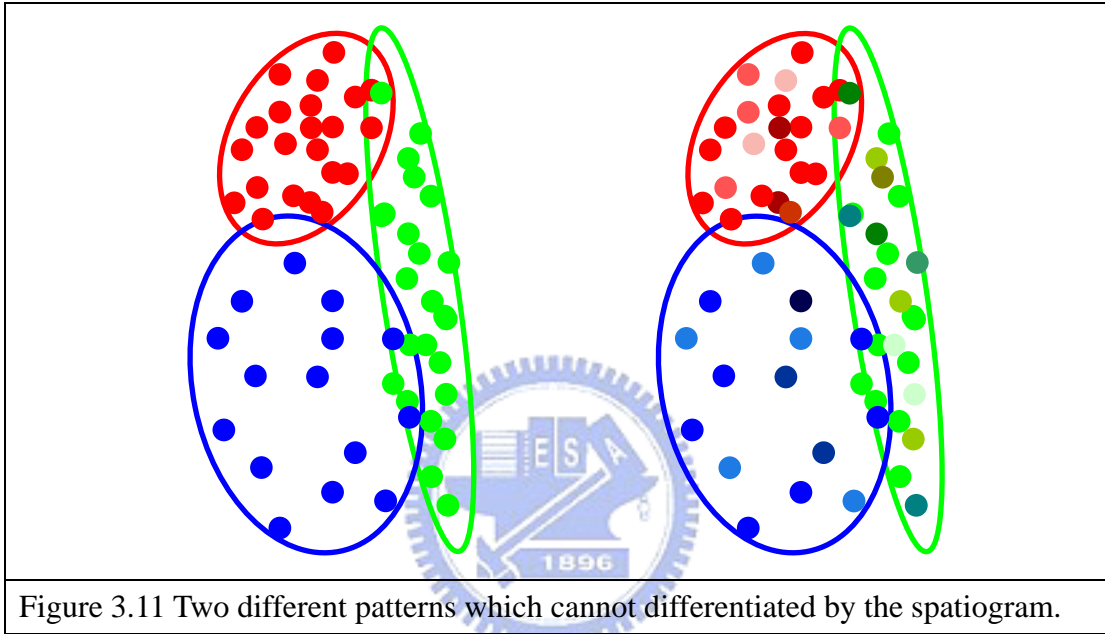
Based on the above equations, we can mark the 2-sigma contour as a bounding ellipse to indicate the foreground region. Figure 3.10 shows the detection result of the *Watson* sequence where  $\bar{\mu} = [408.97 \quad 273.04]$  and  $V = \begin{bmatrix} 1217.1 & -1019.7 \\ -1019.7 & 6246.5 \end{bmatrix}$ .



Figure 3.10 Using a bounding ellipse to define the moving object.

### 3.2.2 Multi-Blob Model

Color histogram is a robust representation of the object's color occurrence. In [14], spatiogram is considered as a generalized histogram containing spatial information. To allocate pixels into certain color bins can be regarded as doing quantization in the feature space. However, there may be variations between pixels belonging to the same color bin. Figure 3.11 shows two different patterns which have the same values of spatiogram.



To preserve feature domain information, we gather statistics in the feature space for each color bin. That is, we define the multi-blob model as

$$\text{Model}(d) = \langle n_d, \bar{\mu}_{sd}, V_{sd}, \bar{\mu}_{cd}, V_{cd}, T_d \rangle, \quad d = 1, \dots, B, \quad \text{Eq. 3-18}$$

where  $n_d$  is the number of pixels in the  $d$ th bin,  $\bar{\mu}_{sd}$  and  $V_{sd}$  are the mean vector and covariance matrices in the spatial domain,  $\bar{\mu}_{cd}$  and  $V_{cd}$  are the mean vector and covariance matrices in the RGB color space. The number  $B$  is the number of color bins. According to this model, object pixels can be separated into  $B$  blobs. Nevertheless, the blobs should have different reliabilities. Thus, we add another parameter  $T_d$  to represent the reliability of each blob. Here,  $T_d$  is initialized to 1. We can update reliabilities according to the status or the situation during tracking.

To make the calculation more efficient, we introduce the recursive formula to gather the statistical measures  $\bar{\mu}_{sd}$ ,  $V_{sd}$ ,  $\bar{\mu}_{cd}$  and  $V_{cd}$ . By definition,

$$\mu_{x,n} = \frac{\sum_{i=1}^n x_i}{n} \Rightarrow \mu_{x,n+1} = \frac{\sum_{i=1}^{n+1} x_i}{n+1} = \frac{\sum_{i=1}^n x_i}{n+1} + \frac{x_{n+1}}{n+1} = \frac{n}{n+1} \mu_{x,n} + \frac{x_{n+1}}{n+1} \quad \text{Eq. 3-19}$$

$$\therefore \bar{\mu}_{sd,n+1} = \frac{n}{n+1} \bar{\mu}_{sd,n} + \frac{\bar{x}_{n+1}}{n+1}; \quad \bar{\mu}_{cd,n+1} = \frac{n}{n+1} \bar{\mu}_{cd,n} + \frac{\bar{c}_{n+1}}{n+1}, \quad \text{Eq. 3-20}$$

where  $\bar{x}_n = [x_n \ y_n]$ ,  $\bar{c}_n = [r_n \ g_n \ b_n]$ .

$$\begin{aligned} \sigma_{x,n}^2 &= \frac{\sum_{i=1}^n (x_i - \mu_{x,n})^2}{n} \\ \Rightarrow \sigma_{x,n+1}^2 &= \frac{\sum_{i=1}^{n+1} (x_i - \mu_{x,n+1})^2}{n+1} = \frac{\sum_{i=1}^n (x_i - \mu_{x,n+1})^2 + (x_{n+1} - \mu_{x,n+1})^2}{n+1}, \end{aligned}$$

where  $\sum_{i=1}^n (x_i - \mu_{x,n+1})^2 - \sum_{i=1}^n (x_i - \mu_{x,n})^2$

$$= \sum_{i=1}^n [(x_i - \mu_{x,n+1})^2 - (x_i - \mu_{x,n})^2]$$

$$= \sum_{i=1}^n [(\mu_{x,n} - \mu_{x,n+1})(2x_i - \mu_{x,n+1} - \mu_{x,n})]$$

$$= (\mu_{x,n} - \mu_{x,n+1}) \left[ \sum_{i=1}^n x_i - n\mu_{x,n+1} \right].$$

Eq. 3-21

Based on Eq. 3-20, Eq. 3-21 becomes  $\sum_{i=1}^n (x_i - \mu_{x,n+1})^2 - \sum_{i=1}^n (x_i - \mu_{x,n})^2 = \frac{1}{n} (\mu_{x,n+1} - \mu_{x,n})^2$

$$\text{Hence, } \therefore \sigma_{x,n+1}^2 = \frac{\sum_{i=1}^n (x_i - \mu_{x,n})^2 + \frac{1}{n} (x_{n+1} - \mu_{x,n+1})^2 + (x_{n+1} - \mu_{x,n+1})^2}{n+1}$$

$$= \frac{n}{n+1} \sigma_{x,n}^2 + \frac{1}{n} (x_{n+1} - \mu_{x,n+1})^2, \text{ where } \sigma_1^2 = 0.$$

Similarly,  $\sigma_{xy,n+1} = \frac{n}{n+1}\sigma_{xy,n+1} + \frac{1}{n}(x_{n+1} - \mu_{x,n+1})(y_{n+1} - \mu_{y,n+1})$ . Thus, we have

$$V_{sd,n+1} = \frac{n}{n+1}V_{sd,n} + \frac{1}{n}(\bar{x}_{n+1} - \bar{\mu}_{sd,n+1})^T (\bar{x}_{n+1} - \bar{\mu}_{sd,n+1}), \text{ where } V_{sd,1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{Eq. 3-22}$$

$$V_{cd,n+1} = \frac{n}{n+1}V_{cd,n} + \frac{1}{n}(\bar{c}_{n+1} - \bar{\mu}_{cd,n+1})^T (\bar{c}_{n+1} - \bar{\mu}_{cd,n+1}), \text{ where } V_{cd,1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad \text{Eq. 3-23}$$

We let the diagonal terms of initial value be 1 to ensure the covariance matrices invertible. Figure 3.12 shows the multi-blob model build from the detection result shown in Figure 3.10. In the RGB color space, we have 4 bins per channel to build the model. Separating each channel into too many bins may loss correlation between feature domain and spatial domain. We rank the blobs according to  $n_d$ , and mark the top 5 blobs with their mean values in the RGB color space.

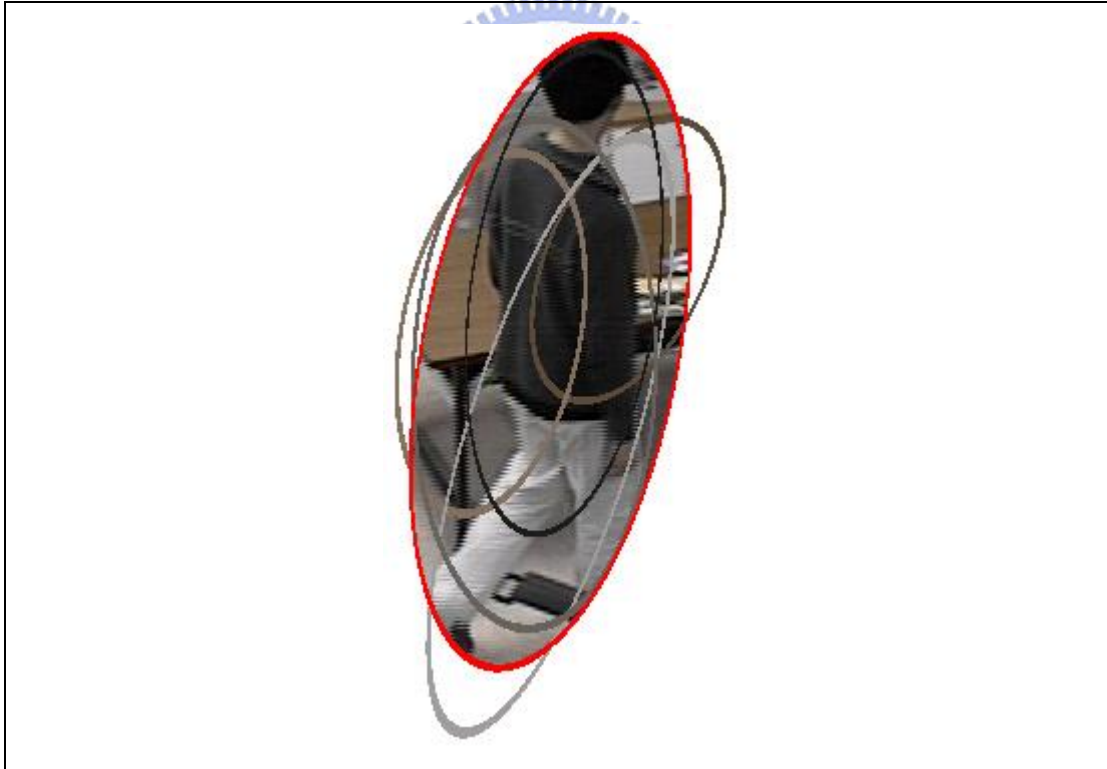


Figure 3.12 Five blobs that appear most frequently are marked with their mean values in the RGB color space.

Based on the multi-blob model, we can roughly know how the colors distribute in the bounding ellipse in both spatial domain and feature domain. The knowledge of the relative positions among these blobs can increase tracking accuracy.

### 3.2.3 Similarity Measure

We are not willing to define the similarity between two multi-blob models but to define the similarity between a region of the image frame and a multi-blob model. The mean-shift process deduced from the similarity between two multi-blob models has to build a model for every iteration. This will cause information loss and increase computational complexity. In Eq. 2-10, we accumulate the similarity value caused by each pixel. Comparing to the model, a closer pixel with a similar color causes a larger similarity. We follow this idea and define our similarity measure based on a multi-blob model.

Let  $\bar{y}$  be the center of the target ellipse and  $\bar{\mu}$  be the center of the model ellipse. Then, we define the similarity  $J$  as

$$J(\bar{y}) = \frac{1}{N} \sum_{i \in I} \sum_{d=1}^B w((\bar{y}_i - \bar{y}) - (\bar{\mu}_{sd} - \bar{\mu}), V_{sd}) w(\bar{c}_i - \bar{\mu}_{cd}, V_{cd}) T_d \delta(b(\bar{y}_i) - d), \quad \text{Eq. 3-24}$$

where  $w(\bar{x}, V) = \exp\left(-\frac{\bar{x}V^{-1}\bar{x}^T}{2}\right)$  is a multi-variant Gaussian kernel,  $I = \{\bar{y}_i, \bar{c}_i\}$  is the

samples enclosed by the target ellipse, and  $N$  is the number of samples. Each pixel is weighted by the Gaussian kernels in both spatial domain and feature space and then multiplied by the reliability coefficient. The definition can be regarded as the mean of the weighting. Again, as an example, we detect the moving object and build a multi-blob model using Frame 15 of the *Hans* sequence. We apply the same bounding ellipse to Frame 250 and calculate the similarity surface around the object. Then, we find the location  $\bar{y}$  with the maximal similarity value. Figure 3.13 shows that the similarity measure now is more discriminative than the Bhattacharyya coefficient, which has been shown in Figure 3.6.

The similarity measure can be generalized to be

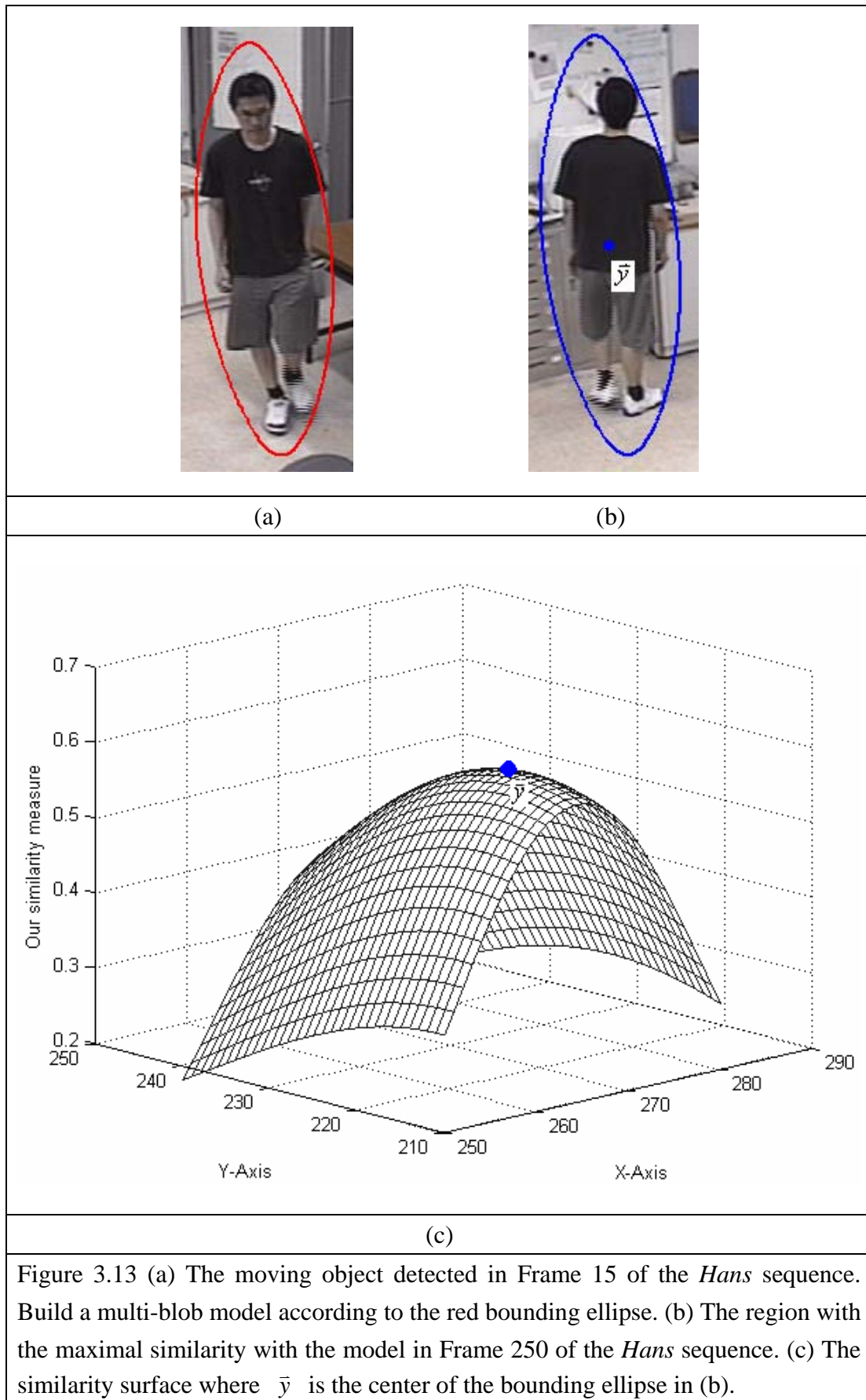
$$J(\bar{y}) = \frac{1}{N} \sum_{i \in I} \sum_{d=1}^B w((\bar{y}_i - \bar{y}) - (\bar{\mu}_{sd} - \bar{\mu}), V_{sd})^\alpha w(\bar{c}_i - \bar{\mu}_{cd}, V_{cd})^\beta T_d \delta(b(\bar{y}_i) - d), \quad \text{Eq. 3-25}$$

where  $\alpha$  and  $\beta$  represent the dominance of the Gaussian kernel. The spatial Gaussian kernel will dominate if  $\alpha \gg \beta$ . If both  $\alpha$  and  $\beta$  are much smaller than 1, the similarity will become

$$J(\bar{y}) \cong \frac{1}{N} \sum_{i \in I} \sum_{d=1}^B T_d \delta(b(\bar{y}_i) - d), \quad \text{Eq. 3-26}$$

which corresponds to the average reliability of the samples enclosed by the bounding ellipse. For simplicity, we choose  $\alpha = \beta = 1$ .





### 3.2.4 Mean-Shift Tracking Procedure

The distance function is defined as

$$L(\bar{y}) = -\log J(\bar{y}). \quad \text{Eq. 3-27}$$

The gradient of the distance function with respect to the vector  $\bar{y}$  is

$$\nabla L(\bar{y}) = -\frac{\nabla J(\bar{y})}{J(\bar{y})}, \quad \text{Eq. 3-28}$$

where

$$\begin{aligned} \nabla J(\bar{y}) &= \frac{1}{N} \sum_{i \in I} \sum_{d=1}^B \nabla w((\bar{y}_i - \bar{y}) - (\bar{\mu}_{sd} - \bar{\mu}), V_{sd}) w(\bar{c}_i - \bar{\mu}_{cd}, V_{cd}) T_d \delta(b(\bar{y}_i) - d) \\ &= \frac{1}{N} \sum_{i \in I} \sum_{d=1}^B [(\bar{\mu}_{sd} - \bar{\mu}) - (\bar{y}_i - \bar{y})] g(\Delta \bar{y} - \Delta \bar{\mu}, V_{sd}) w(\Delta \bar{c}, V_{cd}) T_d \delta(b(\bar{y}_i) - d), \end{aligned}$$

and  $g(\cdot) = -w(\cdot)$  is the shadow kernel, which is also a multi-variant Gaussian profile in our case, with  $\Delta \bar{y} = (\bar{y}_i - \bar{y})$ ,  $\Delta \bar{\mu} = (\bar{\mu}_{sd} - \bar{\mu})$ ,  $\Delta \bar{c} = (\bar{c}_i - \bar{\mu}_{cd})$ . Thus, Eq. 3-28 becomes

$$\begin{aligned} \nabla L(\bar{y}) &= \frac{\sum_{i \in I} \sum_{d=1}^B [(\bar{y}_i - \bar{y}) - (\bar{\mu}_{sd} - \bar{\mu})] g(\Delta \bar{y} - \Delta \bar{\mu}, V_{sd}) w(\Delta \bar{c}, V_{cd}) T_d \delta(b(\bar{y}_i) - d)}{\sum_{i \in I} \sum_{d=1}^B g(\Delta \bar{y} - \Delta \bar{\mu}, V_{sd}) w(\Delta \bar{c}, V_{cd}) T_d \delta(b(\bar{y}_i) - d)} \\ &= \frac{\sum_{i \in I} \sum_{d=1}^B [\bar{y}_i - \bar{\mu}_{sd}] g(\Delta \bar{y} - \Delta \bar{\mu}, V_{sd}) w(\Delta \bar{c}, V_{cd}) T_d \delta(b(\bar{y}_i) - d)}{\sum_{i \in I} \sum_{d=1}^B g(\Delta \bar{y} - \Delta \bar{\mu}, V_{sd}) w(\Delta \bar{c}, V_{cd}) T_d \delta(b(\bar{y}_i) - d)} - \bar{y} + \bar{\mu}. \end{aligned} \quad \text{Eq. 3-29}$$

By letting Eq. 3-29 equal to 0, we have

$$\bar{y}_1 = \frac{\sum_{i \in I} \sum_{d=1}^B [\bar{y}_i - \bar{\mu}_{sd}] g(\Delta \bar{y}_0 - \Delta \bar{\mu}, V_{sd}) w(\Delta \bar{c}, V_{cd}) T_d \delta(b(\bar{y}_i) - d)}{\sum_{i \in I} \sum_{d=1}^B g(\Delta \bar{y}_0 - \Delta \bar{\mu}, V_{sd}) w(\Delta \bar{c}, V_{cd}) T_d \delta(b(\bar{y}_i) - d)} + \bar{\mu}, \quad \text{Eq. 3-30}$$

where  $\Delta \bar{y}_0 = \bar{y}_i - \bar{y}_0$  and an iterative mean-shift formula is obtained.

Or by letting  $\nabla J(\bar{y}) = 0$ , we have

$$\begin{aligned}
& \sum_{i \in I} \sum_{d=1}^B [(\bar{\mu}_{sd} - \bar{\mu}) - (\bar{y}_i - \bar{y})] g(\Delta \bar{y} - \Delta \bar{\mu}, V_{sd}) w(\Delta \bar{c}, V_{cd}) T_d \delta(b(\bar{y}_i) - d) = 0 \\
\Rightarrow & (\bar{y} - \bar{\mu}) \sum_{i \in I} \sum_{d=1}^B g(\Delta \bar{y} - \Delta \bar{\mu}, V_{sd}) w(\Delta \bar{c}, V_{cd}) T_d \delta(b(\bar{y}_i) - d) \\
& = \sum_{i \in I} \sum_{d=1}^B (\bar{y}_i - \bar{\mu}_{sd}) g(\Delta \bar{y} - \Delta \bar{\mu}, V_{sd}) w(\Delta \bar{c}, V_{cd}) T_d \delta(b(\bar{y}_i) - d) \\
\Rightarrow & \bar{y}_1 = \frac{\sum_{i \in I} \sum_{d=1}^B [\bar{y}_i - \bar{\mu}_{sd}] g(\Delta \bar{y}_0 - \Delta \bar{\mu}, V_{sd}) w(\Delta \bar{c}, V_{cd}) T_d \delta(b(\bar{y}_i) - d)}{\sum_{i \in I} \sum_{d=1}^B g(\Delta \bar{y}_0 - \Delta \bar{\mu}, V_{sd}) w(\Delta \bar{c}, V_{cd}) T_d \delta(b(\bar{y}_i) - d)} + \bar{\mu},
\end{aligned}$$

which is the same as Eq. 3-30.

For the generalized similarity measure as expressed in Eq. 3-25, we have

$$\begin{aligned}
\nabla J(\bar{y}) = & \frac{1}{N} \sum_{i \in I} \sum_{d=1}^B [(\bar{\mu}_{sd} - \bar{\mu}) - (\bar{y}_i - \bar{y})] \alpha w((\bar{y}_i - \bar{y}) - (\bar{\mu}_{sd} - \bar{\mu}), V_{sd})^{\alpha-1} \\
& \times g(\Delta \bar{y} - \Delta \bar{\mu}, V_{sd}) w(\bar{c}_i - \bar{\mu}_{cd}, V_{cd})^\beta T_d \delta(b(\bar{y}_i) - d),
\end{aligned}$$

where  $g(\cdot) = -w(\cdot)$  is also a multi-variant Gaussian profile in our case. Therefore,

$$\begin{aligned}
& \nabla J(\bar{y}) = 0 \\
\Rightarrow & \sum_{i \in I} \sum_{d=1}^B [(\bar{\mu}_{sd} - \bar{\mu}) - (\bar{y}_i - \bar{y})] g(\Delta \bar{y} - \Delta \bar{\mu}, V_{sd})^\alpha w(\bar{c}_i - \bar{\mu}_{cd}, V_{cd})^\beta T_d \delta(b(\bar{y}_i) - d) = 0 \\
\Rightarrow & \bar{y}_1 = \frac{\sum_{i \in I} \sum_{d=1}^B [\bar{y}_i - \bar{\mu}_{sd}] g(\Delta \bar{y}_0 - \Delta \bar{\mu}, V_{sd})^\alpha w(\Delta \bar{c}, V_{cd})^\beta T_d \delta(b(\bar{y}_i) - d)}{\sum_{i \in I} \sum_{d=1}^B g(\Delta \bar{y}_0 - \Delta \bar{\mu}, V_{sd})^\alpha w(\Delta \bar{c}, V_{cd})^\beta T_d \delta(b(\bar{y}_i) - d)} + \bar{\mu}, \quad \text{Eq. 3-31}
\end{aligned}$$

which is the generalized mean-shift equation.

### 3.2.5 Updating Size and Orientation

To update the size and orientation of the bounding ellipse, we refer to the region around the bounding ellipse. The bounding ellipse is defined as the 2-sigma contour. Here, we consider the band between the 3-sigma contour and the 2-sigma contour as the background field. Figure 3.14 indicates the foreground region and the background region based on a bounding ellipse.

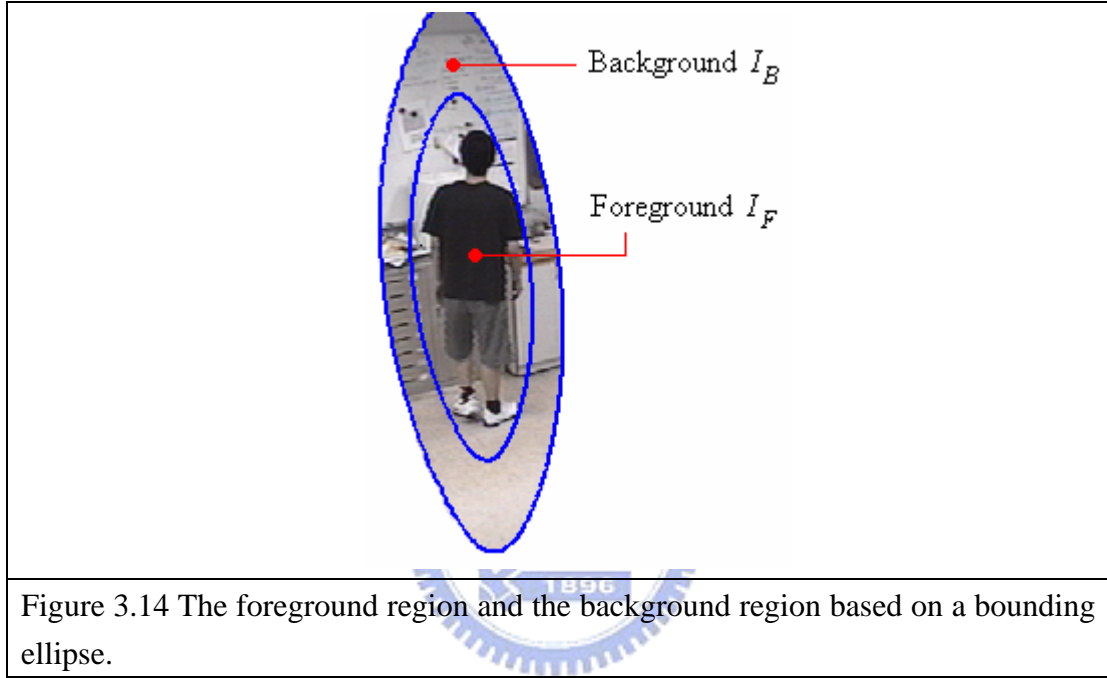


Figure 3.14 The foreground region and the background region based on a bounding ellipse.

To reach the modification of bounding ellipse, we modify each color blob first. We re-compute  $\mu_{sd}$  and  $V_{sd}$  of each blob by weighting the samples belonging to  $I_B$  and  $I_F$ . Again, for computational efficiency, the formula must be recursive.

Let  $q_i = w\left(\left(\bar{y}_i - \bar{y}\right) - \left(\bar{\mu}_{sd} - \bar{\mu}\right), V_{sd}\right)^\alpha w\left(\bar{c}_i - \bar{\mu}_{cd}, V_{cd}\right)^\beta$  be the weighting of the  $i$ th sample  $\bar{y}_i = (x_i, y_i)$ , where  $d = b(\bar{y}_i)$ .

$$\begin{aligned} \mu_{x,n} &= \frac{\sum_{i=1}^n q_i x_i}{\sum_{i=1}^n q_i} \Rightarrow \mu_{x,n+1} = \frac{\sum_{i=1}^{n+1} q_i x_i}{\sum_{i=1}^{n+1} q_i} = \frac{\sum_{i=1}^n q_i x_i}{\sum_{i=1}^n q_i} + \frac{q_{n+1}}{\sum_{i=1}^{n+1} q_i} x_{n+1} \\ &= \frac{Q_n}{Q_n + q_{n+1}} \mu_{x,n} + \frac{q_{n+1}}{Q_n + q_{n+1}} x_{n+1}, \end{aligned}$$

where  $Q_n = \sum_{i=1}^n q_i$ ,  $\mu_0 = 0$ .

Similarly,

$$\mu_{y,n+1} = \frac{Q_n}{Q_n + q_{n+1}} \mu_{y,n} + \frac{q_{n+1}}{Q_n + q_{n+1}} y_{n+1},$$

$$\therefore \bar{\mu}_{n+1} = \frac{Q_n}{Q_n + q_{n+1}} \bar{\mu}_n + \frac{q_{n+1}}{Q_n + q_{n+1}} \bar{y}_{n+1} \quad \text{Eq. 3-32}$$

$$\sigma_{x,n}^2 = \frac{\sum_{i=1}^n q_i (x_i - \mu_{x,n})^2}{\sum_{i=1}^n q_i} \Rightarrow \sigma_{x,n+1}^2 = \frac{\sum_{i=1}^{n+1} q_i (x_i - \mu_{x,n+1})^2}{\sum_{i=1}^{n+1} q_i} = \frac{\sum_{i=1}^n q_i (x_i - \mu_{x,n+1})^2 + q_{n+1} (x_{n+1} - \mu_{x,n+1})^2}{Q_n + q_{n+1}}$$

$$\text{where } \sum_{i=1}^n q_i (x_i - \mu_{x,n+1})^2 - \sum_{i=1}^n q_i (x_i - \mu_{x,n})^2$$

$$= \sum_{i=1}^n q_i [(\mu_{x,n} - \mu_{x,n+1})(2x_i - \mu_{x,n+1} - \mu_{x,n})]$$

$$= (\mu_{x,n} - \mu_{x,n+1}) \left[ \sum_{i=1}^n q_i x_i - \mu_{x,n+1} Q_n \right]$$

$$= \frac{q_{n+1}^2}{Q_n} (\mu_{n+1} - x_{n+1})^2$$

$$\therefore \sigma_{x,n+1}^2 = \frac{\sum_{i=1}^n q_i (x_i - \mu_{x,n})^2 + \frac{q_{n+1}^2}{Q_n} (x_{n+1} - \mu_{x,n+1})^2 + q_{n+1} (x_{n+1} - \mu_{x,n+1})^2}{Q_n + q_{n+1}}$$

$$= \frac{Q_n}{Q_n + q_{n+1}} \sigma_{x,n}^2 + \frac{q_{n+1}}{Q_n} (x_{n+1} - \mu_{x,n+1})^2,$$

where  $\sigma_{x,0}^2 = \sigma_{x,1}^2 = 0$ .

Similarly,

$$\sigma_{y,n+1}^2 = \frac{Q_n}{Q_n + q_{n+1}} \sigma_{y,n}^2 + \frac{q_{n+1}}{Q_n} (y_{n+1} - \mu_{y,n+1})^2,$$

$$\sigma_{xy,n+1} = \frac{Q_n}{Q_n + q_{n+1}} \sigma_{xy,n} + \frac{q_{n+1}}{Q_n} (x_{n+1} - \mu_{x,n+1})(y_{n+1} - \mu_{y,n+1}),$$

$$\therefore V_{n+1} = \frac{Q_n}{Q_n + q_{n+1}} V_n + \frac{q_{n+1}}{Q_n} (\bar{y}_{n+1} - \bar{\mu}_{n+1})^T (\bar{y}_{n+1} - \bar{\mu}_{n+1}). \quad \text{Eq. 3-33}$$

Based on Eq. 3-32 and Eq. 3-33, each blob can be updated. To obtain the new bounding ellipse, we have to merge together all the blobs. By definition,

$$\bar{\mu}_{sd} = \frac{\sum_{i=1}^{n_d} \bar{y}_i}{n_d} \Rightarrow \bar{\mu} = \frac{\sum_{j=1}^n \bar{y}_j}{n} = \frac{\sum_{d=1}^B \sum_{i=1}^{n_d} \bar{y}_i}{\sum_{d=1}^B n_d} = \frac{\sum_{d=1}^B n_d \bar{\mu}_{sd}}{\sum_{d=1}^B n_d}. \quad \text{Eq. 3-34}$$

$$\sigma_{xd}^2 = \frac{\sum_{i=1}^{n_d} x_i^2}{n_d} - \mu_{xd}^2 \Rightarrow \sum_{i=1}^{n_d} x_i^2 = n_d (\sigma_{xd}^2 + \mu_{xd}^2),$$

$$\sigma_x^2 = \frac{\sum_{j=1}^n x_j^2}{n} - \mu_x^2 = \frac{\sum_{d=1}^B \sum_{i=1}^{n_d} x_i^2}{\sum_{d=1}^B n_d} - \mu_x^2 = \frac{\sum_{d=1}^B n_d (\sigma_{xd}^2 + \mu_{xd}^2)}{\sum_{d=1}^B n_d} - \mu_x^2.$$

Similarly,

$$\sigma_y^2 = \frac{\sum_{d=1}^B n_d (\sigma_{yd}^2 + \mu_{yd}^2)}{\sum_{d=1}^B n_d} - \mu_y^2,$$

$$\sigma_{xy} = \frac{\sum_{d=1}^B n_d (\sigma_{xyd} + \mu_{xd} \mu_{yd})}{\sum_{d=1}^B n_d} - \mu_x \mu_y,$$

$$\therefore V = \frac{\sum_{d=1}^B n_d (V_{sd} + \bar{\mu}_{sd}^T \bar{\mu}_{sd})}{\sum_{d=1}^B n_d} - \bar{\mu}^T \bar{\mu} \quad \text{Eq. 3-35}$$

From Eq. 3-34 and Eq. 3-35, the overall mean vector and covariance matrix can be regarded as the weighted combination of blobs with weighting  $n_d$ . To consider the reliability, we modify Eq. 3-34 and Eq. 3-35 to be

$$\bar{\mu} = \frac{\sum_{d=1}^B n_d T_d^\gamma \bar{\mu}_{sd}}{\sum_{d=1}^B n_d T_d^\gamma}; \quad V = \frac{\sum_{d=1}^B n_d T_d^\gamma (V_{sd} + \bar{\mu}_{sd}^T \bar{\mu}_{sd})}{\sum_{d=1}^B n_d T_d^\gamma} - \bar{\mu}^T \bar{\mu}, \quad \text{Eq. 3-36}$$

where  $\gamma$  represents the dominance of reliability. A large  $\gamma$  will cause the mean vector and covariance matrix to be dominated by certain blobs with high reliability.

When using Gaussian profile to estimate a distribution  $f(x, y)$ , the variance should be obtained from  $\sigma_x^2 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \mu)^2 f(x, y) dx dy$ , where  $\mu = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x f(x, y) dx dy$ . In our case, the samples are weighted by  $q_i = w((\bar{y}_i - \bar{y}) - (\bar{\mu}_{sd} - \bar{\mu}), V_{sd})^\alpha w(\bar{c}_i - \bar{\mu}_{cd}, V_{cd})^\beta$ . Hence, the variance should be

$$\sigma_x^2 = \frac{1}{2\pi |V|^{1/2}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \mu_x)^2 w([x \ y] - [\mu_x \ \mu_y], V)^\alpha w(\bar{c}_i - \bar{\mu}_{cd}, V_{cd})^\beta dx dy, \quad \text{Eq. 3-37}$$

where  $w(\bar{c}_i - \bar{\mu}_{cd}, V_{cd})$  is a Gaussian random variable independent of  $x$  and  $y$ . Due to the independence,  $w(\bar{c}_i - \bar{\mu}_{cd}, V_{cd})$  does not affect the variance. However, the above procedure only considers the region enclosed by the t-sigma contour. Hence, the covariance matrix obtained by Eq. 3-36 is smaller than the true one.

Let the true covariance matrix be  $\tilde{V} = \begin{bmatrix} \tilde{\sigma}_x^2 & \tilde{\sigma}_{xy} \\ \tilde{\sigma}_{xy} & \tilde{\sigma}_y^2 \end{bmatrix} = f(\alpha, t) \times V$ . Since  $f(\alpha, t)$  is

independent of orientation, we assume  $\sigma_{xy} = 0$  and  $\bar{\mu} = [0 \ 0]$  for simplicity.

$$\sigma_x^2 = \frac{1}{2\pi\tilde{\sigma}_x\tilde{\sigma}_y} \oint\!\!\!\oint_I x^2 e^{\frac{-\alpha x^2}{2\tilde{\sigma}_x^2}} e^{\frac{-\alpha y^2}{2\tilde{\sigma}_y^2}} dx dy, \quad \text{Eq. 3-38}$$

where  $I$  is the region enclosed by the t-sigma contour. By letting  $X = \frac{\sqrt{\alpha}x}{\tilde{\sigma}_x}$ ,  $Y = \frac{\sqrt{\alpha}y}{\tilde{\sigma}_y}$ , Eq.

3-38 becomes

$$\sigma_x^2 = \frac{\tilde{\sigma}_x^2}{2\pi\alpha^2} \oint\!\!\!\oint_I X^2 e^{\frac{-X^2}{2}} e^{\frac{-Y^2}{2}} dXdY.$$

Let  $X = R \cos \theta$  and  $Y = R \sin \theta$ , we have

$$\begin{aligned} \sigma_x^2 &= \frac{\tilde{\sigma}_x^2}{2\pi\alpha^2} \int_0^{2\pi} \int_0^{\sqrt{\alpha}t} R^3 \cos^2 \theta e^{\frac{-R^2}{2}} dR d\theta, \\ &= \frac{\tilde{\sigma}_x^2}{2\pi\alpha^2} \int_0^{2\pi} \cos^2 \theta d\theta \int_0^{\sqrt{\alpha}t} R^3 e^{\frac{-R^2}{2}} dR. \end{aligned}$$

By partial integral, we let  $u = R^2$ ,  $dv = R e^{\frac{-R^2}{2}} dR \Rightarrow dv = 2R dR$ ,  $v = -e^{\frac{-R^2}{2}}$ .

$$\begin{aligned} \sigma_x^2 &= \frac{\tilde{\sigma}_x^2}{2\pi\alpha^2} \int_0^{2\pi} \frac{\cos 2\theta + 1}{2} d\theta \left[ -R^2 e^{\frac{-R^2}{2}} \Big|_0^{\sqrt{\alpha}t} + 2 \int_0^{\sqrt{\alpha}t} R e^{\frac{-R^2}{2}} dR \right]. \\ &= \frac{\tilde{\sigma}_x^2}{2\alpha^2} \left[ -\alpha t^2 e^{\frac{-\alpha t^2}{2}} - 2e^{\frac{-R^2}{2}} \Big|_0^{\sqrt{\alpha}t} \right] \\ &= \frac{\tilde{\sigma}_x^2}{\alpha^2} \left[ 1 - e^{\frac{-\alpha t^2}{2}} - \frac{\alpha t^2}{2} e^{\frac{-\alpha t^2}{2}} \right]. \end{aligned} \quad \text{Eq. 3-39}$$

Therefore, the new covariance matrix obtained by Eq. 3-36 should be multiplied by the

factor  $f(\alpha, t) = \alpha^2 \left( 1 - e^{\frac{-\alpha t^2}{2}} - \frac{\alpha t^2}{2} e^{\frac{-\alpha t^2}{2}} \right)^{-1}$ . That is,



$$\bar{\mu} = \frac{\sum_{d=1}^B n_d T_d^\gamma \bar{\mu}_{sd}}{\sum_{d=1}^B n_d T_d^\gamma}; \quad V = \left[ \frac{\sum_{d=1}^B n_d T_d^\gamma (V_{sd} + \bar{\mu}_{sd}^T \bar{\mu}_{sd})}{\sum_{d=1}^B n_d T_d^\gamma} - \bar{\mu}^T \bar{\mu} \right] f(\alpha, t). \quad \text{Eq. 3-40}$$

For  $t = 2$ ,  $\alpha = 1$ ,  $f(\alpha, t) = 1.6835$ .

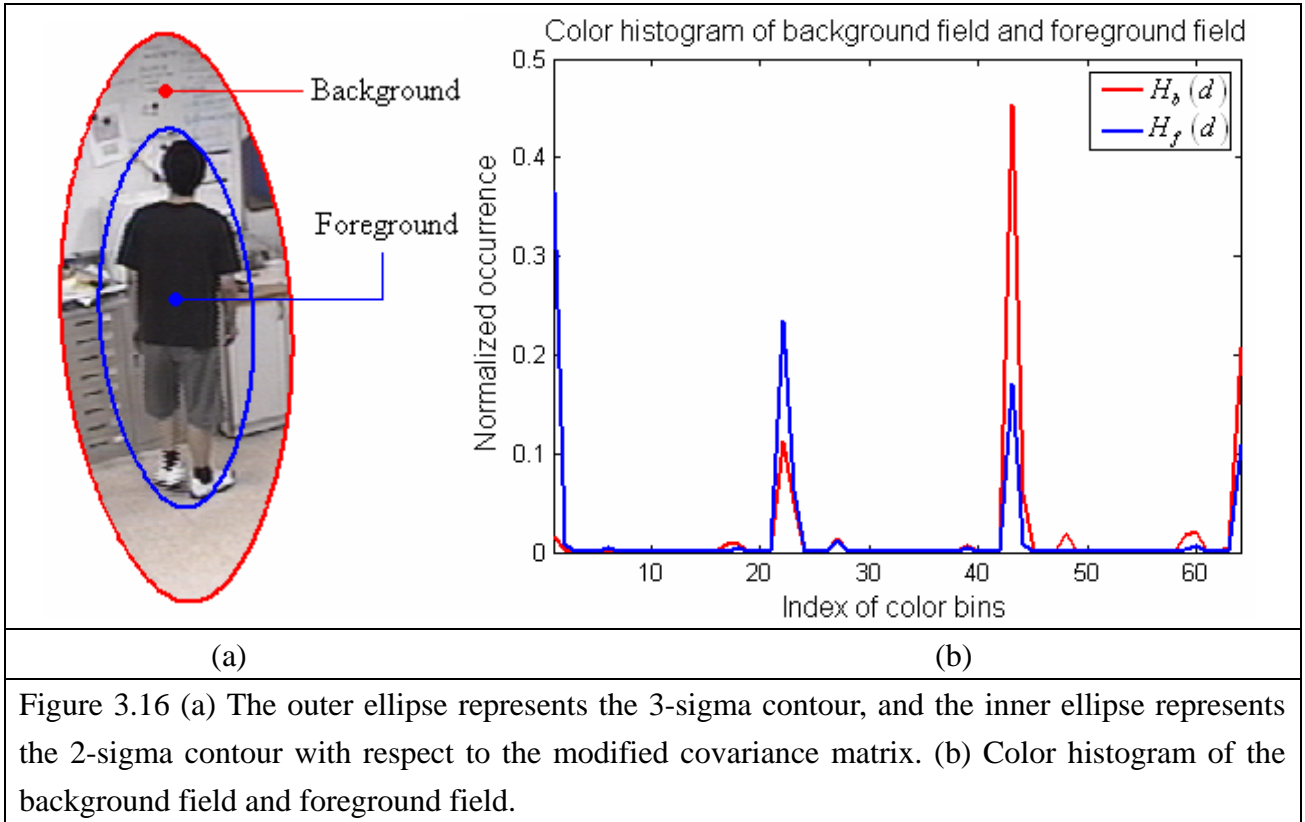
In Figure 3.14, the bounding ellipse is moved to the position with the maximal similarity. The motion of bounding ellipse is pure translation. Applying this updating process after the tracking process will make the bounding ellipse better fit the moving object. We set the  $\gamma$  in Eq. 3-40 to be 0.3. Figure 3.15 shows the result of the size and orientation updating.



Figure 3.15 Result of size and orientation updating. The blue ellipses represent the 2-sigma contour and the 3-sigma contour with respect to the original covariance matrix. The red ellipse represents the 2-sigma contour with respect to the modified covariance matrix.

### 3.2.6 Updating Reliability

To avoid distraction of tracking, colors appearing in both foreground field and background field should have lower reliability. Furthermore, if a color blob represents background pixels, the position relative to the center of the bounding ellipse will be varying during the tracking process. Based on these ideas, we proposed two strategies to update the reliability coefficient. Similar to Figure 3.14, we define the background field and foreground field as Figure 3.16(a). We let  $H_f(d)$  be a color histogram for the pixels in the foreground field and let  $H_b(d)$  be a color histogram for the pixels in the background field.



Define  $T_s(d) = \frac{H_f(d)}{\max\{H_b(d), \delta\}}$ , where  $\delta$  is a small value, say 0.001, to prevent the dividing

by zero. Blobs with a small  $T_s(d)$  value may get distracted by the background. At the same time, we use the samples of foreground field to build a multi-blob model as the target candidate. The target candidate is denoted as

$$\text{Candidate}(d) = \langle n'_d, \bar{\mu}'_{sd}, V'_{sd}, \bar{\mu}'_{cd}, V'_{cd}, T'_d \rangle, \quad d = 1, \dots, B, \quad \text{Eq. 3-41}$$

To measure the movement of a blob relative to the center, we define

$$\psi_d = (\Delta \bar{\mu}'_d - \Delta \bar{\mu}_d) \hat{V}_{sd}^{-1} (\Delta \bar{\mu}'_d - \Delta \bar{\mu}_d)^T \quad \text{Eq. 3-42}$$

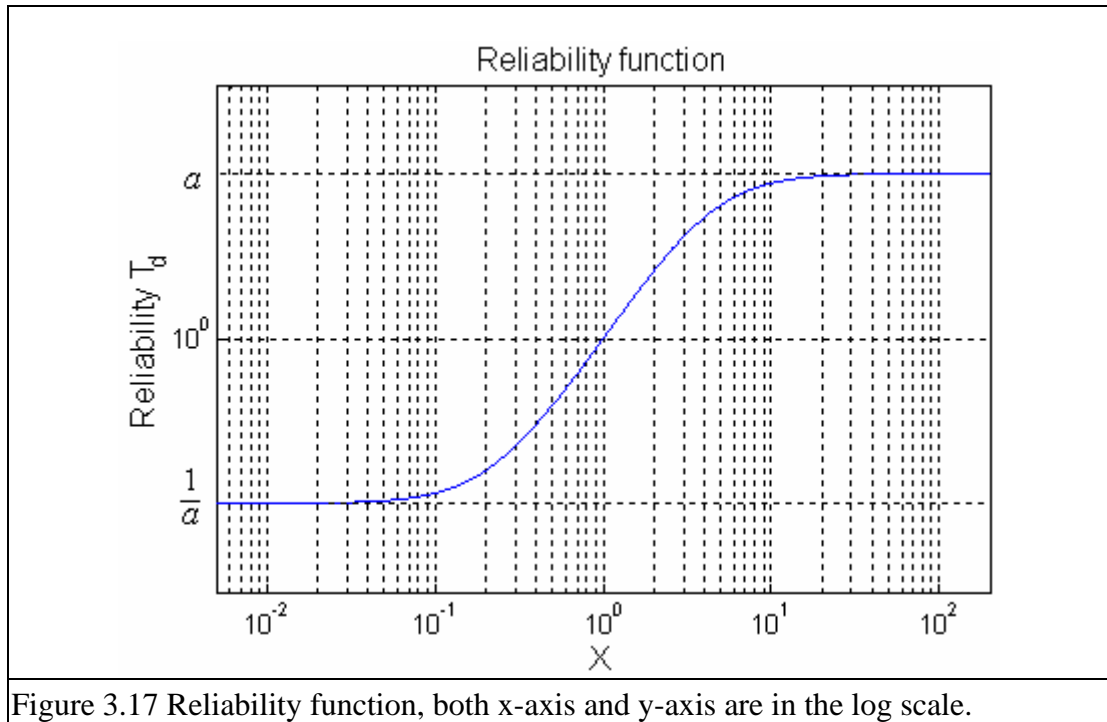
as the translation, where  $\Delta \bar{\mu}'_d = \bar{\mu}'_{sd} - \bar{y}_1$ ,  $\Delta \bar{\mu}_d = \bar{\mu}_{sd} - \bar{\mu}$  and  $\hat{V}_{sd}^{-1} = \frac{1}{2} (V'_{sd}{}^{-1} + V_{sd}{}^{-1})$ .

Notice that  $\psi_d$  is the average of two Mahalanobis distance, one between  $\Delta \bar{\mu}'_d$  and  $\Delta \bar{\mu}_d$ , and the other between  $\Delta \bar{\mu}'_d$  and  $\Delta \bar{\mu}_d$ . The reliability of the  $d$ -th blob is thought as increasable if  $\psi_d$  lies in the 1-sigma contour.

Reliability should be bounded to prevent the tracking behavior being dominated by certain blobs. On the other hand, reliability should spread widely to have discrimination between blobs. Thus, we define the reliability function as

$$T(X) = \frac{1 + aX^2}{a + X^2}, \quad \text{Eq. 3-43}$$

where  $a$  represents the discrimination between blobs. Low reliability blobs can be ignored if  $a$  is large enough.



From  $T_s(d)$ ,  $\psi_d$  and reliability function, we develop our reliability updating strategy.

There are five steps in updating reliability.

$$\text{Step 1. } T_d = \frac{1+aX^2}{a+X^2} \Rightarrow X_0 = \sqrt{\frac{aT_d-1}{a-T_d}}.$$

$$\text{Step 2. } X_1 = X_0 e^{\frac{(1-\psi_d^2)}{2}}.$$

$$\text{Step 3. Case 1: } T_s(d) \geq 3, \quad X_2 = X_1 \times 1.5$$

$$\text{Case 2: } 2 \geq T_s(d) > 1, \quad X_2 = \frac{X_1}{1.5}$$

$$\text{Case 3: } 1 \geq T_s(d), \quad X_2 = \frac{X_1}{3}.$$

$$\text{Step 4. } X = \max\left(\min(10a, X_2), \frac{1}{10a}\right).$$

$$\text{Step 5. } T_{d\_new} = \frac{1+aX^2}{a+X^2}.$$

In Step 4,  $X$  is bounded to prevent  $T_d = a$ , which will cause “dividing by zero” in Step 1. Another reason is to let the reliability be more sensitive. We choose  $a = 20$  in practice. In Figure 3.18, pixels with high reliability are marked in white; oppositely, pixels with low reliability are marked in black. Pixels remaining the original color represent medium reliability. Based on the reliability map, moving object can be roughly identified.



Figure 3.18 The reliability map, which can roughly identify the moving object.

### 3.2.7 Updating Target Model

Since a target candidate is already obtained when we update reliability, all we have to do is to decide when to update the model. The reliability map of two frames in the *Watson* sequence is shown in Figure 3.19. Considering the situation in Figure 3.19(a), obviously, this moment is not appropriate to update the target model. The unexpected background information may cause tracking failure. In Figure 3.19(b), the bounding ellipse tracks the moving object well, and the background is with low reliability. This is a suitable moment to update the target model.



(a)



(b)

Figure 3.19 Frames with pixels marked depend on reliability. (a) Frame 50, Sequence *Watson*. (b) Frame 65, Sequence *Watson*.

To evaluate the adequacy of updating target model, we introduce the two-class variance ratio. Let  $T_{d\_f}$  be the set of reliability for pixels in the foreground field and  $T_{d\_b}$  be the set of reliability for pixels in the background field.

$$\text{Variance Ratio} = \frac{\text{var}(T_{d\_f} + T_{d\_b})}{\text{var}(T_{d\_f}) + \text{var}(T_{d\_b})}, \quad \text{Eq. 3-44}$$

The variance ratio of reliability map shown in Figure 3.19(a) is 0.5866 and the variance ratio of reliability map shown in Figure 3.19(b) is 0.9280. We set a threshold at 0.65. Thus, the target model will be updated whenever the variance ratio is larger than 0.65.



### 3.2.8 Overall Object Tracking Process

To sum up, the flow chart of the proposed object tracking procedure is shown in Figure 3.20. Compared to the flow chart of the traditional mean-shift tracking process in Figure 3.3, we need not to build the candidate during the mean-shift iterations. The location converge

condition is  $\|\bar{y}_1 - \bar{y}_0\|^2 < 1$  and the orientation converge condition is  $\left| \frac{\det(V_{new})}{\det(V)} - 1 \right| < 0.05$ ,

which means the amount of samples varying is less than 5%.

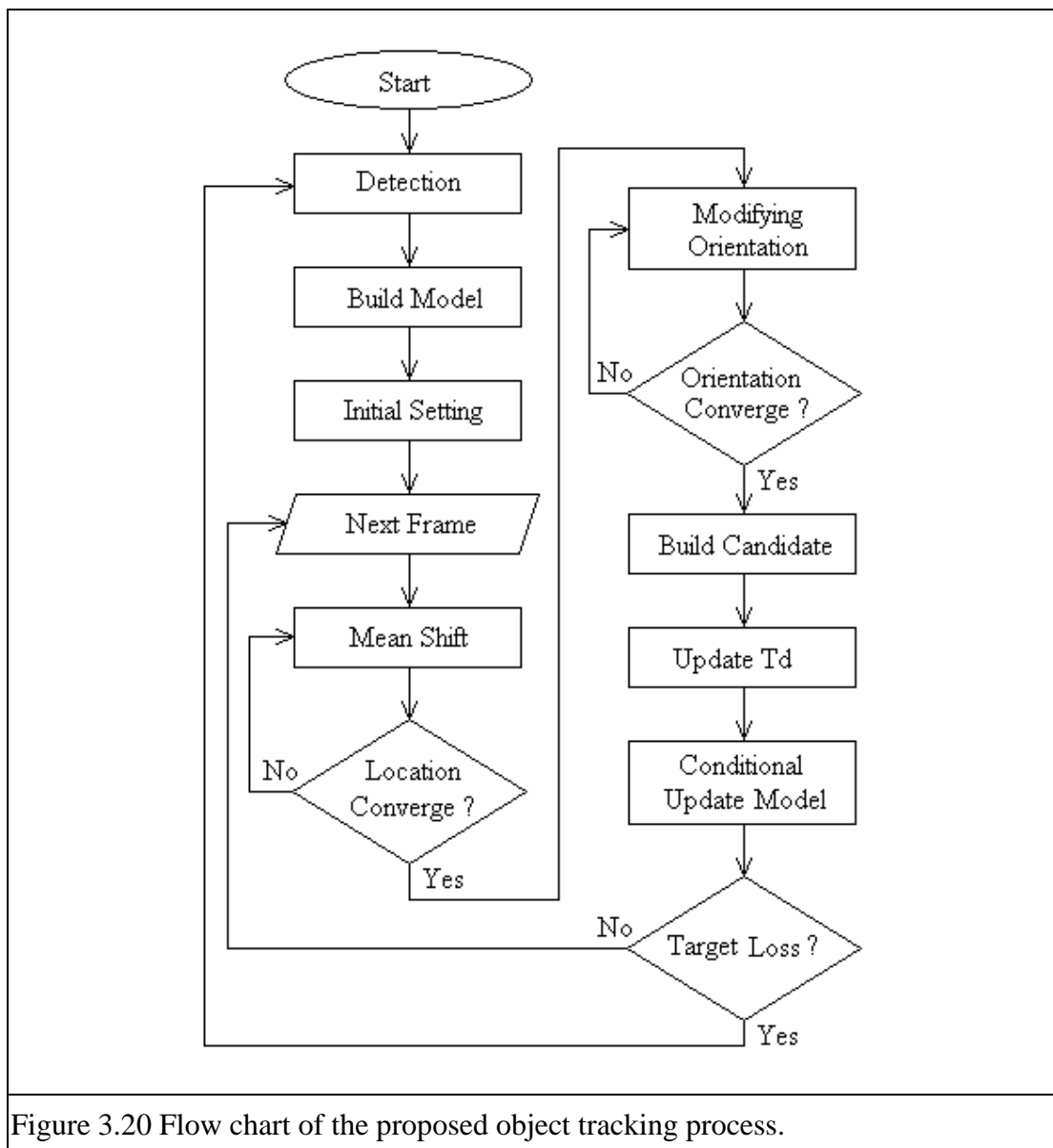


Figure 3.20 Flow chart of the proposed object tracking process.

Furthermore, we can do some simple predictions between frames or between mean-shift iterations to shorten the processing time. On the other hand, the judgment of

target loss is an additional stage to increase robustness. We have two simple rules to judge whether the tracking failure or not. The first rule is to check the covariance matrix  $V$  obtained by Eq. 3-40. If  $\det(V) \leq 0$ , we are not able to define the bounding ellipse. Recall Eq. 3-17, a negative area is not reasonable. In practice, instead of the bounding ellipse, a hyperbola is obtained. The second rule is to check the reliability. If all blobs are with low reliability, the tracking result is likely to be a failed one.



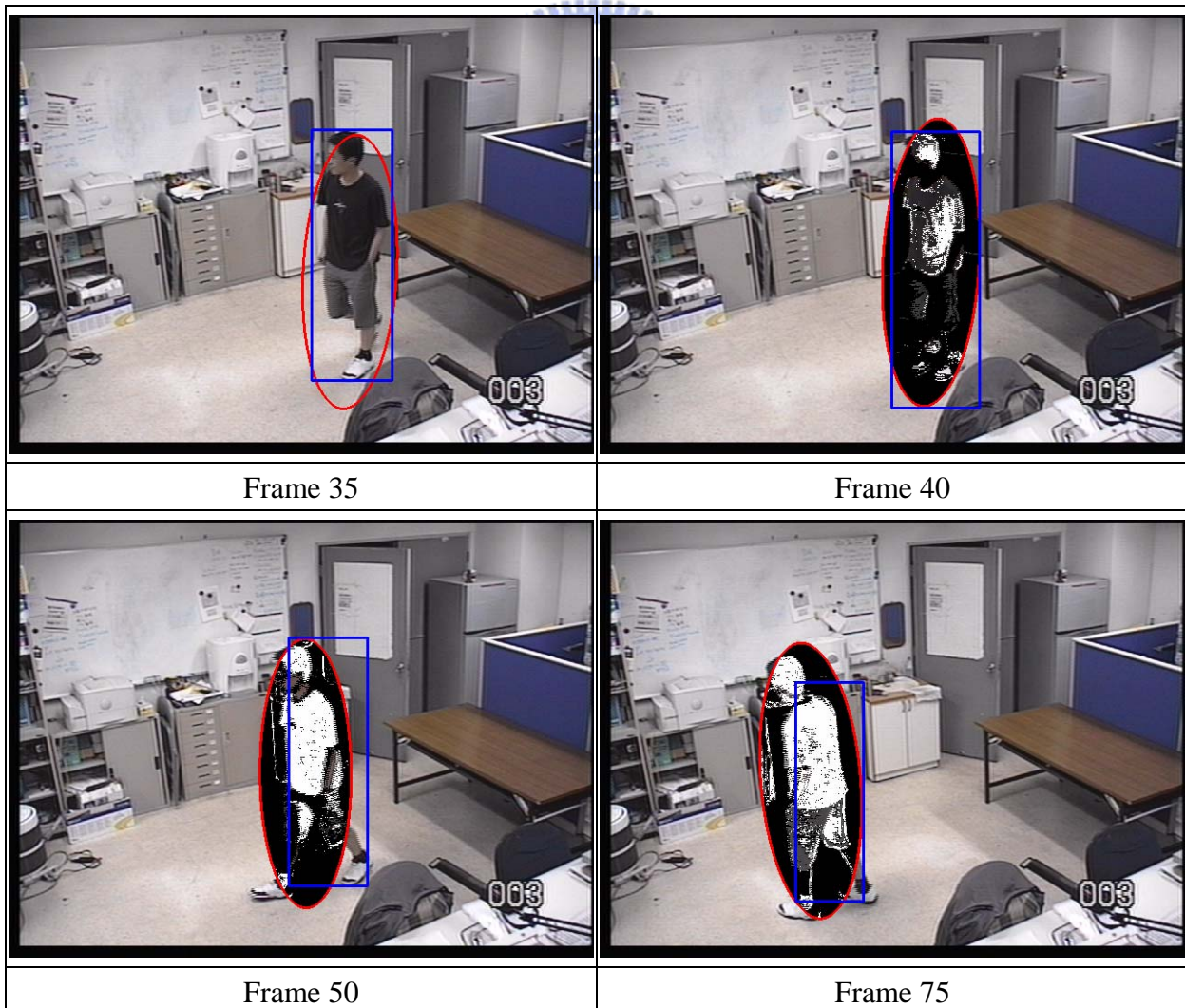


# Chapter 4.

## Experimental Results

We present some object tracking results using the proposed algorithm in this chapter. The Resolution of all sequences is  $640 \times 480$ . The algorithm is implemented with the MATLAB 6.5 platform and runs on a 3GHz Pentium4 PC with 512MB DRAM. The red ellipse and blue bounding box represent the result of proposed algorithm and the traditional mean-shift process, respectively.

In the first experiment, our mean-shift algorithm was run on the sequence “*Hans*”. There is neither scene change nor occlusion in this sequence. The tracking result is shown below. The reliability map is also shown to show how the reliabilities change during tracking. The multi-blob model is built at Frame 35. All the reliabilities are initialized to 1. In addition, we ran the traditional mean-shift procedure at the same time, which employs the “plus or minus 10 percent” scale adaptation method and uses a  $16 \times 16 \times 16$  histogram in the RGB space as the model.

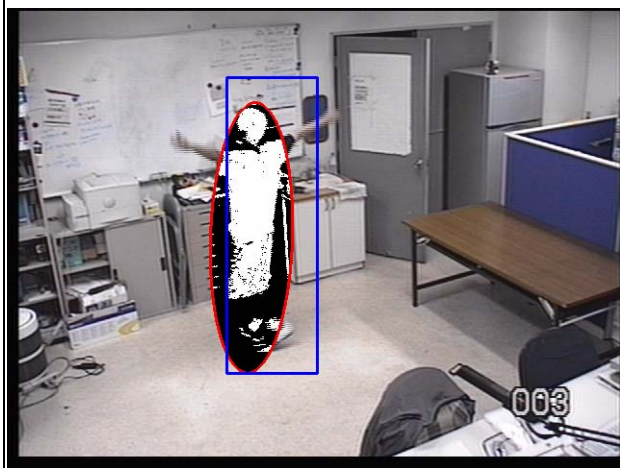




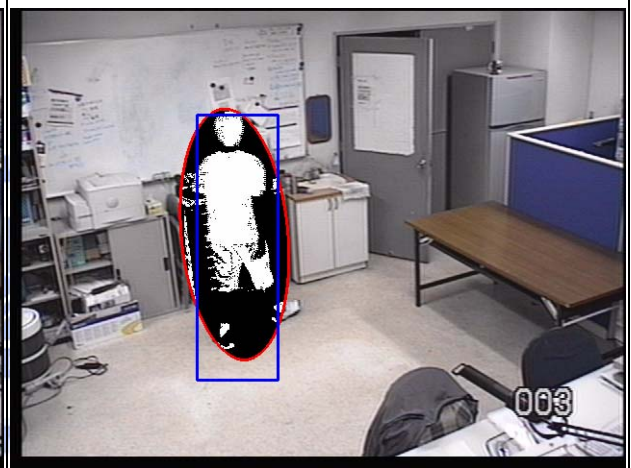
Frame 120



Frame 160



Frame 195



Frame 235



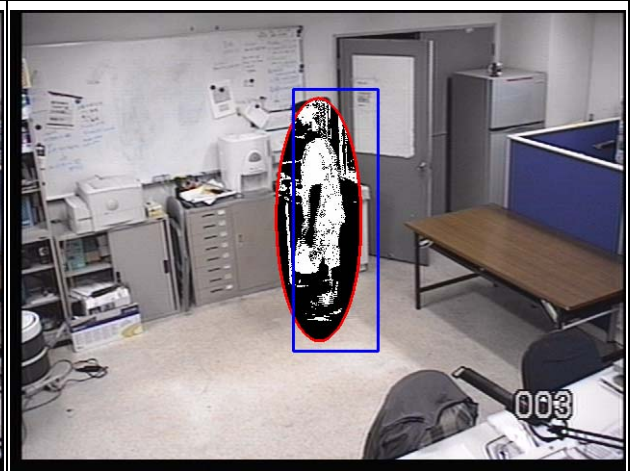
Frame 295



Frame 360



Frame 420



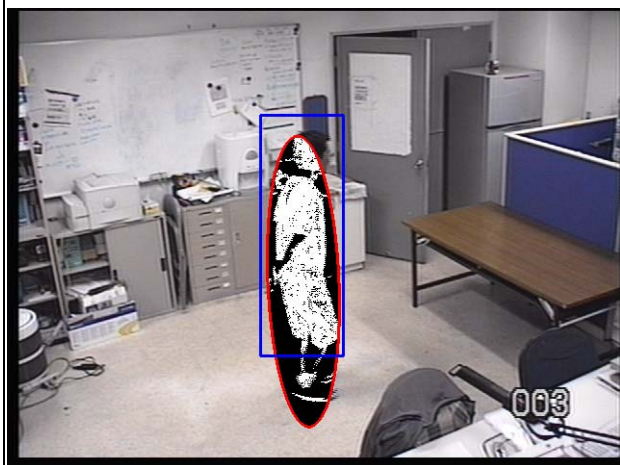
Frame 485



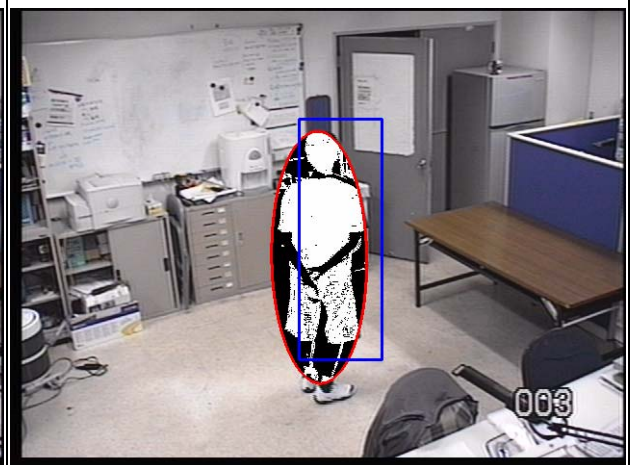
Frame 515



Frame 590



Frame 630



Frame 695

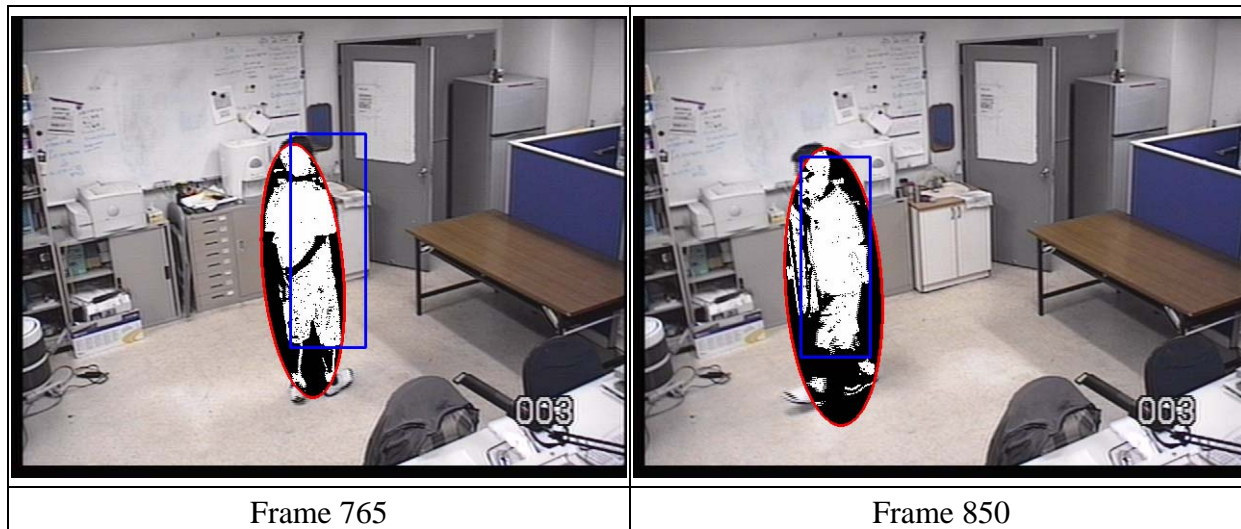


Figure 4.1 Experimental results of the sequence “Hans”. The orientation of the bounding ellipse and the reliability of blobs can be updated during tracking.

We use the RGB color space and separate each channel into 8 bins; therefore, our target model contains 512 blobs. The color of pants belongs to Blobs 147 and 148 and the color of the iron shelves belongs to Blob 148. Therefore, the reliability of Blob 148 should be high when the moving object is far away from the iron shelves but low when they are close. Figure 4.2 shows the reliability of these two blobs are properly updated according to the background information.

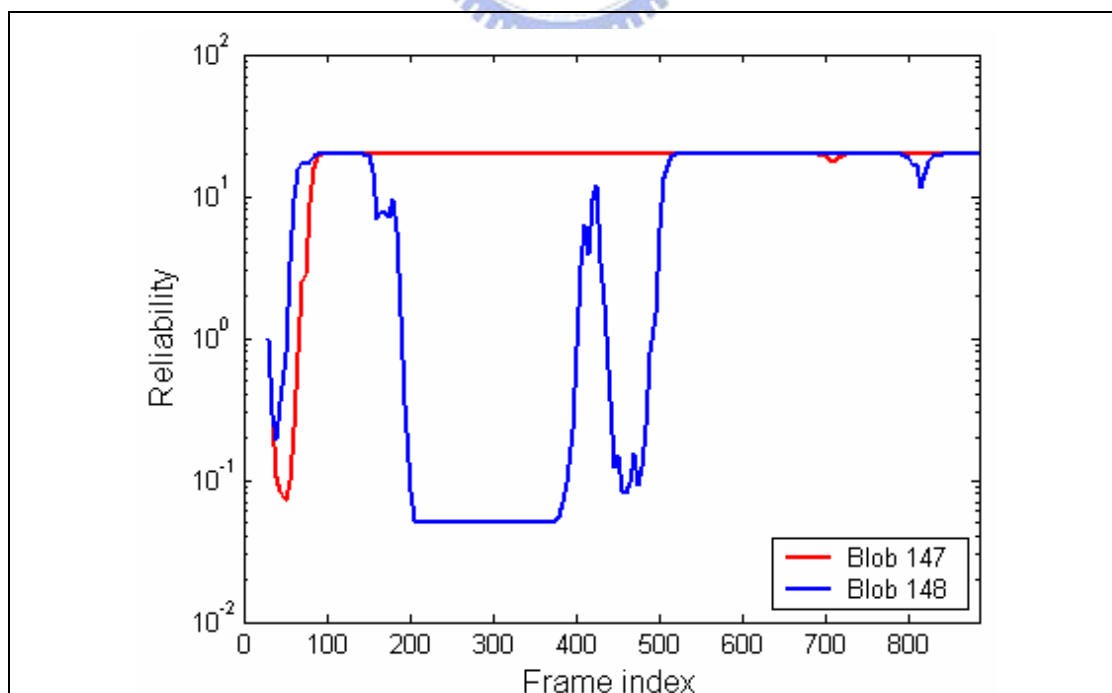
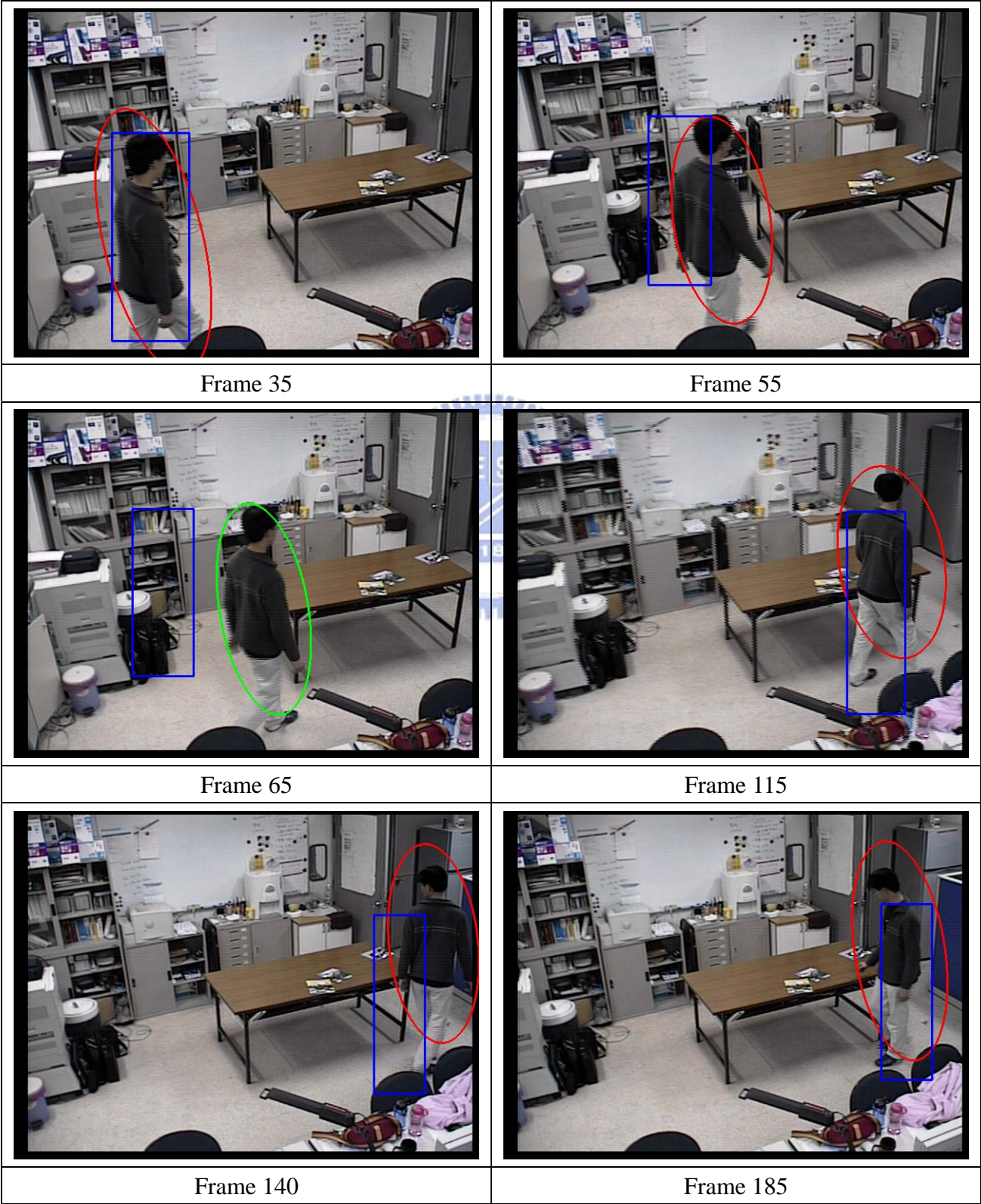


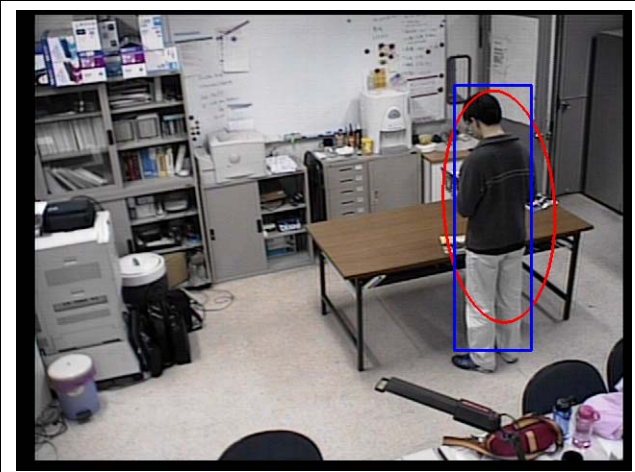
Figure 4.2 The reliability of Blobs 147 and 148. They are updated according to the background information.

In the second experiment a more complex clip is tested. Occlusion and scene change appears. Moreover, the color of cloth is very close to some parts of the background. To test the robustness of our method, we only separate each channel into 4 bins; therefore, our target model contains 64 blobs. Figure 4.3 shows the tracking result. We mark the moving object by the green ellipse instead of the red one when updating the target model.





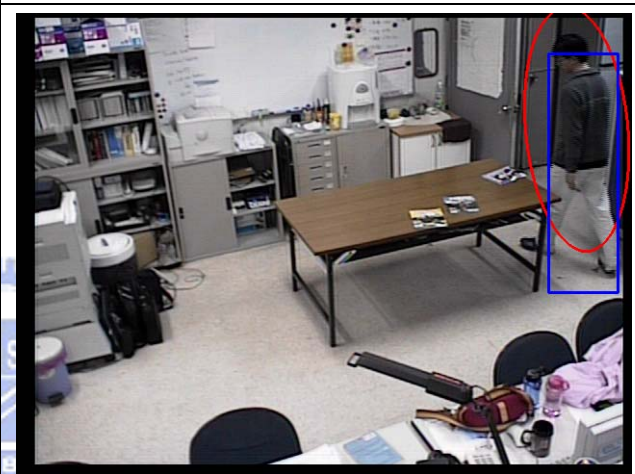
Frame 270



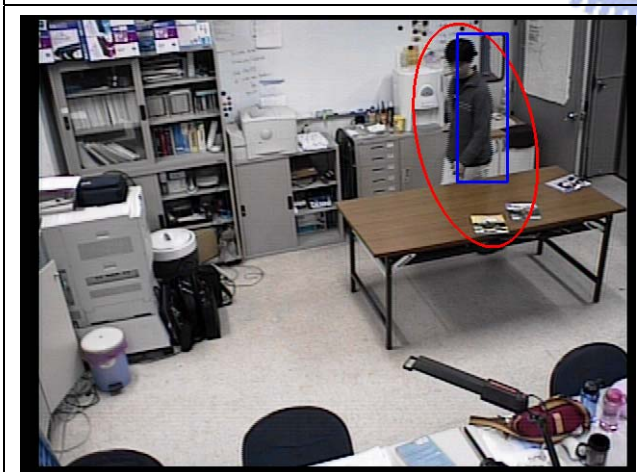
Frame 400



Frame 470



Frame 505



Frame 570



Frame 595



Figure 4.3 The tracking result of sequence *Watson*. The target model updates at frame 65.

Due to the distraction caused by the background, the target model has to update at appropriate moments to track successfully. We use the variance ratio to evaluate the degree of appropriateness. Figure 4.4 shows the variance ratio of the tracking result.

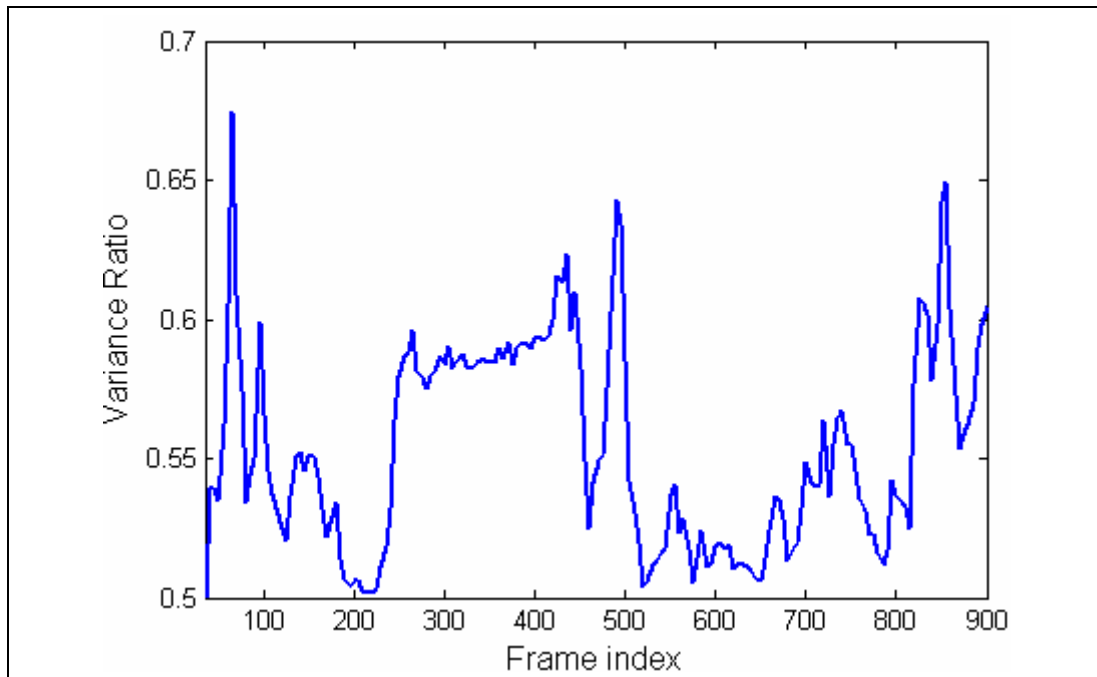


Figure 4.4 The variance ratio of tracking result.

Since all reliabilities are initialized to be 1, variance ratio at Frame 35 is 0.5. Variance ratio changes as the update of reliabilities. At Frame 65, the variance ratio reached the peak value 0.6743 and the algorithm update the target model. The variance ratio drops rapidly when the moving object is close to the door or the shelves or when the occlusion happens.

In the third experiment we ran the same code in the second experiment on the sequence *Wesar*. This sequence contains zooming and the moving object moved away from the camera. Hence, the size of moving object decreased through the sequence. Figure 4.5 shows the result, where the model was built on Frame 35.





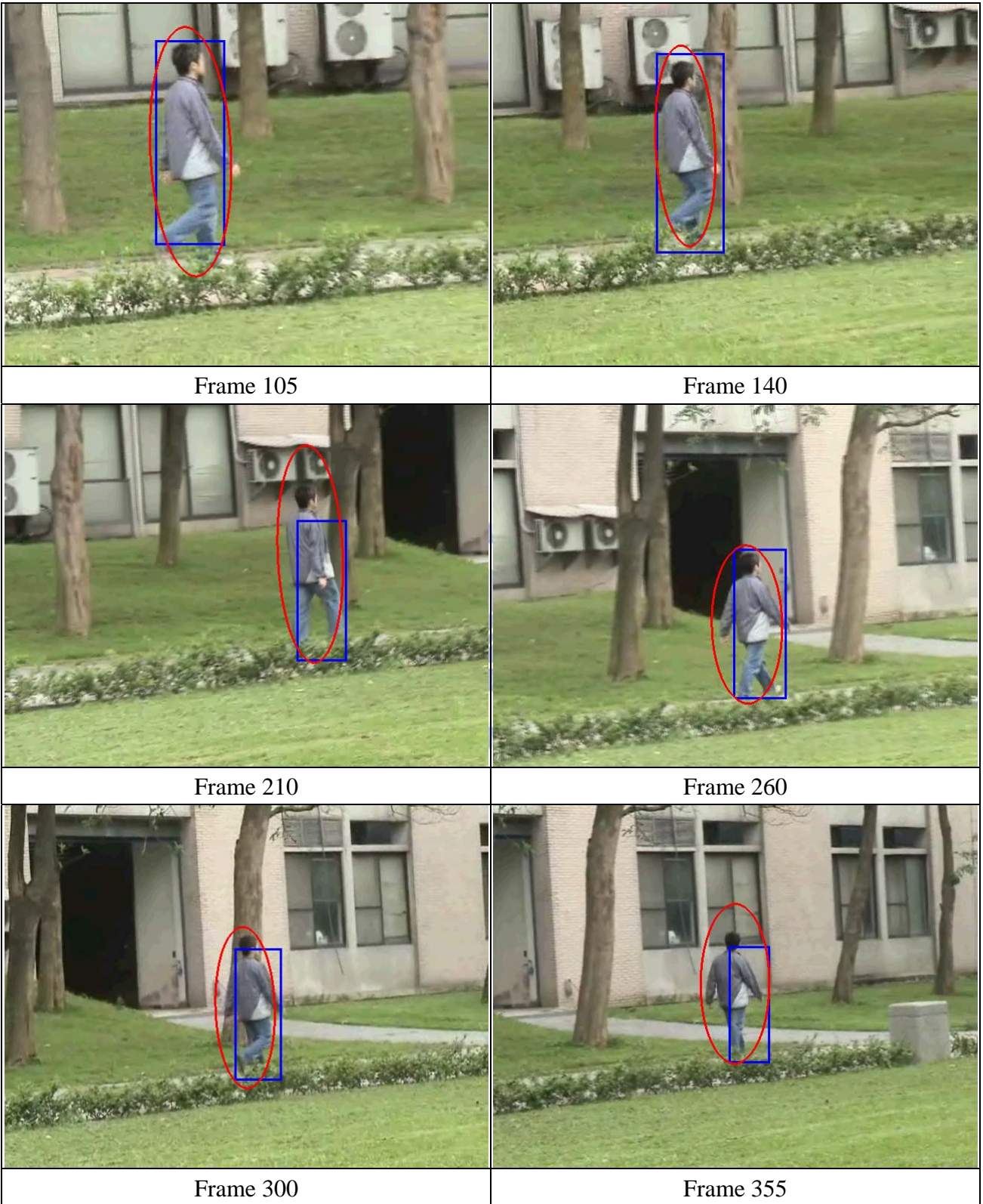


Figure 4.5 The localization of the traditional mean-shift process is poor when the object's size decreases.

Due to the severe scene change and the complex background, size and orientation update is not stable. The reliability update is not sensitive enough under this situation.

The definition of target model affects not only the localization, but also the number of iterations. The better localization will cause a steeper similarity surface and more protruding peaks. Hence, less iterations are needed.

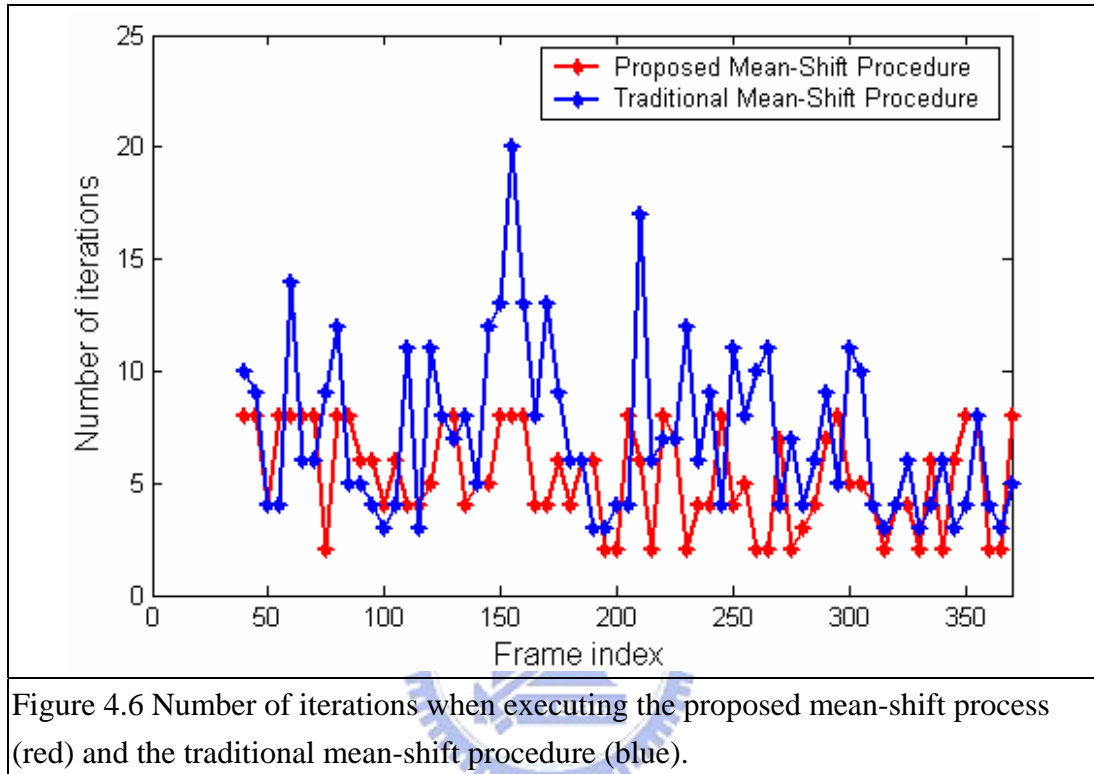
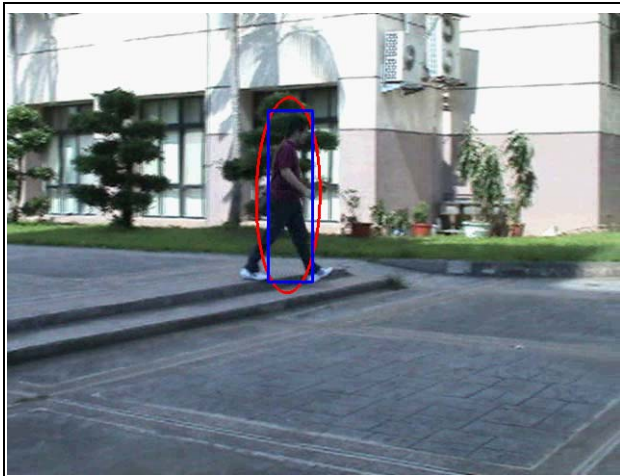


Figure 4.6 Number of iterations when executing the proposed mean-shift process (red) and the traditional mean-shift procedure (blue).

In the final experiment, we use the sequence *Stan*, which contains a great difference in luminance. A complicated background and scene change also appears in the sequence. In Figure 4.7, the object moved from a shadow region to a bright region at Frame 65. The severe change in luminance may cause tracking failure.

In prior researches, other features are used to increase robustness. For example, [13] use 2D image gradients as features, [16] use HSV color space and [17] use so-called excess color features, such as 2G-R-B, etc. However, to select different features to use in different cases is cumbersome. In Section 3.2.8, we proposed two simple rules to judge whether the tracking fails or not.

Whenever the tracker loses the target, our algorithm will restart the tracking process automatically by detecting the moving object again. Figure 4.7 shows the experimental result, where the red ellipse indicates the target. The color is switched from red to green to indicate the restart moment of motion detection..



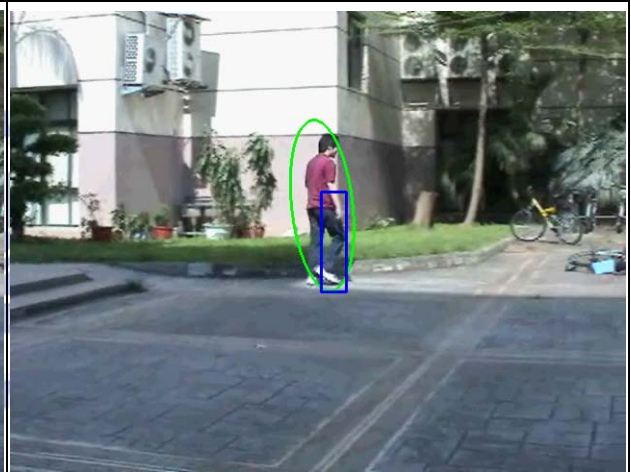
Frame 5



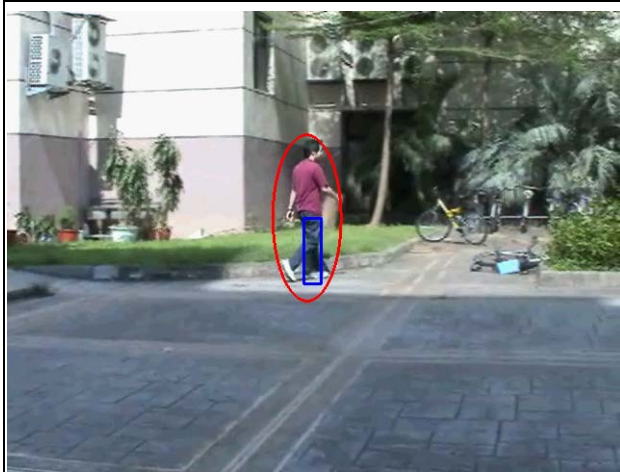
Frame 35



Frame 65



Frame 75



Frame 100



Frame 165

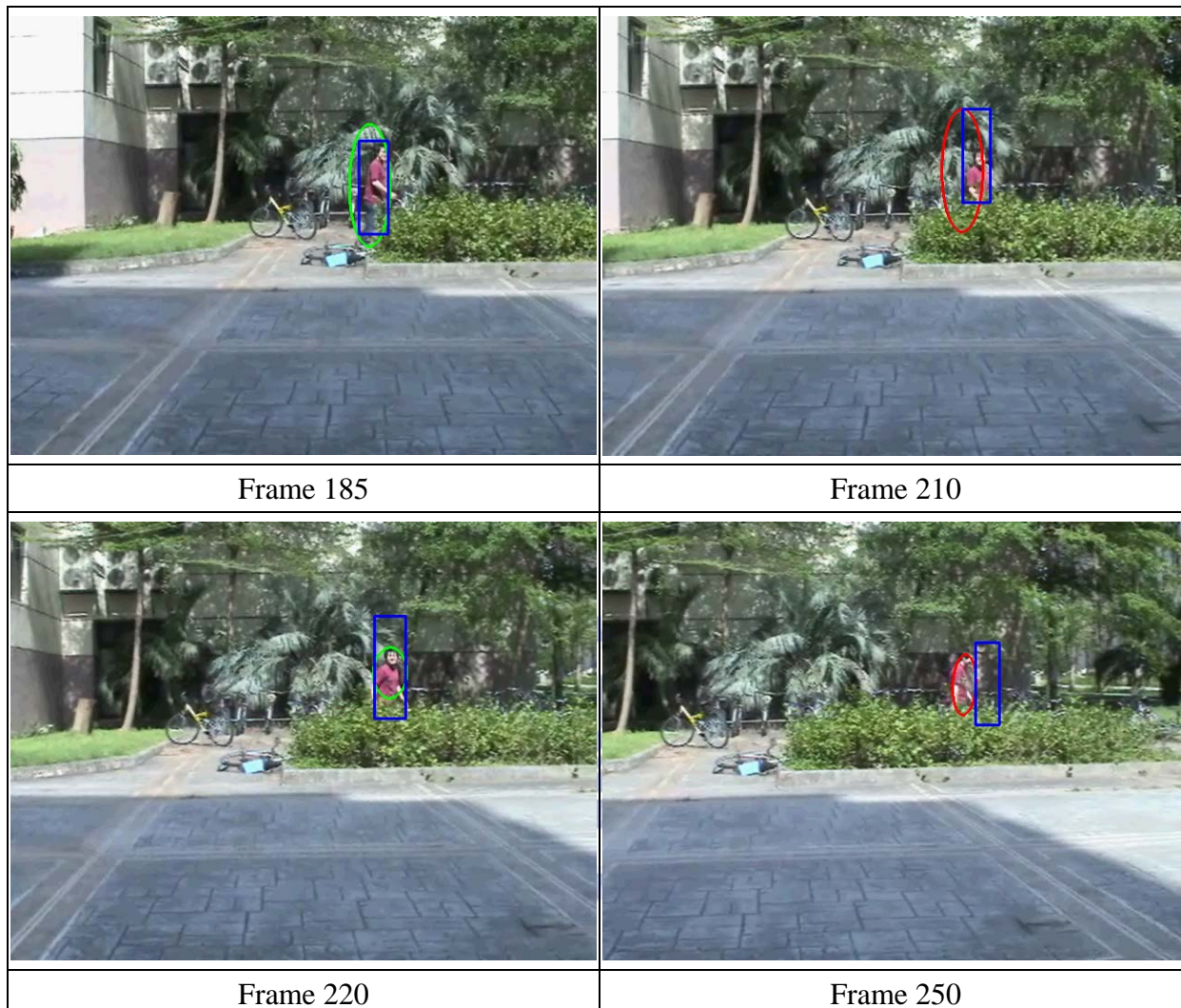


Figure 4.7 The experimental result, where the red ellipse indicates the target. Due to the target loss, the algorithm detected the object at Frame 75, Frame 185 and Frame 220. We switch the color from red to green at these frames.

# Chapter 5.

## Conclusions

We proposed a complete object tracking algorithm including motion detection, motion tracking and target updating. We have presented a new model to describe the target, which contains information in both spatial domain and feature domain. Based on the multi-blob model, we define a similarity measurement and a mean-shift tracking procedure. After the location of moving object has been tracked by mean-shift, we design a process to modify the size and orientation of bounding ellipse. To improve the robustness of the system, we set a target model updating criterion and some rules to check whether the tracking fails or not.

The proposed object tracking system can deal with the case of scene change and occlusion. An outdoor sequence with severe change in luminance is also tested. Due to the localization of multi-blob model and the discriminative similarity measurement, the proposed algorithm converges faster than the traditional one. The size and orientation update is also achieved.



# Reference

- [1] Comaniciu D., Ramesh V., Meer P., “Kernel-based object tracking”, in Proc. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 25, no. 5, pp. 564-575, May 2003.
- [2] W. E. L. Grimson, C. Stauffer, R. Romano, L. Lee, “Using adaptive tracking to classify and monitor activities in a site”, in Proc. IEEE Conf. Computer Vision and Pattern Recognition, Santa Barbara, CA, 1998, pp. 22–31.
- [3] A. Mittal and D. Huttenlocher, “Scene modeling for wide area surveillance and image synthesis”, in Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 2, 2000, pp. 160–167.
- [4] T. Wada, T. Matsuyama, “Appearance sphere: Background model for pan-tilt-zoom camera”, in Proc. 13th Int. Conf. Pattern Recognition pp.A-718-A-722, Vienna, Austria, 1996.
- [5] Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.S., “Background and foreground modeling using nonparametric kernel density estimation for visual surveillance”, in Proc. IEEE Volume 90, Issue 7, July 2002 Page(s):1151 – 1163
- [6] A. J. Lipton, H. Fujiyoshi, R. S. Patil, “Moving target classification and tracking from real-time video”, in Proc. IEEE Workshop Applications of Computer Vision, 1998, pp. 8–14.
- [7] C. R. Wren, A. Azarbayejani, T. Darrell, A. P. Pentland, “Pfinder: real-time tracking of the human body”, in Proc. IEEE Trans. Pattern Anal. Machine Intell., vol. 19, pp. 780–785, July 1997.
- [8] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, H. Wechsler, “Tracking groups of people”, in Proc. Comput. Vis. Image Understanding, vol. 80, no. 1, pp. 42–56, 2000.
- [9] Hironobu Fujiyoshi, Alan J. Lipton, “Real-time human motion analysis by image skeletonization”, in Proc. 4th IEEE Workshop on Applications of Computer Vision (WACV'98), pp. 15-21, October 19 - 21, 1998, Princeton, New Jersey.
- [10] N. Peterfreund, “Robust tracking of position and velocity with Kalman snakes”, in Proc. IEEE Trans. Pattern Anal. Machine Intell., vol. 22, pp. 564–569, June 2000.
- [11] I. A. Karaulova, P. M. Hall, A. D. Marshall, “A hierarchical model of dynamics for tracking people with a single video camera”, in Proc. British Machine Vision Conf., 2000, pp. 262–352.
- [12] D.-S. Jang, H.-I. Choi, “Active models for tracking moving objects”, in Proc. Pattern Recognit., vol. 33, no. 7, pp. 1135–1146, 2000.
- [13] Changjiang Yang, Duraiswami, R., Davis, L., “Efficient mean-shift tracking via a new similarity measure”, in Proc. Computer Vision and Pattern Recognition, 2005. CVPR

2005. IEEE Computer Society Conference on Volume 1, 20-25 June 2005 Page(s):176 - 183 vol. 1 Digital Object Identifier 10.1109/CVPR.2005.139
- [14] Birchfield, S.T., Sriram Rangarajan, "Spatiograms versus histograms for region-based tracking", in Proc. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on Volume 2, 20-25 June 2005 Page(s):1158 - 1163 vol. 2.
- [15] Collins, R.T., "Mean-shift blob tracking through scale space", in Proc. Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on Volume 2, 18-20 June 2003 Page(s): II - 234-40 vol.2
- [16] Zivkovic, Z., Krose, B., "An EM-like algorithm for color-histogram-based object tracking", in Proc. Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on Volume 1, 27 June-2 July 2004 Page(s):I-798 - I-803 Vol.1
- [17] Robert T. Collins, Yanxi Liu, Marius Leordeanu, "Online Selection of Discriminative Tracking Features", in Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 10, pp. 1631-1643, Oct., 2005.

