

# 國立交通大學

電子工程學系電子研究所碩士班

## 碩士論文

基於等位函數法之運動物體偵測與追蹤

Detection and Tracking of Moving Objects based on  
Level Set Theory

研究生：蔡孟修

指導教授：王聖智 博士

中華民國九十五年六月

基於等位函數法之運動物體偵測與追蹤  
Detection and Tracking of Moving Objects based on  
Level Set Theory

研究生：蔡孟修

Student : Meng-Hsiu Tsai

指導教授：王聖智 博士

Advisor : 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

# 基於等位函數法之運動物體偵測與追蹤


研究生：蔡孟修

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

## 中文摘要



監控系統中的單一攝影裝置通常包含建立背景模型、偵測運動物體、追蹤運動物體等三個步驟。在本論文中，我們將討論如何在這些步驟當中，運用等位函數法來記錄運動物體的輪廓。在整個運動偵測與追蹤的過程中，我們首先建立環境模型以利於使用「背景相減法」來達成運動物體偵測。當使用動態攝影機來追蹤運動物體時，由於背景資訊會隨時間改變，此時需要改採「區塊追蹤模型」才能持續追蹤運動物體。為了減低背景環境的干擾，原始的區塊追蹤模型會被加以修改，以考慮前後兩張畫面等位函數曲面之間的交互關係。此外，利用畫面中統計特性加入機率預測的模型，可增強區塊追蹤的強韌性。論文最後會提出一套整合的監控系統架構，架構中的不同元件會選擇採用適當的輪廓模型來分別解決運動物體偵測與追蹤的問題。



# Detection and Tracking of Moving Objects based on Level Set Theory

Student : Meng-Hsiu Tsai

Advisor : Sheng-Jyh Wang

Department of Electronics Engineering, Institute of Electronics  
National Chiao Tung University

## Abstract

An intelligent video surveillance system usually performs the tasks of background modeling, motion detection, and tracking. In this thesis, a level set function is used to record the moving objects for these three operations. The background model is first constructed before the background subtraction is performed for the motion detection. Then, a mobile camera keeps tracking the moving objects with a region tracking model. The original region tracking model is modified to alleviate the interference of cluttered environment. The relation between two level surfaces of successive two frames is taken into consideration. The probability model built from the statistic property of an image is also included. Finally, an integrated surveillance system is proposed. Different units in the surveillance system may choose appropriate contour models to solve their problems.



# 誌謝

感謝王聖智老師兩年多來的教導，老師豐富的學識與涵養讓我獲益良多。感謝實驗室的所有同學，陪我度過這段大家共同學習的日子。感謝與我一起加油打氣的朋友們，你們給我的鼓勵是讓我在挫折中堅持到最後的原動力。感謝我的家人，我珍惜你們多年來對我的支持與包容，我愛你們。







# Contents

中文摘要 .....	i
Abstract .....	iii
誌謝 .....	v
Chapter 1. Introduction .....	1
Chapter 2. Active Contours .....	5
2.1 Snake Model .....	5
2.2 Geodesic Active Contour .....	6
2.3 Level Set Theory .....	11
2.4 Active Region Model .....	15
Chapter 3. Detection and Tracking of Moving Objects .....	23
3.1 Motion Detection with Background .....	23
3.1.1 Active Contours without Edges .....	24
3.1.2 Background Subtraction .....	30
3.2 Region Tracking .....	33
3.2.1 Region Tracking without Motion Computation .....	33
3.2.2 A New Region Tracking Model .....	37
3.3 Probability Model .....	42
3.3.1 Kernel Density Estimation .....	43
3.3.2 ML and MAP Estimation .....	45
Chapter 4. Implementation Issues and Experimental Results .....	51
4.1 Implementation Issues .....	51
4.1.1 Distance Transform .....	51
4.1.2 Solve PDE Numerically .....	55
4.1.3 Re-initialization .....	58
4.2 Build Background .....	63
4.2.1 Motion Detection without Background .....	63
4.2.2 Background Modeling .....	66
4.3 Experimental Results .....	68
Chapter 5. Conclusions .....	75
References .....	77

# List of Figures

<b>Figure 1-1</b>	General framework of a visual surveillance system [20].	2
<b>Figure 2-1</b>	Snake Model.	5
<b>Figure 2-2</b>	Level Surface and Contours	12
<b>Figure 2-3</b>	Geodesic Active Contour initialized totally outside the objects.	15
<b>Figure 2-4</b>	Geodesic Active Contour with bad initialization.	15
<b>Figure 2-5</b>	Stoke's Theorem	17
<b>Figure 2-6</b>	Convergence of active contours with region information.	20
<b>Figure 2-7</b>	Convergence of active contours with some objects are partly outside the image.	20
<b>Figure 2-8</b>	Convergence of active contours when the initial curve is too small.	21
<b>Figure 3-1</b>	Background Subtraction. (a) Background Image. (b) Frame 180.	24
<b>Figure 3-2</b>	$F_1(C)$ and $F_2(C)$ . Four situations of $F_1(C)$ and $F_2(C)$ with different contours. When the image data inside and outside the contours are not uniform, the values of $F_1(C)$ or $F_2(C)$ produce a positive value. Only when the contours matches the boundaries of the objects, both $F_1(C)$ and $F_2(C)$ are zero and the energy function in (Eq. 3-1) reaches its minimum.	25
<b>Figure 3-3</b>	Active contours without edges. $c_1$ and $c_2$ are the average values inside and outside the contours, respectively.	26
<b>Figure 3-4</b>	Apply the "active contours without edges" method to the Gaussian smoothed image. $c_1$ and $c_2$ are the average values inside and outside the contours, respectively.	27
<b>Figure 3-5</b>	Apply the "active contours without edges" approach to the image with Gaussian noise. The standard deviation of the noise is 30.	27
<b>Figure 3-6</b>	Statistical histogram of Figure 3-5. The blue bar is the histogram of the whole image, the red bar is the histogram of the inside region, and the green bar is the histogram of the outside region.	28
<b>Figure 3-7</b>	Apply the "active contours without edges" approach to the image with Gaussian noise. The standard deviation of the noise is 80.	29
<b>Figure 3-8</b>	Apply the "active contours without edges" approach to the image with improper initialization. The standard deviation of the noise is 20. Note that $c_1$ is larger than $c_2$ . The region "inside" the contours is actually the background region.	29
<b>Figure 3-9</b>	Classify the difference image data of Frame 180 and Frame 1 of the	

	“Hall” test sequence.....	31
<b>Figure 3-10</b>	Classify the difference image data of Frame 180 and Frame 1 of the “Hall” test sequence. The initial contour is replaced by a few small contours.....	31
<b>Figure 3-11</b>	Apply the “active contours without edges” model to the hall sequences. Frame 1 is taken as the background. Multiple objects are successfully identified based on the level set approach. ....	32
<b>Figure 3-12</b>	Comparison of required time of two different initializations. The solid line denotes the required time based on the large contour, while the dot line denotes the required time based on the set of small contours.....	32
<b>Figure 3-13</b>	The force $F$ defined by the region tracking model. (a) The pre-defined region in the reference frame. (b) The pixels where $F < 0$ in the current frame.....	37
<b>Figure 3-14</b>	Apply the morphing operator to the level surface of the reference frame. (a) The level surface of the reference frame. (b) Morphing result. (c) Apply a low-pass filtering and the re-initialization to the surface in (b). (c) is used to estimate the new level surface in the current frame. ....	38
<b>Figure 3-15</b>	The region force with and without the weight $P_\phi$ . The cracked false detected points are reduced by adding $P_\phi$ . The influence of the surrounding environment is reduced and makes the region tracking model more robust. ....	39
<b>Figure 3-16</b>	The car-phone test sequences. The face region is defined in the first frame. The level surface is only updated inside the green rectangle. The face is successfully tracked even though the scene changes outside the car.....	40
<b>Figure 3-17</b>	Football sequences. The camera moves very quickly and the target is blurred in some frames. The contour does not grow incorrectly due to the use of $\varphi_{estimate}^n(\bar{x})$ in (Eq. 3-17).....	40
<b>Figure 3-18</b>	Football sequences with original region tracking model. The contour grows incorrectly because of the interference of the environment clutter. ....	41
<b>Figure 3-19</b>	Track a small target in the coastguard sequence. The level surface with higher resolution is only evaluated inside the green rectangle. More computational resources can be allotted to a specific region and produce a higher quality result.....	41

<b>Figure 3-20</b>	Use a level surface with lower resolution. This reduces large computational cost. The essential position is successfully tracked but the fine boundary of the object is lost.....	42
<b>Figure 3-21</b>	The blue line is the histogram of the gray-level Lena image. The red line is the kernel density estimation result.....	44
<b>Figure 3-22</b>	Improper kernel bandwidth selection. If the bandwidth is too small, it causes under-smoothed. If the bandwidth is too large, it causes over-smoothed.....	44
<b>Figure 3-23</b>	Maximum likelihood. (a) The preset contours which define the objects in the reference frame. (b) The objects move in the current frame. (c) The probability of inside region in the current frame. (d) The probability of outside region in the current frame. (e) and (f) are the histograms inside and outside the contours in (a).....	45
<b>Figure 3-24</b>	Maximum likelihood estimation. (a) Define the object in the reference frame. (b) The current frame. (c) The probability of the inside region according to the distribution in the RGB space. The RGB space is divided into $16 \times 16 \times 16$ bins.....	46
<b>Figure 3-25</b>	Obtain the prior information. The current frame is divided into many blocks. Every block finds the most matched region within a search range in the reference frame. Then the proportion of $R_0$ is used to estimate the prior $P(IN)$ for all pixels of Block( $i, j$ ) in the current frame.....	48
<b>Figure 3-26</b>	The probability characteristic of the current frame in Figure 3-24 (b). (a) and (b) are the color version of Figure 3-23 (c) and (d). (c) is the prior information obtained by (Eq. 3-22). (d) is the MAP estimation from (Eq. 3-21).....	49
<b>Figure 3-27</b>	Merge the inside information of the new region tracking model and the probability model. The total inside information can be formed by using different amounts of weight for the two components.....	49
<b>Figure 3-28</b>	The experimental result of real sequences acquired in the laboratory. The contour of the person is preset in the first frame.....	50
<b>Figure 4-1</b>	Distance Transform. In (b), the values outside the object depend on their distance away from the object.....	52
<b>Figure 4-2</b>	Two-pass algorithm in 2D. The binary input image is shown as the right star image. The first pass accumulates the distance value from top-left to bottom-right, while the second pass accumulates the distance value from bottom-right to top-left.....	53
<b>Figure 4-3</b>	Four distance measures for creating a distance function.....	54

<b>Figure 4-4</b>	Error propagation result due to improper approximation of $\nabla\phi$ .....	55
<b>Figure 4-5</b>	The propagation of the 1-D level set function. The solid blue line is the level set function and the green line is the value of the velocity function. ....	57
<b>Figure 4-6</b>	Re-initialization demonstration of (Eq. 4-8). The green line is the preset level function and the blue line shows the variation of the level function according to the re-initialization procedure. The time step is set to be 0.5. ....	59
<b>Figure 4-7</b>	The propagation of the level set function with velocity $F$ in Figure 4-5 and adopt the re-initialization at the end of each time step. The glitches in Figure 4-5 are eliminated. ....	60
<b>Figure 4-8</b>	The re-initialization in two dimension space. The level function is initially set to be -1 inside the objects and 1 outside the objects. ....	62
<b>Figure 4-9</b>	Motion Detection by thresholding the inter-frame difference image. The black pixels represents the motion block with the measurement $d$ . ....	64
<b>Figure 4-10</b>	Make a solid detection result. (a) The blue blocks show the original SAD detection result. (b) Morphological closing with a 3x3 structuring element. (c) Morphological closing with a 5x5 structuring element. (d) Fill the hole inside the moving object via (Eq. 4-11).....	65
<b>Figure 4-11</b>	Contour propagation using (Eq. 4-11). ....	66
<b>Figure 4-12</b>	Test sequences for background modeling. ....	67
<b>Figure 4-13</b>	Background modeling process with $n = 30$ . The white regions are the background region. ....	67
<b>Figure 4-14</b>	Background modeling using different $n$ 's. (a) $n = 30$ , (b) $n = 60$ . The discontinuous regions in the pink ellipses are due to the different intensity values from different frames. ....	68
<b>Figure 4-15</b>	A simplified version of Figure 1-1. Every camera contains “background modeling”, “background subtraction” and “region tracking” processes. ....	69
<b>Figure 4-16</b>	Flow chart of a single camera’s system. This figure shows the details about how to control the process of every camera in Figure 4-15. ....	70
<b>Figure 4-17</b>	The simulation result of Figure 4-16. The images with green grid and blue mask are in the background modeling status. When the background is modeled in all blocks, a background subtraction result is obtained. The frames with a green rectangle represent the tracking results. ....	74
<b>Figure 4-18</b>	The consuming time of processing the sequence in Figure 4-17. The frames with small time consuming are in the status of background	

modeling or background subtraction. The frames with a relatively large time consuming are in the status of region tracking. .... 74

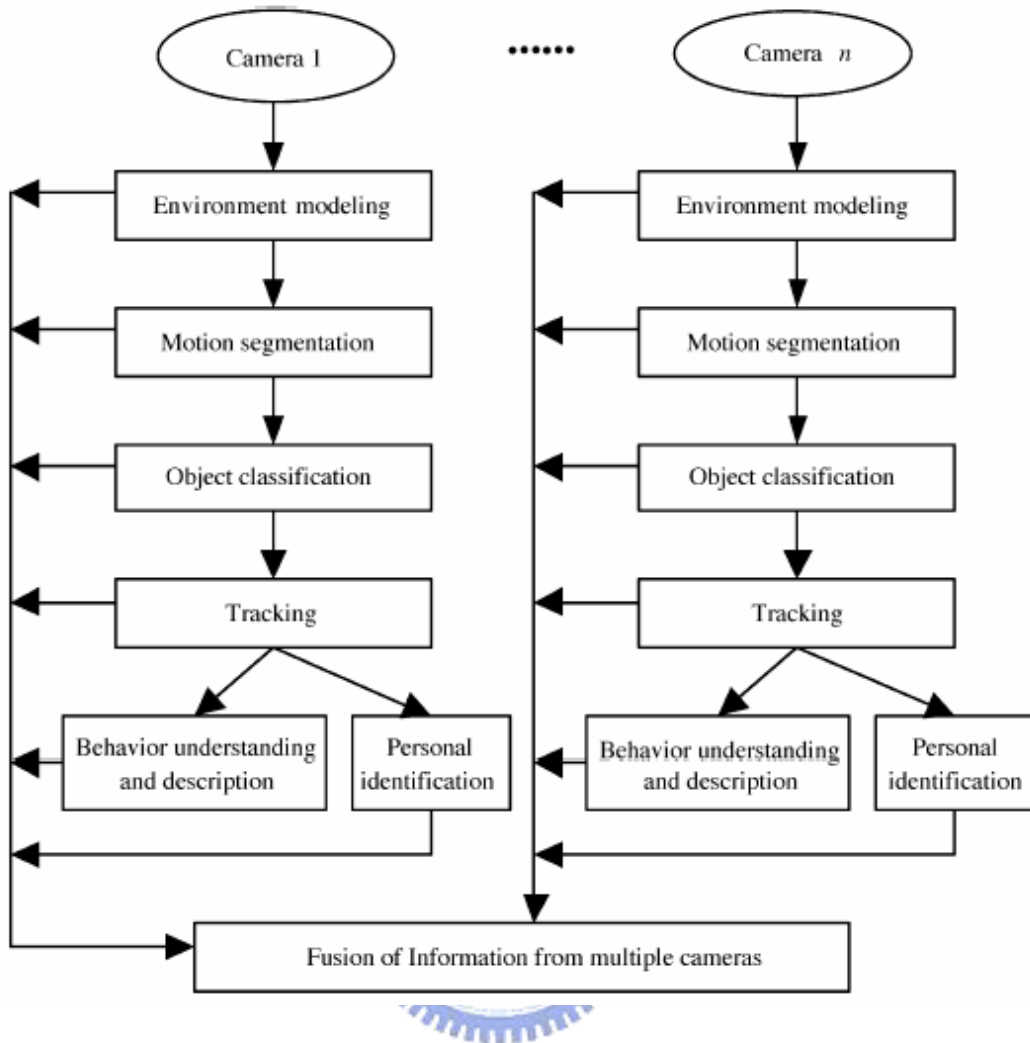


# Chapter 1. Introduction

The applications of visual surveillance systems are getting more and more popular. Image capture devices become more available today. Moreover, powerful computers make complex digital image process more realizable. To make the image acquisition process more valuable, visual surveillance systems become smarter than before. Here, the analysis of moving objects could be the first important step. Some associated topics will be introduced and a brief introduction of this thesis will be given at the end of this chapter.

The visual surveillance system proposed in [20] is shown as Figure 1-1. A complete surveillance system may contain multiple cameras. These cameras cooperate with each other and the acquired information is fused. The tasks after the video acquisition step can be divided into two parts. One is motion detection and the other one is motion tracking.

In Figure 1-1, the detection part contains “environment modeling”, “motion segmentation”, and “object classification”. The environment or the background should be modeled before the moving objects can be recognized. Ahmed Elgammal *et al* [18] proposed a way to model the foreground and background by using nonparametric kernel density functions. In [19], a graph method like a finite state machine records the status of each pixel. The motion region is stored in a layer map. In [21], salient motion is detected without background information. This approach can resolve problems caused by luminance variation and slight movement of the background. (ex. wavering trees in an outdoor scene).



**Figure 1-1** General framework of a visual surveillance system [20].

Regarding tracking, there exist many different methods for the tracking of objects in image sequences. Mean-shift [17] is a robust method to track an ellipse or rectangle region. Even though the size of the tracking window may change adaptively, this method doesn't extract the boundary of moving objects. On the other hand, the snake model [1] proposed by Kass builds an energy function to control contour propagation. The contour finally locates at the boundary of an object to reach a local minimum of the energy function. Micheal Isard and Andrew Blake use B-spline curves to parameterize the contour for their CONDENSATION algorithm [22]. To further improve the performance of the snake model, Nikos Paragios *et al* [23] use the Geodesic Active Contours model to detect and track moving objects. The contours are described by a level surface. The level-set theory constructs the relation between the contours and the level surface. This level surface is very useful in the recording of multiple objects. In [25], the residuals of several frames after motion compensation are collected by a 3-dimensional level set function. This is a method to accomplish the



contour tracking for sequences acquired by a mobile camera.

Level set theory is a very powerful tool to handle contours. Once the contour propagation model is defined, it can be implemented by a level surface. The snake model [1] can be transformed into a geodesic active contour model based on [2]. The optimal contour propagation can be found by using the steepest descent method. Then, an associate level surface is obtained. The active region model proposed in [7][24] takes the statistical property into account. The “active contours without edges” model [5] is even more robust in the classification of the image data into two regions in the statistical sense. This model can be used to decompose the background subtraction result into motion region and static region. On the other hand, the region tracking model [8] tracks the most similar regions in two successive frames. This model is very useful for sequences acquired by a mobile camera [9][10]. In Section 3.2, a further improved model will be proposed. The maximum likelihood (ML) and maximum a posteriori (MAP) estimations are then discussed in Section 3.3.

There are many issues about implementing the level set theory. Distance transform is a main topic. The “two-pass algorithm” offers a way to approximate a distance transform. On the other hand, partial differential equations may cause problems since all computations are done in the discrete domain. The essentially non-oscillatory (ENO) polynomial interpolation [11][12][13] has to be used to avoid errors. The re-initialization is also introduced in [12][13] to make the contour propagation more stable. A larger value of time step can be used with re-initialization to speed up the evolution. On the other hand, Roman Goldenberg *et al* [26] proposed a fast geodesic active contour model. The computation of divergence is decomposed based on the additive operator splitting (AOS) scheme. In [27], the computation of PDE is completely unnecessary. The level surface is replaced by 4 values to represent the inside and outside regions and the contours. The contours are evaluated by the “check in” and “check out” subroutines. In [28], Yonggang Shi *et al* applied their fast level set method to the tracking problem.

Some basic backgrounds about the contours and level set theory are introduced in Chapter 2. Chapter 3 introduces new contour models to deal with motion detection and region tracking problems. Chapter 4 discusses some implementation issues. A simple method is proposed to build the background. Finally an integrated system with background modeling, motion detection, and region tracking is proposed. The conclusions and the future work are made in Chapter 5.

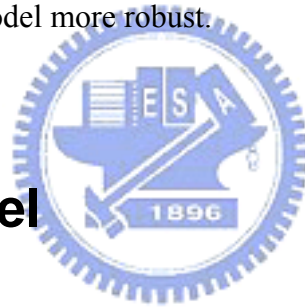


# Chapter 2. Active Contours

The basic concept of active contour can be found in Kass's paper in [1]. An active contour is named as "snake" at the first time. The shape and the propagation of a contour are controlled by the internal and the external forces. The curvature of the contour determines the internal force, while the image characteristics on the contour define the external force. The contour stops propagating when the sum of the internal force and the external force reaches a local minimal.

However, Kass's method has some limitations. The main drawback is that a good initialization is required. Another drawback is that multiple contours require multiple initializations. In comparison, based on the level surface, multiple contours can be described at the same time. The level set theory relates the contour propagation model with the update of the level surface. Finally the adoption of the "active region" concept makes the contour model more robust.

## 2.1 Snake Model



The snake model was first proposed by Kass in 1988 [1]. As shown in Figure 2-1, the red curve is initialized in the left image. The contour propagates toward the object's boundary to minimize the energy function. When the contour touches the boundary, as shown in the right image, it corresponds to the minimal energy. In general, the energy function contains internal energy, image energy, and additional constraints.

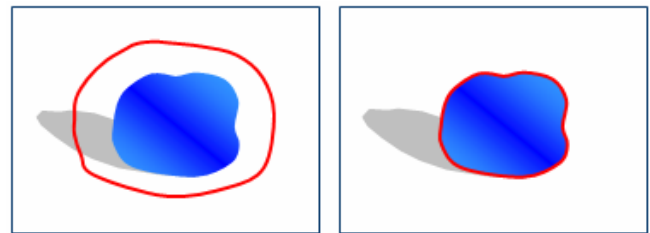


Figure 2-1 Snake Model.

$$E[(C)(p)] = \alpha \int_0^1 E_{int}(C(p))dp + \beta \int_0^1 E_{img}(C(p))dp + \gamma \int_0^1 E_{con}(C(p))dp$$

(Eq. 2-1)

Here,  $E_{\text{int}}(C(p))$  denotes the internal energy. This term depends on the smoothness, or the inverse of curvature, of the contour. If the curvature of the contour is large, the contour contains higher energy. Otherwise, the internal energy is low.

$E_{\text{img}}(C(p))$  denotes the image energy. This term is affected by the property of the image data on the contour. Intensity values, colors, or edges are usually used to define the image energy term. For example: a pixel with a higher gradient value contains lower energy.

$E_{\text{cons}}(C(p))$  denotes the constrained energy. The additional constraint can be used to make the results more desirable.

This snake model needs to be supervised by user and the determination of the parameters  $\alpha$ ,  $\beta$ ,  $\gamma$  are case by case. Because the solution is a local minimum, the initial curve must be preset near the boundary. Besides, every contour needs its own energy function. Multiple contours need their corresponding individual energy functions. Because the number of objects in the image is usually initially unknown, this contour model is inadequate for the object contour tracking problem.

To overcome these drawbacks, the snake model can be transformed into a “geodesic active contour” model. This model combines the internal energy and the image energy in one product term for every contour. Then, multiple contours can be described in a single energy function.

## 2.2 Geodesic Active Contour

In the geodesic active contour model, the classical snake model is rewritten as

$$E(\bar{C}) = \alpha \int_0^1 |\bar{C}'(q)|^2 dq + \beta \int_0^1 |\bar{C}''(q)|^2 dq - \lambda \int_0^1 |\nabla I(\bar{C}(q))| dq \quad (\text{Eq. 2-2})$$

where  $\bar{C}(q): [0,1] \rightarrow R^2$  represents a parameterized planar curve, and  $I(x): [0,a] \times [0,b] \rightarrow R^+$  denotes the image data at the position  $x$ . The first two terms are the internal energy terms. The integration of  $|\bar{C}'(q)|$  is the length of the contour. The integration of  $|\bar{C}''(q)|$  represents the bending property of the curve. Minimizing this energy function means that the length and the curvature of the contour must be as small as possible. In other words, the curve tends to become a small circle if there is

no other force.

The negative term is used to make the image energy near the boundary small. When the contour touches the object's boundary, the integral of  $|\nabla I(\bar{C}(q))|$  is very large. Then the total value of the energy function reaches its minimum.

The snake propagation can be split into two steps. When the contour does not reach the boundary of the object, the energy is minimized by shortening the contour and making the contour smoother. The energy decreases rapidly when the contour touches the boundary with high gradient values.

Defining a decreasing function  $g(x):[0,+\infty] \rightarrow R^+$  such that  $g(x) \rightarrow 0$  as  $x \rightarrow \infty$ , (Eq. 2-1) becomes a general energy function by taking  $\beta = 0$ :

$$E(C) = \alpha \int_0^1 |\bar{C}'(q)|^2 dq + \int_0^1 g(|\nabla I(\bar{C}(q))|)^2 dq \quad (\text{Eq. 2-3})$$

The above energy minimization problem is equivalent to finding a geodesic curve in the Riemannian space. The details are based on the Riemannian geometry. The derivation in [2] uses the Maupertuis' Principle [3] to prove that minimizing (Eq. 2-3) is equivalent to minimizing

$$L_R = \int_0^1 g(|\nabla I(\bar{C}(q))|) |\bar{C}'(q)| dq \quad (\text{Eq. 2-4})$$

Readers can find another proof in [4]. (Eq. 2-4) is called the geodesic active contour model. It can be viewed as calculating the length of  $\bar{C}(q)$  weighted by  $g(|\nabla I(\bar{C}(q))|)$ . When the contour passes by the boundary of the object, the value of  $g(|\nabla I(\bar{C}(q))|)$  is small and the energy is minimized. The addition of  $|\bar{C}'(q)|$  and  $g(|\nabla I(\bar{C}(q))|)$  in (Eq. 2-3) is transformed into the multiplication in (Eq. 2-4). This model can define multiple contours in one energy function because the property of each contour is described in one product term.

Now the propagation of the contour can be derived from (Eq. 2-4) by the Euler-Lagrange approach. Assume the contour  $\bar{C}(t, q) = [x(t, q) \ y(t, q)]^T$  is controlled by the parameter  $q \in [0, 1]$  and the time  $t$ . In order to find the variation of  $L_R$  with respect to time  $t$ , the derivative of  $L_R$  in (Eq. 2-4) is computed as:

$$\frac{d}{dt} L_R(\bar{C}(t)) = \int_0^1 \frac{d}{dt} g(\bar{C}(t, q)) |\bar{C}_q(t, q)| dq + \int_0^1 g(\bar{C}(t, q)) \frac{d}{dt} |\bar{C}_q(t, q)| dq \quad (\text{Eq. 2-5})$$

In the first term of (Eq. 2-5), we have

$$\begin{aligned} \frac{d}{dt} g(\bar{C}(t, q)) &= \frac{d}{dt} g([x(t, q) \ y(t, q)]^T) \\ &= \frac{\partial g(\bar{C}(t, q))}{\partial x} \frac{dx}{dt} + \frac{\partial g(\bar{C}(t, q))}{\partial y} \frac{dy}{dt} = \langle \nabla g(\bar{C}(t, q)), \bar{C}_t(t, q) \rangle, \end{aligned} \quad (\text{Eq. 2-6})$$

where  $\langle \bar{a}, \bar{b} \rangle$  is the inner product of vectors  $\bar{a}$  and  $\bar{b}$ .

Then, the first term of (Eq. 2-5) becomes

$$\int_0^1 \frac{d}{dt} g(\bar{C}(t, q)) |\bar{C}_q(t, q)| dq = \int_0^1 \langle \nabla g(\bar{C}(t, q)), \bar{C}_t(t, q) \rangle |\bar{C}_q(t, q)| dq. \quad (\text{Eq. 2-7})$$

Now consider the second term of (Eq. 2-5).

Since  $|\bar{C}_q(t, q)|^2 = \langle \bar{C}_q(t, q), \bar{C}_q(t, q) \rangle$  and  $\frac{d}{dt} \langle \bar{u}, \bar{u} \rangle = 2 \langle \bar{u}, \bar{u}_t \rangle$ , we have

$$\begin{aligned} \frac{d}{dt} |\bar{C}_q(t, q)| &= \frac{d}{dt} \left( \frac{|\bar{C}_q(t, q)|^2}{|\bar{C}_q(t, q)|} \right) = \frac{d}{dt} \left( \frac{\langle \bar{C}_q(t, q), \bar{C}_q(t, q) \rangle}{|\bar{C}_q(t, q)|} \right) \\ &= \frac{2 \langle \bar{C}_q(t, q), \bar{C}_{iq}(t, q) \rangle |\bar{C}_q(t, q)| - \langle \bar{C}_q(t, q), \bar{C}_q(t, q) \rangle \frac{d}{dt} |\bar{C}_q(t, q)|}{|\bar{C}_q(t, q)|^2}. \end{aligned}$$

$$\text{Thus, } \frac{d}{dt} |\bar{C}_q(t, q)| = \frac{\langle \bar{C}_q(t, q), \bar{C}_{iq}(t, q) \rangle}{|\bar{C}_q(t, q)|} = \left\langle \frac{\bar{C}_q(t, q)}{|\bar{C}_q(t, q)|}, \bar{C}_{iq}(t, q) \right\rangle = \langle \bar{T}(t, q), \bar{C}_{iq}(t, q) \rangle, \quad (\text{Eq. 2-8})$$

where  $\bar{T}(t, q) = \frac{\bar{C}_q(t, q)}{|\bar{C}_q(t, q)|}$  is the unit tangent vector on the contour  $\bar{C}(t, q)$ .

Use (Eq. 2-8), the second term in (Eq. 2-5) becomes

$$\int_0^1 g(\bar{C}(t, q)) \frac{d}{dt} |\bar{C}_q(t, q)| dq = \int_0^1 g(\bar{C}(t, q)) \langle \bar{T}(t, q), \bar{C}_{iq}(t, q) \rangle dq. \quad (\text{Eq. 2-9})$$

Then, we may apply the “integration by part” formula  $\int \langle \bar{u}, d\bar{v} \rangle = \langle \bar{u}, \bar{v} \rangle - \int \langle \bar{v}, d\bar{u} \rangle$

and set

$$\begin{aligned}\bar{u} &= g(\bar{C}(t, q)) \bar{T}(t, q) \\ d\bar{u} &= \left[ \frac{d}{dq} g(\bar{C}(t, q)) \right] \bar{T}(t, q) dq + g(\bar{C}(t, q)) \bar{T}_q(t, q) dq \\ &= \langle \nabla g(\bar{C}(t, q)), \bar{C}_q(t, q) \rangle \bar{T}(t, q) dq + g(\bar{C}(t, q)) \bar{T}_q(t, q) dq,\end{aligned}$$

where  $\frac{d}{dq} g(\bar{C}(t, q)) = \langle \nabla g(\bar{C}(t, q)), \bar{C}_q(t, q) \rangle$  is similar to (Eq. 2-6) and

$d\bar{v} = \bar{C}_{iq}(t, q) dq$ ,  $\bar{v} = \bar{C}_i(t, q)$ . With the above deduction, (Eq. 2-9) becomes

$$\langle g(\bar{C}(t, q)) \bar{T}(t, q), \bar{C}_i(t, q) \rangle \Big|_0^1 - \int_0^1 \langle \bar{C}_i(t, q), [\langle \nabla g(\bar{C}(t, q)), \bar{C}_q(t, q) \rangle \bar{T}(t, q) + g(\bar{C}(t, q)) \bar{T}_q(t, q)] \rangle dq$$

The first term is eliminated because the contour starts and ends at the same point. Hence, now we have

$$\begin{aligned}& - \int_0^1 \langle \bar{C}_i(t, q), \langle \nabla g(\bar{C}(t, q)), \bar{C}_q(t, q) \rangle \bar{T}(t, q) \rangle dq - \int_0^1 \langle \bar{C}_i(t, q), g(\bar{C}(t, q)) \bar{T}_q(t, q) \rangle dq \\ &= - \int_0^1 \langle \nabla g(\bar{C}(t, q)), \bar{C}_q(t, q) \rangle \langle \bar{C}_i(t, q), \bar{T}(t, q) \rangle dq - \int_0^1 g(\bar{C}(t, q)) \langle \bar{C}_i(t, q), \bar{T}_q(t, q) \rangle dq\end{aligned}$$

**(Eq. 2-10)**

Combining (Eq. 2-7) and (Eq. 2-10), (Eq. 2-5) becomes

$$\frac{d}{dt} L_R(\bar{C}(t)) = \int_0^1 \langle \nabla g(\bar{C}), \bar{C}_i \rangle |\bar{C}_q| - \langle \nabla g(\bar{C}), \bar{C}_q \rangle \langle \bar{C}_i, \bar{T} \rangle - g(\bar{C}) \langle \bar{C}_i, \bar{T}_q \rangle dq$$

**(Eq. 2-11)**

Furthermore, the parameter  $q$  is transformed to the arc-length  $s$ , which is defined as

$$s(q) = \int_0^q |\bar{C}_q(q')| dq', \text{ or } \frac{ds}{dq} = |\bar{C}_q|.$$

**(Eq. 2-12)**

On the other hand, recall that  $\bar{C} = [x \ y]$ . From (Eq. 2-12), we have

$$\bar{C}_q = \frac{d\bar{C}}{dq} = \frac{d\bar{C}}{ds} \frac{ds}{dq} = \bar{C}_s \cdot |\bar{C}_q|.$$

**(Eq. 2-13)**

Similarly,

$$\bar{T}_q = \bar{T}_s \cdot |\bar{C}_q|.$$

**(Eq. 2-14)**

Hence, (Eq. 2-11) can be derived as follows

$$\begin{aligned} \frac{d}{dt} L_R(\bar{C}(t)) &= \int_0^1 \left[ \langle \nabla g(\bar{C}), \bar{C}_t \rangle - \langle \nabla g(\bar{C}), \bar{C}_s \rangle \langle \bar{C}_t, \bar{T} \rangle - g(\bar{C}) \langle \bar{C}_t, \bar{T}_s \rangle \right] |\bar{C}_q| dq \\ &= \int_0^{L(\bar{C})} \left[ \langle \nabla g(\bar{C}) - \langle \nabla g(\bar{C}), \bar{C}_s \rangle \bar{T} - g(\bar{C}) \bar{T}_s, \bar{C}_t \rangle ds. \end{aligned}$$

Since  $\nabla g(\bar{C}) = \langle \nabla g(\bar{C}), \bar{T} \rangle \cdot \bar{T} + \langle \nabla g(\bar{C}), \bar{N} \rangle \cdot \bar{N}$  and  $\bar{C}_s = \frac{\bar{C}_q}{|\bar{C}_q|} = \bar{T}$ , we have

$$\frac{d}{dt} L_R(\bar{C}(t)) = \int_0^{L(\bar{C})} \left[ \langle \nabla g(\bar{C}), \bar{N} \rangle \bar{N} - g(\bar{C}) \bar{T}_s \right], \bar{C}_t \rangle ds. \quad (\text{Eq. 2-15})$$

In (Eq. 2-15), the variation of the tangent vector on the contour  $\bar{T}_s$  has a relation with the normal vector  $\bar{N}$  in terms of the curvature  $\kappa$ . What follows will prove the relation that  $\bar{T}_s = \kappa \bar{N}$ .

Any tangent vector on the contour has the x component and y component. If

$\bar{T} = \frac{1}{\sqrt{x_s^2 + y_s^2}} [x_s \ y_s]^T$ , then the tangential angle  $\phi$  is defined as

$$\phi = \tan^{-1}\left(\frac{y_s}{x_s}\right).$$

Then, the curvature  $\kappa$  is defined as the variation rate of  $\phi$  with respect to the arc-length  $s$ . That is,

$$\kappa = \frac{d\phi}{ds} = \frac{d \tan^{-1}(y_s / x_s)}{ds} = \frac{1}{1 + (y_s / x_s)^2} \frac{y_{ss} x_s - x_{ss} y_s}{x_s^2} = \frac{y_{ss} x_s - x_{ss} y_s}{x_s^2 + y_s^2}.$$

Because the unit normal vector is  $\bar{N} = \frac{[-y_s \ x_s]^T}{\sqrt{x_s^2 + y_s^2}}$ , we have

$$k \bar{N} = \left[ \frac{x_{ss} y_s^2 - x_s y_s y_{ss}}{(x_s^2 + y_s^2)^{3/2}} \quad \frac{y_{ss} x_s^2 - x_s y_s x_{ss}}{(x_s^2 + y_s^2)^{3/2}} \right]^T. \quad (\text{Eq. 2-16})$$

Now, consider the variation of the tangential vector with respect to arc-length  $s$ . Here, we have



$$\begin{aligned} \frac{d}{ds} \frac{x_s}{\sqrt{x_s^2 + y_s^2}} &= \frac{x_{ss} \sqrt{x_s^2 + y_s^2} - \frac{1}{2}(x_s^2 + y_s^2)^{-1/2} (2x_s x_{ss} + 2y_s y_{ss}) x_s}{x_s^2 + y_s^2} \\ &= \frac{x_{ss}(x_s^2 + y_s^2) - x_s(x_s x_{ss} + y_s y_{ss})}{(x_s^2 + y_s^2)^{3/2}} = \frac{x_{ss} y_s^2 - x_s y_s y_{ss}}{(x_s^2 + y_s^2)^{3/2}} \quad \text{and} \\ \frac{d}{ds} \frac{y_s}{\sqrt{x_s^2 + y_s^2}} &= \frac{y_{ss}(x_s^2 + y_s^2) - y_s(x_s x_{ss} + y_s y_{ss})}{(x_s^2 + y_s^2)^{3/2}} = \frac{y_{ss} x_s^2 - x_s y_s x_{ss}}{(x_s^2 + y_s^2)^{3/2}}. \end{aligned}$$

Hence, we have

$$\vec{T}_s = \begin{bmatrix} \frac{x_{ss} y_s^2 - x_s y_s y_{ss}}{(x_s^2 + y_s^2)^{3/2}} & \frac{y_{ss} x_s^2 - x_s y_s x_{ss}}{(x_s^2 + y_s^2)^{3/2}} \end{bmatrix}^T \quad (\text{Eq. 2-17})$$

From (Eq. 2-16) and (Eq. 2-17), we have  $\vec{T}_s = \kappa \vec{N}$ .

If  $\vec{T}_s$  in (Eq. 2-15) is replaced by  $\kappa \vec{N}$ , we have

$$\frac{d}{dt} L_R(\bar{C}(t)) = \int_0^{L(\bar{C})} \langle [\langle \nabla g(\bar{C}), \vec{N} \rangle \vec{N} - g(\bar{C}) k \vec{N}], \bar{C}_t \rangle ds.$$

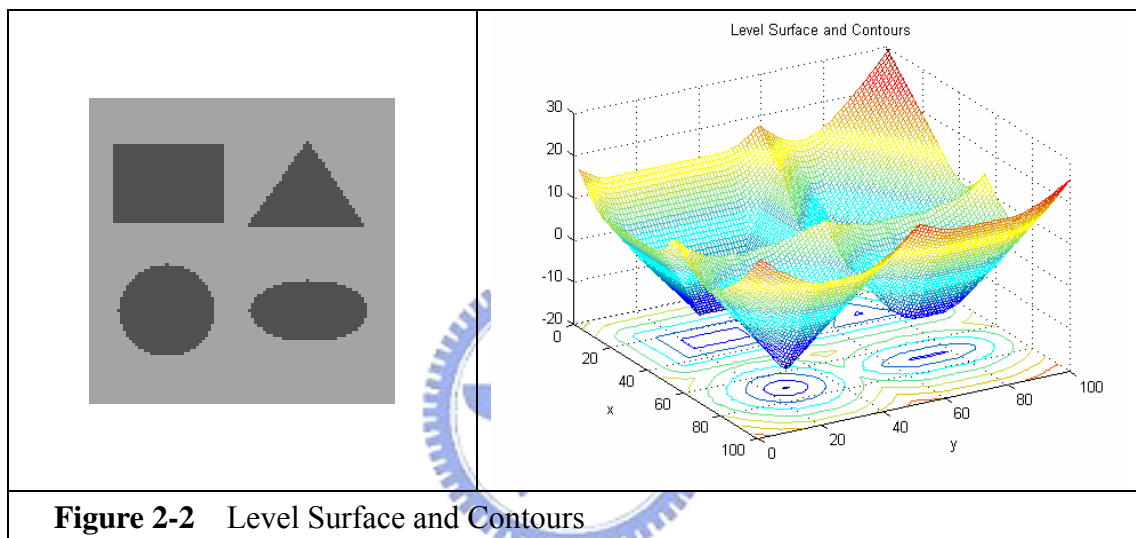
According to the steepest-descent method, the best way to propagate the  $\bar{C}$  with respect to time  $t$  is

$$\bar{C}_t = (g(\bar{C}) \kappa - \langle \nabla g(\bar{C}), \vec{N} \rangle) \vec{N}. \quad (\text{Eq. 2-18})$$

## 2.3 Level Set Theory

The Geodesic Active Contour model in the previous section describes how the contour propagates along its normal vector. The propagation speed is proportional to a scalar function, which is dominated by  $\langle \nabla g(\bar{C}), \vec{N} \rangle$  and the curvature  $\kappa$ . This model can be transformed into an associated level set function. Multiple contours can then be split or merged by updating the level surface.

An example of level surface is shown in Figure 2-2. The right image is the level surface corresponding to the left image. The function of the level surface is called a level set function. The sign of the level set function is negative inside the object, while positive outside the object. The level curve with zero value corresponds to the boundaries in the left image. The magnitude of the level set function depends on the nearest distance from the contour. The positions near the contour have smaller magnitudes, while those far away from the contour have larger magnitudes. The contour propagation problem is then transformed into the updating of the level set function. All properties, like the normal vector and curvature, can be directly computed from the level set function.



**Figure 2-2** Level Surface and Contours

Consider a contour  $\vec{C}(t) = [x(t) \ y(t)]^T$ . A level set function which depends on the contour  $\vec{C}(t)$  and the time  $t$  is

$$\varphi(\vec{C}(t), t) : \{R^2 \times [0, T]\} \rightarrow R.$$

The position of the contour is the zero level set of the level set function. Hence,

$$\varphi(\vec{C}(t), t) = 0 \quad \text{and thus} \quad \frac{\partial \varphi}{\partial x} dx + \frac{\partial \varphi}{\partial y} dy + \frac{\partial \varphi}{\partial t} dt = 0.$$

This equation can be rewritten as

$$\left\langle \nabla \varphi, \frac{\partial \vec{C}}{\partial t} \right\rangle + \frac{\partial \varphi}{\partial t} = 0.$$

If  $\frac{\partial \vec{C}}{\partial t} = F(\kappa) \vec{N}$ , where  $F(\kappa)$  is a speed function of the curvature  $\kappa$ , the above equation becomes

$$\frac{\partial \varphi}{\partial t} = -\langle \nabla \varphi, F(\kappa) \bar{N} \rangle \quad (\text{Eq. 2-19})$$

Recall (Eq. 2-18),  $F(\kappa) = g(\bar{C}) \kappa - \langle \nabla g(\bar{C}), \bar{N} \rangle$ .

Because the contour is the level curve of  $\varphi(\bar{C}(t), t)$  where  $\varphi(\bar{C}(t), t) = 0$ , the relative change of  $\varphi(\bar{C}(t), t)$  along the contour  $\bar{C}$  is zero. If we differentiate  $\varphi(\bar{C}(t), t)$  with respect to the arc-length  $s$ , we have

$$\varphi_s = 0 \quad \text{and} \quad \varphi_s = \varphi_x x_s + \varphi_y y_s = \langle \nabla \varphi, \bar{C}_s \rangle = 0.$$

This means that the vector  $\nabla \varphi$  is orthogonal to the tangential vector  $\bar{C}_s$ . Hence, the normal vector  $\bar{N}$  can be defined as  $\bar{N} = -\frac{\nabla \varphi}{|\nabla \varphi|}$ . Applying it to the (Eq. 2-19), we have

$$\frac{\partial \varphi}{\partial t} = \left\langle \nabla \varphi, F(\kappa) \frac{\nabla \varphi}{|\nabla \varphi|} \right\rangle = F(\kappa) |\nabla \varphi|. \quad (\text{Eq. 2-20})$$

This equation builds the relation between curve propagation and the level set function.

The following derivation will show that the curvature  $\kappa$  can be estimated from the level set function  $\varphi(\bar{C}(t), t)$ . Considering the second derivative of  $\varphi(\bar{C}(t), t)$  with respect to  $s$ , we have

$$\begin{aligned} \frac{\partial^2 \varphi}{\partial s^2} &= \frac{\partial}{\partial s} (\varphi_x x_s + \varphi_y y_s) = \frac{\partial \varphi_x}{\partial s} x_s + \varphi_x x_{ss} + \frac{\partial \varphi_y}{\partial s} y_s + \varphi_y y_{ss} \\ &= (\varphi_{xx} x_s + \varphi_{xy} y_s) x_s + (\varphi_{yx} x_s + \varphi_{yy} y_s) y_s + \varphi_x x_{ss} + \varphi_y y_{ss} \\ &= \varphi_{xx} x_s^2 + 2\varphi_{xy} x_s y_s + \varphi_{yy} y_s^2 + \varphi_x x_{ss} + \varphi_y y_{ss} = 0. \end{aligned} \quad (\text{Eq. 2-21})$$

On the other hand, we have the fact that

$$\bar{N} = \frac{[-y_s \ x_s]}{\sqrt{x_s^2 + y_s^2}} = -\frac{\nabla \varphi}{|\nabla \varphi|} = \left[ \frac{-\varphi_x}{\sqrt{\varphi_x^2 + \varphi_y^2}} \quad \frac{-\varphi_y}{\sqrt{\varphi_x^2 + \varphi_y^2}} \right].$$

Assume  $r = \frac{\sqrt{\varphi_x^2 + \varphi_y^2}}{\sqrt{x_s^2 + y_s^2}}$ . Then we have

$$\varphi_x = r y_s, \quad \varphi_y = -r x_s.$$

Recall that the curvature  $\kappa$  is expressed as

$$\kappa = \frac{y_{ss}x_s - x_{ss}y_s}{x_s^2 + y_s^2}.$$

From the above two equations, we have

$$\kappa(x_s^2 + y_s^2) = (y_{ss}x_s - x_{ss}y_s) = -\frac{1}{r}(\varphi_y y_{ss} + \varphi_x x_{ss}).$$

Hence,  $\varphi_x x_{ss} + \varphi_y y_{ss} = -\kappa r(x_s^2 + y_s^2)$ .

Using the above equation, (Eq. 2-21) becomes:

$$\kappa = \frac{\varphi_{xx}x_s^2 + 2\varphi_{xy}x_s y_s + \varphi_{yy}y_s^2}{r(x_s^2 + y_s^2)} = \frac{\varphi_{xx}\varphi_y^2 - 2\varphi_{xy}\varphi_x\varphi_y + \varphi_{yy}\varphi_x^2}{r(\varphi_x^2 + \varphi_y^2)}.$$

Because  $\sqrt{x_s^2 + y_s^2} = 1$ , we have  $r = \frac{\sqrt{\varphi_x^2 + \varphi_y^2}}{\sqrt{x_s^2 + y_s^2}} = \sqrt{\varphi_x^2 + \varphi_y^2}$  and

$$\kappa = \frac{\varphi_{xx}\varphi_y^2 - 2\varphi_{xy}\varphi_x\varphi_y + \varphi_{yy}\varphi_x^2}{(\varphi_x^2 + \varphi_y^2)^{3/2}} = \operatorname{div} \left( \frac{\nabla \varphi}{|\nabla \varphi|} \right), \quad (\text{Eq. 2-22})$$

where  $\operatorname{div}(\bar{A})$  denotes the divergence of the vector  $\bar{A}$ .

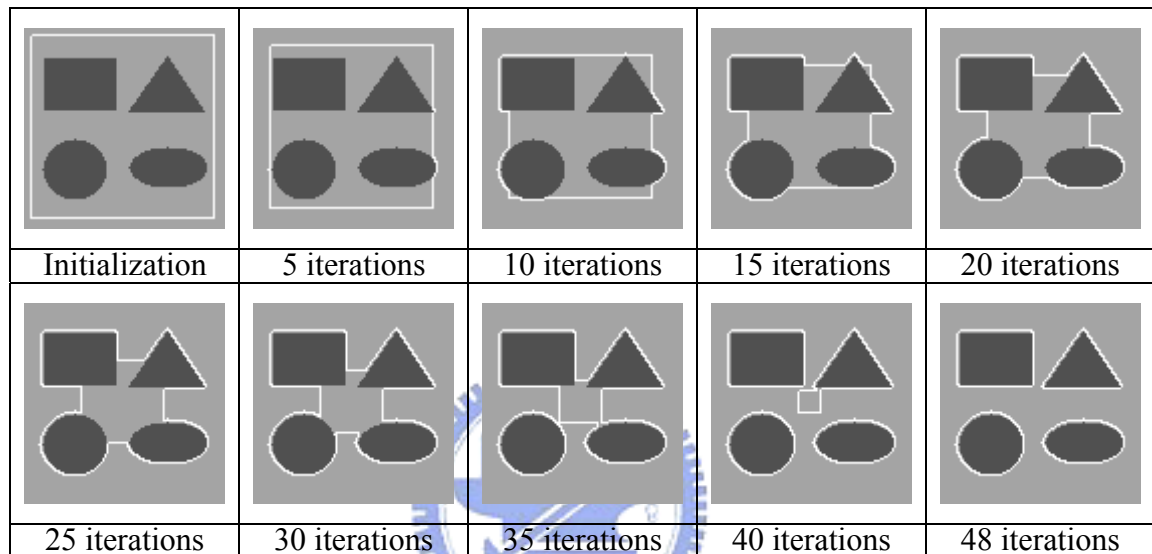
Now, the updating equation of the level set function in (Eq. 2-20) becomes

$$\begin{aligned} \frac{\partial \varphi}{\partial t} &= \left\langle \nabla \varphi, F(\kappa) \frac{\nabla \varphi}{|\nabla \varphi|} \right\rangle = F(\kappa) |\nabla \varphi| \\ &= \left( g(\bar{C}) \kappa - \langle \nabla g(\bar{C}), \bar{N} \rangle \right) |\nabla \varphi| = \left( g(\bar{C}) \kappa + \left\langle \nabla g(\bar{C}), \frac{\nabla \varphi}{|\nabla \varphi|} \right\rangle \right) |\nabla \varphi|. \end{aligned} \quad (\text{Eq. 2-23})$$

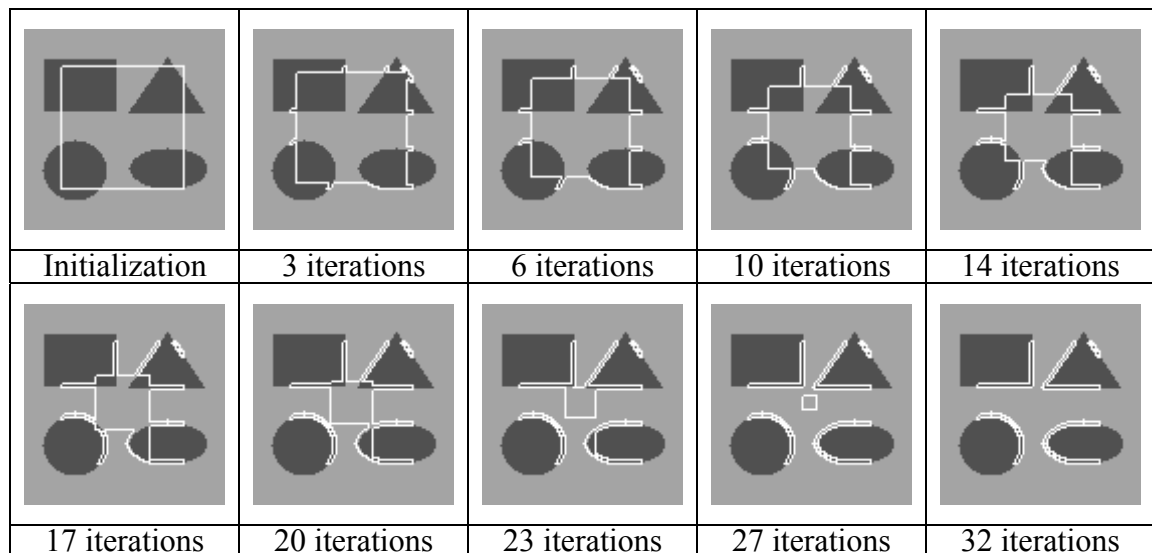
and the curvature  $\kappa$  is estimated by (Eq. 2-22).

## 2.4 Active Region Model

The geodesic active contour model introduced in the previous section produces proper contour information along the object boundary as shown in Figure 2-3. The initial curve must be totally outside the object; otherwise, an undesired result may be produced as shown in Figure 2-4.



**Figure 2-3** Geodesic Active Contour initialized totally outside the objects.



**Figure 2-4** Geodesic Active Contour with bad initialization.

The original energy equation (Eq. 2-2) only considers the properties along the contours. Minimization of this energy function makes contours move toward the boundary of the object. When the initial contour crosses the boundaries of objects, like the situation in Figure 2-4, the contours move toward the boundaries but the region information is lost. A more robust model is introduced in this section which takes the region information into account.

Consider a posteriori segmentation density function  $p_s(P(R)|I)$ , where  $P(R)$  is the partition status of region  $R$  and  $I$  is the input image. The density function is decomposed into

$$p_s(P(R)|I) = \frac{p(I|P(R))}{p(I)} p(P(R)) \quad (\text{Eq. 2-24})$$

based on the Bayes' Rule.

$p(P(R))$  is assumed to be equally possible so  $p(P(R))=1/Z$ , where  $Z$  is the number of partition regions.  $p(I)$  is constant and is ignored. The above equation becomes:

$$p_s(P(R)|I) \equiv p(I|P(R)) = p(I|\{R_A, R_B\}),$$

where  $R_A$  is the interior region (with the level function value  $\varphi < 0$ ) and  $R_B$  is the exterior region (with the level function value  $\varphi > 0$ ).

Assume the intensity distributions in  $R_A$  and  $R_B$  are independent. Then,

$$p_s(P(R)|I) = p([I|R_A] \cap [I|R_B]) = p(I|R_A)p(I|R_B).$$

Again, assume that intensity values of the pixels within each region also are independent of each other. Then, the maximum a posteriori problem is equivalent to maximizing the following equation:

$$\tilde{p}_s(P(R)|I) = \prod_{s \in R_A} p_A(I(s)) \prod_{s \in R_B} p_B(I(s)).$$

The energy function of the region part is modeled by using the  $[-\log(\ )]$  function of the above probability density function. That is,

$$E(P(R)) = -\iint_{R_A} \log[p_A(I(x, y))] dx dy - \iint_{R_B} \log[p_B(I(x, y))] dx dy. \quad (\text{Eq. 2-25})$$

In order to apply the Euler-Lagrange equation to the minimization of (Eq. 2-25), the double integration must be transformed to the form of contour line integration similar to (Eq. 2-4). The transformation in [6] applies Stoke's Theorem, which is

expressed as

$$\iint_R \nabla \times \bar{A} \cdot d\bar{s} = \oint_C \bar{A} \cdot d\bar{l}, \quad (\text{Eq. 2-26})$$

where  $(\nabla \times \bar{A}) \cdot \bar{n} = \lim_{S \rightarrow 0} \frac{\oint_{\text{small } C} \bar{A} \cdot d\bar{s}}{S}$  is the curl of  $\bar{A}$ .

The concept of the Stoke's Theorem is illustrated in Figure 2-5. The curl of  $\bar{A}$  is the integration of  $\bar{A} \cdot d\bar{s}$  in the small surface region of the small contours on the surface S. The integration of  $\nabla \times \bar{A}$  on the open surface S is equivalent to the integration of  $\bar{A}$  along the contour C. The integrations of  $\bar{A}$  on the small contours are cancelled with each other on the surface S.

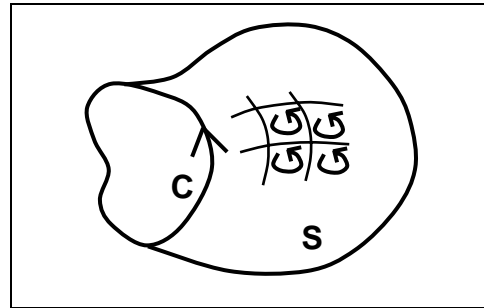


Figure 2-5 Stoke's Theorem

If the vector field  $\bar{A} = (P(x, y), Q(x, y))$ , then

$$\nabla \times \bar{A} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ P & Q & 0 \end{vmatrix} = \hat{k} \left( \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right).$$

According to (Eq. 2-26),

$$\iint_R \left( \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dx dy = \oint_C [P \ Q] \begin{bmatrix} dx \\ dy \end{bmatrix} = \int_0^L (P(x, y)x_s + Q(x, y)y_s) ds.$$

If properly choosing  $P$  and  $Q$ , a scalar function  $f$  is decomposed as

$$f(x, y) = \left( \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right).$$

For example, if we define

$$Q(x, y) = \frac{1}{2} \int_0^x f(t, y) dt \quad \text{and} \quad P(x, y) = -\frac{1}{2} \int_0^y f(x, t) dt,$$

then

$$E = \iint_R f(x, y) dx dy = \int_0^L (P(x, y)x_s + Q(x, y)y_s) ds \quad (\text{Eq. 2-27})$$

and

$$\frac{dE}{dt} = \int_0^L \left( \frac{dP}{dt} x_s + P \frac{dx_s}{dt} + \frac{dQ}{dt} y_s + Q \frac{dy_s}{dt} \right) ds = \int_0^L \left( \underbrace{\langle \bar{A}_t, \bar{C}_s \rangle}_{(1)} + \underbrace{\langle \bar{A}_s, \bar{C}_{st} \rangle}_{(2)} \right) ds .$$

**(Eq. 2-28)**

Because

$$(1) = \langle \bar{A}_x, \bar{C}_s \rangle \frac{dx}{dt} + \langle \bar{A}_y, \bar{C}_s \rangle \frac{dy}{dt} ,$$

if we use the integral by part formula and define  $\bar{u} = \bar{A}$ ,  $d\bar{v} = \bar{C}_{st} ds$ ,  $d\bar{u} = \bar{A}_s ds$ ,

and  $\bar{v} = \bar{C}_t$ , we have

$$\int_0^L (2) ds = \langle \bar{A}, \bar{C}_t \rangle_0^L - \int_0^L \langle \bar{C}_t, \bar{A}_s \rangle ds = - \int_0^L \langle \bar{C}_t, \bar{A}_s \rangle ds .$$

The first term is dropped since the contour starts and ends at the same point. Hence,

(Eq. 2-28) becomes

$$\frac{dE}{dt} = \int_0^L \left( \left\langle \left[ \langle \bar{A}_x, \bar{C}_s \rangle \quad \langle \bar{A}_y, \bar{C}_s \rangle \right]^T, \bar{C}_t \right\rangle - \langle \bar{A}_s, \bar{C}_t \rangle \right) ds .$$

Based on the steepest descent method,

$$\bar{C}_t = \bar{A}_s - \left[ \langle \bar{A}_x, \bar{C}_s \rangle \quad \langle \bar{A}_y, \bar{C}_s \rangle \right]^T .$$

In the x direction,

$$\frac{d}{ds} P(x, y) - \left( \frac{\partial P}{\partial x} x_s + \frac{\partial Q}{\partial x} y_s \right) = \left( \frac{\partial P}{\partial x} x_s + \frac{\partial P}{\partial y} y_s \right) - \left( \frac{\partial P}{\partial x} x_s + \frac{\partial Q}{\partial x} y_s \right) = - \left( \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) y_s .$$

In the y direction,

$$\frac{d}{ds} Q(x, y) - \left( \frac{\partial P}{\partial y} x_s + \frac{\partial Q}{\partial y} y_s \right) = \left( \frac{\partial Q}{\partial x} x_s + \frac{\partial Q}{\partial y} y_s \right) - \left( \frac{\partial P}{\partial y} x_s + \frac{\partial Q}{\partial y} y_s \right) = \left( \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) x_s .$$

Recall that  $f(x, y) = \left( \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right)$  and because  $\bar{N} = [-y_s \quad x_s]$ , we have

$$\bar{C}_t = f(x, y) \bar{N} .$$

**(Eq. 2-29)**

The above derivation is summarized here. In order to minimize the double integration of the scalar function  $f(x, y)$ , the energy function is transformed to (Eq. 2-27) by the Stoke's theorem. Then, (Eq. 2-29) is derived based on the



Euler-Lagrange approach. Hence, the minimization of  $\iint_R f(x, y) dx dy$  is simply equivalent to propagate the curve in the normal direction with the magnitude  $f(x, y)$ .

Now the minimization of (Eq. 2-25) can be obtained from (Eq. 2-29). That is,

$$\frac{\partial C}{\partial t} = -\log(P_A(I)) \bar{N}_{RA} - \log(P_B(I)) \bar{N}_{RB}.$$

Recall that  $R_A$  is the interior region (level function value  $\varphi < 0$ ) and  $R_B$  is the exterior region (with the level function value  $\varphi > 0$ ). Because  $\bar{N}_{RA} = -\bar{N}_{RB}$ , we define  $\bar{N} = \bar{N}_{RA} = -\bar{N}_{RB}$ . Hence,

$$\frac{\partial C}{\partial t} = \log\left(\frac{P_B(I)}{P_A(I)}\right) \bar{N}. \quad (\text{Eq. 2-30})$$

Combining the “geodesic active contour” model in (Eq. 2-4) and the “active region” model in (Eq. 2-25), the energy function is modified as

$$E[\Gamma] = \underbrace{\alpha \int_0^1 g(|\nabla I(C(q))|) |C'(q)| dq}_{\text{Geodesic Active Contour}} + \underbrace{(1 - \alpha) \left[ -\iint_{R_A} \log[p_A(I(x, y))] dx dy - \iint_{R_B} \log[p_B(I(x, y))] dx dy \right]}_{\text{Active Region Model}}, \quad (\text{Eq. 2-31})$$

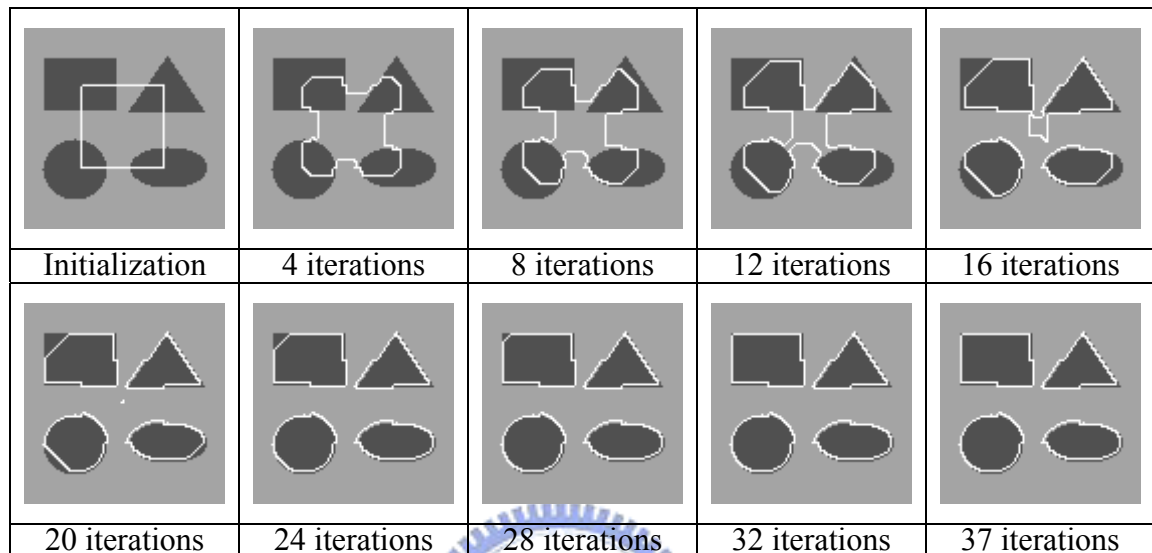
where  $\alpha$  is the weighting of the two models and  $\Gamma$  denotes the contour which produces the regions  $R_A$  and  $R_B$  and is described by the parameter  $q$ .

From the result of (Eq. 2-23) and (Eq. 2-30), minimizing  $E[\Gamma]$  is equivalent to updating the level surface using the following equation:

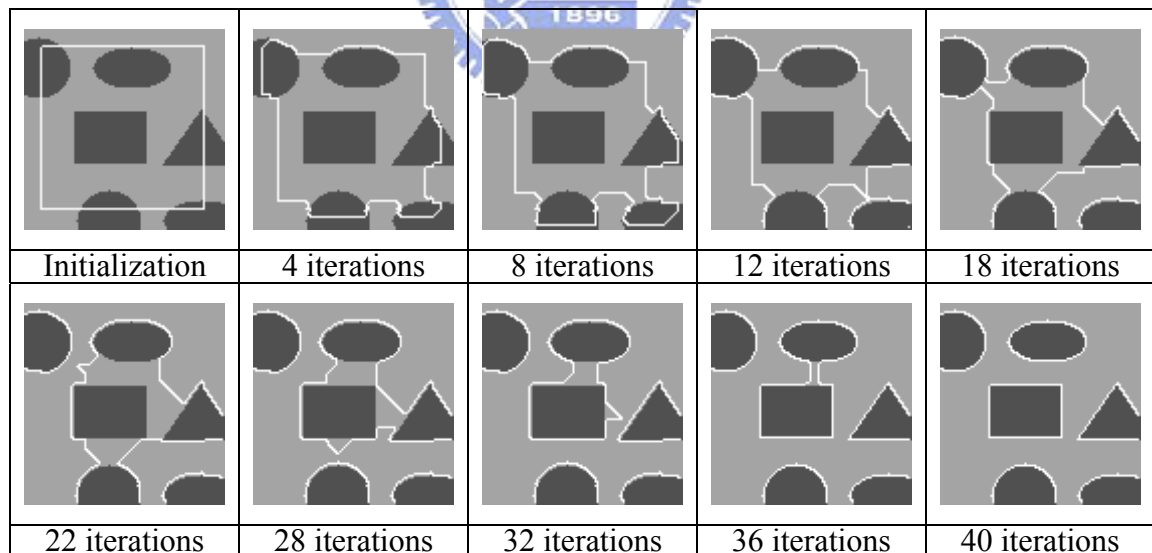
$$\frac{\partial \varphi}{\partial t} = \alpha (g(I(x, y)) \kappa |\nabla \varphi| + \langle \nabla g(I(x, y)), \nabla \varphi \rangle) + (1 - \alpha) \log\left(\frac{P_B(I)}{P_A(I)}\right) |\nabla \varphi|. \quad (\text{Eq. 2-32})$$

Figure 2-6 shows the evolution of the above equation with  $\alpha = 0.5$ . The contour moves both inward and outward and stops at the boundaries of the objects. When the objects are located outside the image, as shown in Figure 2-7, the new model can successfully extract the boundaries of the truncated objects.

Ideally, the whole motion object may locate within the image. However, sometimes parts of the objects are truncated by the image frame if the pan and tilt angles of the active camera is not adequate. In this imperfect situation, the new model is desirable for the extraction of the boundary.



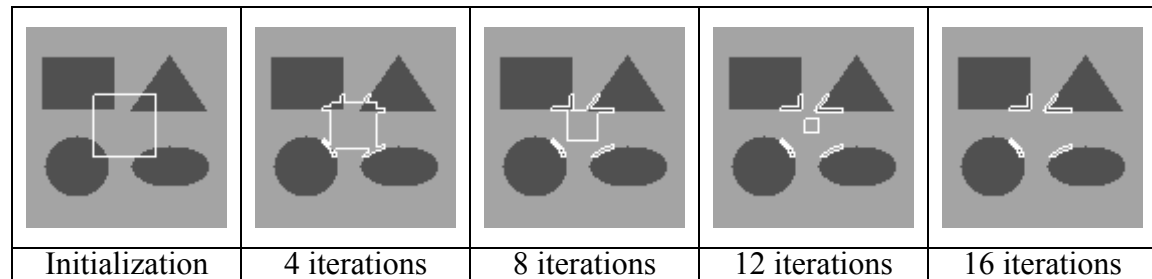
**Figure 2-6** Convergence of active contours with region information.



**Figure 2-7** Convergence of active contours with some objects are partly outside the image.

Although the new model performs well when the initialization curve does not completely encompass the objects, the initial curve must still contain enough

information of objects. In Figure 2-8, the initial curve is too small and it produces a bad result. Hence, the initial curve should be as large as possible under the constraint of computational time.



**Figure 2-8** Convergence of active contours when the initial curve is too small.





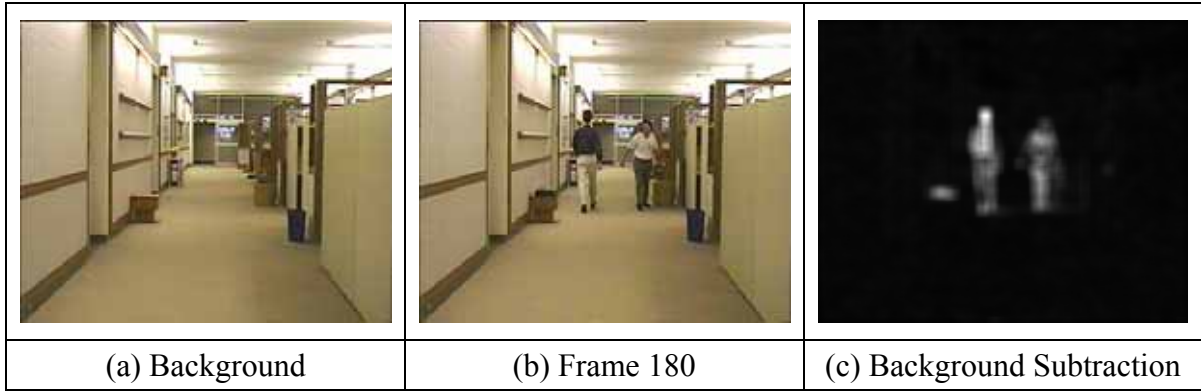
# Chapter 3. Detection and Tracking of Moving Objects

At the beginning of this chapter, the “active contours without edges” issue is first introduced. It is a model which statistically classifies the image data into two regions. Then the background subtraction result is analyzed based on the “active contours without edges” model. Under this approach, multiple moving regions can be successfully detected.

Region tracking is accomplished by estimating the morphing operation between two successive frames. The inter-frame image data are compensated pixel-wisely to approximate the morphing procedure. In the current frame, the pixels with smaller error difference with respect to the pixel inside the contours in the previous frame have a force to push the contour outward. A new region tracking model, which considers the level surface constructed in the previous frame, is proposed. Finally, the statistic property is taken into consideration to cope with the interference of a cluttered environment.

## 3.1 Motion Detection with Background

The motion region in the image can be obtained by background subtraction. Figure 3-1(c) shows the subtraction result of the background in (a) and the Frame 180 in (b). Obviously, the active contour model which needs the edge information will not work well in the difference image in Figure 3-1(c). For this case, the new “active contour model without edges” proposed in [5] may be able to properly classify the image data in Figure 3-1(c) and extract the motion regions in the image.



**Figure 3-1** Background Subtraction. (a) Background Image. (b) Frame 180.

(c) Background subtraction result.

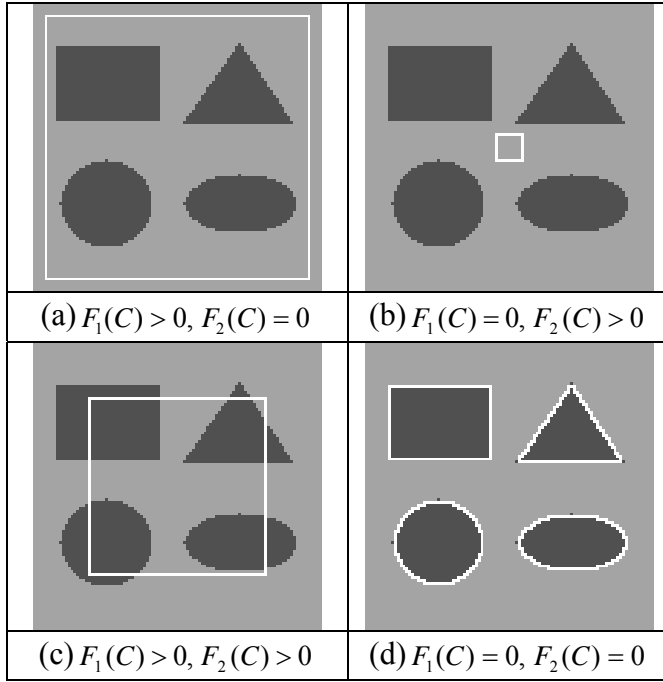
### 3.1.1. Active Contours without Edges

Define a curve  $C$  which divides the image into two regions. The energy function is defined as follows:

$$E = F_1(C) + F_2(C) = \int_{inside(C)} |u(x, y) - c_1|^2 dx dy + \int_{outside(C)} |u(x, y) - c_2|^2 dx dy, \quad (\text{Eq. 3-1})$$

where  $c_1 = \frac{\int_{inside(C)} u(x, y) dx dy}{\int_{inside(C)} dx dy}$ ,  $c_2 = \frac{\int_{outside(C)} u(x, y) dx dy}{\int_{outside(C)} dx dy}$ .

Here,  $u(x, y)$  is the image data at  $(x, y)$ , and  $c_1, c_2$  are the average intensity values inside and outside the contour  $C$ , respectively. The value of  $F_1(C)$  and  $F_2(C)$  have four different combinations which are illustrated in Figure 3-2. In Figure 3-2 (a), the values outside the contour are unique and thus  $F_2(C) = 0$ . The image data inside the contour contains two distinct values and thus  $F_1(C) > 0$ . The same analysis can be applied to Figure 3-2 (b). Now the image data outside the contour contains two values and thus  $F_2(C) > 0$ . If the contour shown in Figure 3-2 (c) passes across the objects, both  $F_1(C)$  and  $F_2(C)$  are non-zero. When the contours match the boundary of the objects,  $F_1(C)$  and  $F_2(C)$  become zero and the energy function reaches its minimum.



**Figure 3-2**  $F_1(C)$  and  $F_2(C)$ . Four situations of  $F_1(C)$  and  $F_2(C)$  with different contours. When the image data inside and outside the contours are not uniform, the values of  $F_1(C)$  or  $F_2(C)$  produce a positive value. Only when the contours matches the boundaries of the objects, both  $F_1(C)$  and  $F_2(C)$  are zero and the energy function in (Eq. 3-1) reaches its minimum.

The updating equation of the level set function is to be derived from (Eq. 3-1). First we define the Heaviside function  $H$ , which is expressed as

$$H(z) = \begin{cases} 1, & \text{if } z \leq 0 \\ 0, & \text{if } z > 0 \end{cases}.$$

If the length of the contours is taken into account, (Eq. 3-1) becomes

$$E = \int_{\Omega} |u(x, y) - c_1|^2 H(\varphi(x, y)) dx dy + \int_{\Omega} |u(x, y) - c_2|^2 (1 - H(\varphi(x, y))) dx dy + \int_{\Omega} |\nabla H(\varphi(x, y))| dx dy \quad (\text{Eq. 3-2})$$

The first two terms are the new “without edges” models and the last term represents the length of the contours. According to the geodesic contour model mentioned in Section 2.2, the contour propagates based on the following equation:

$$\bar{C}_t = \left( g(\bar{C}) \kappa - \langle \nabla g(\bar{C}), \bar{N} \rangle \right) \bar{N}.$$

In this case, the weighting  $g(\bar{C}) = 1$  and the above equation is simplified to be

$$\bar{C}_t = \kappa \bar{N}. \quad (\text{Eq. 3-3})$$

According to the active region model in Section 2.4, the first two terms in (Eq. 3-2) contribute the propagation force

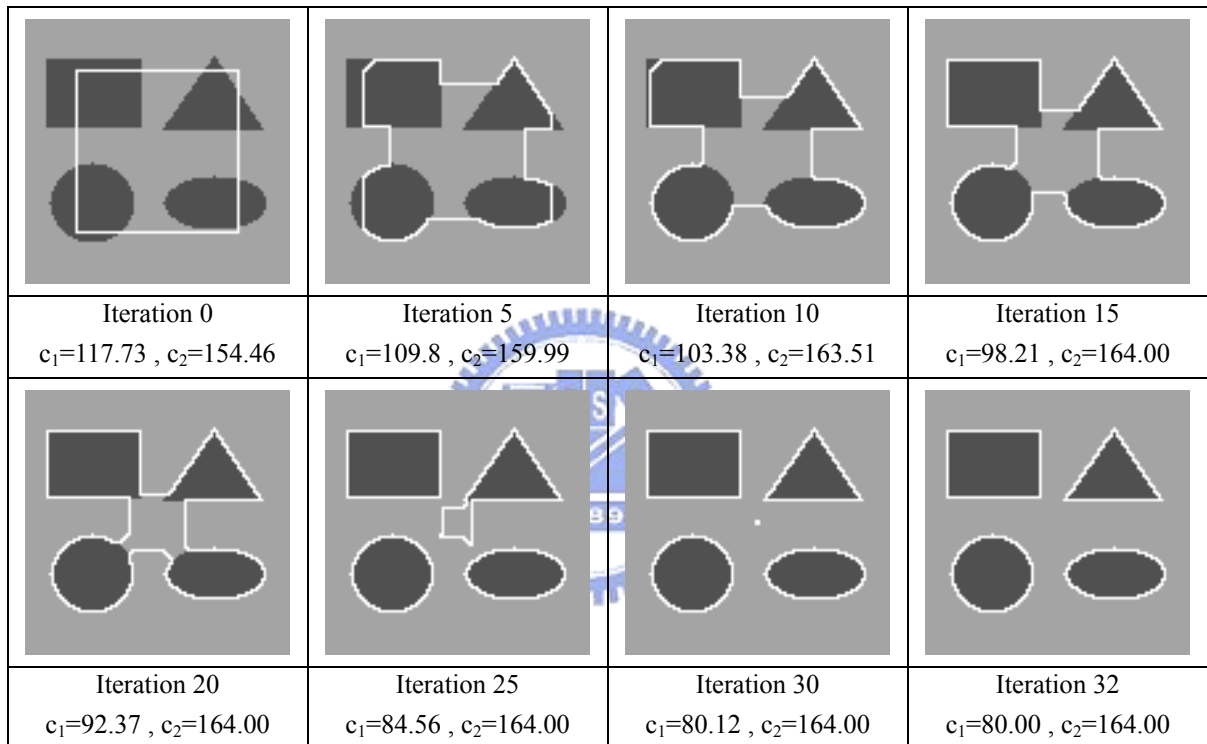
$$\bar{C}_t = |u(x, y) - c_1|^2 \bar{N}_{Inside} + |u(x, y) - c_2|^2 \bar{N}_{Outside}$$

$$= \left( |u(x, y) - c_1|^2 - |u(x, y) - c_2|^2 \right) \bar{N}. \quad (\text{Eq. 3-4})$$

Combining (Eq. 3-3) and (Eq. 3-4) and applying the level set theory, the updating equation becomes

$$\frac{d\phi}{dt} = \left[ |u(x, y) - c_1|^2 - |u(x, y) - c_2|^2 + \text{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) \right] |\nabla \phi|. \quad (\text{Eq. 3-5})$$

The result of active contours without edges is shown in Figure 3-3.

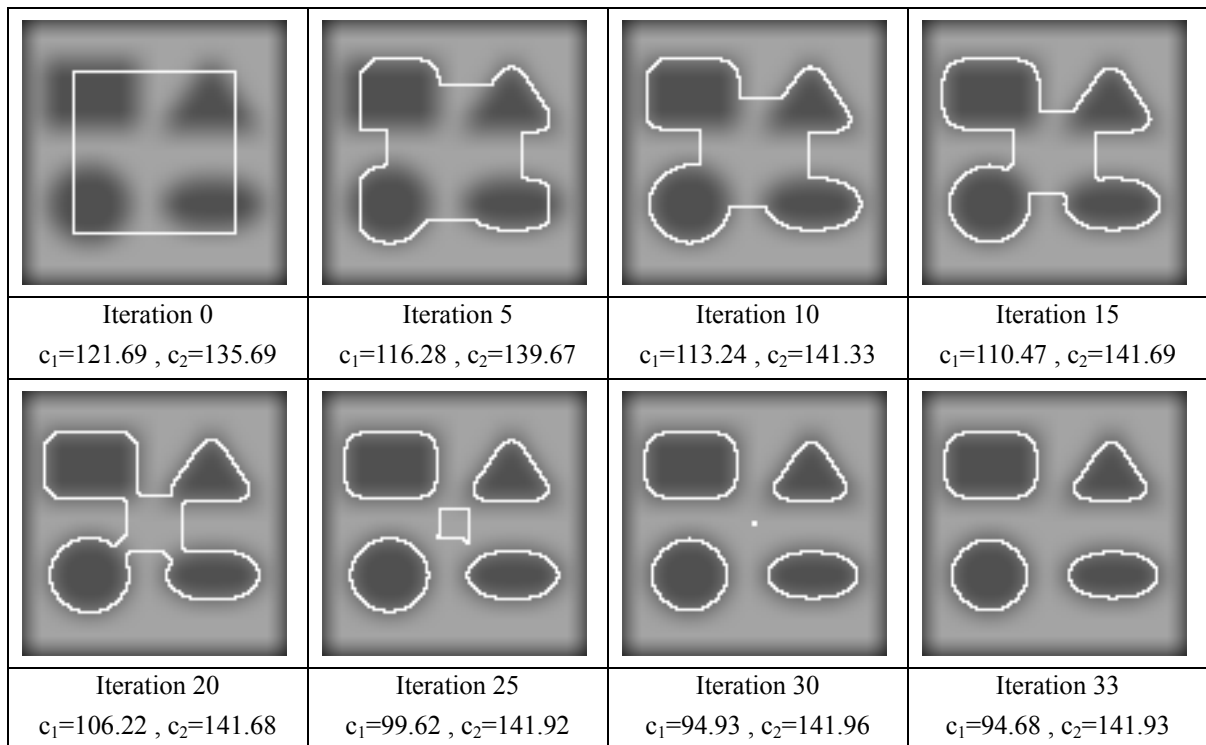


**Figure 3-3** Active contours without edges.  $c_1$  and  $c_2$  are the average values inside and outside the contours, respectively.

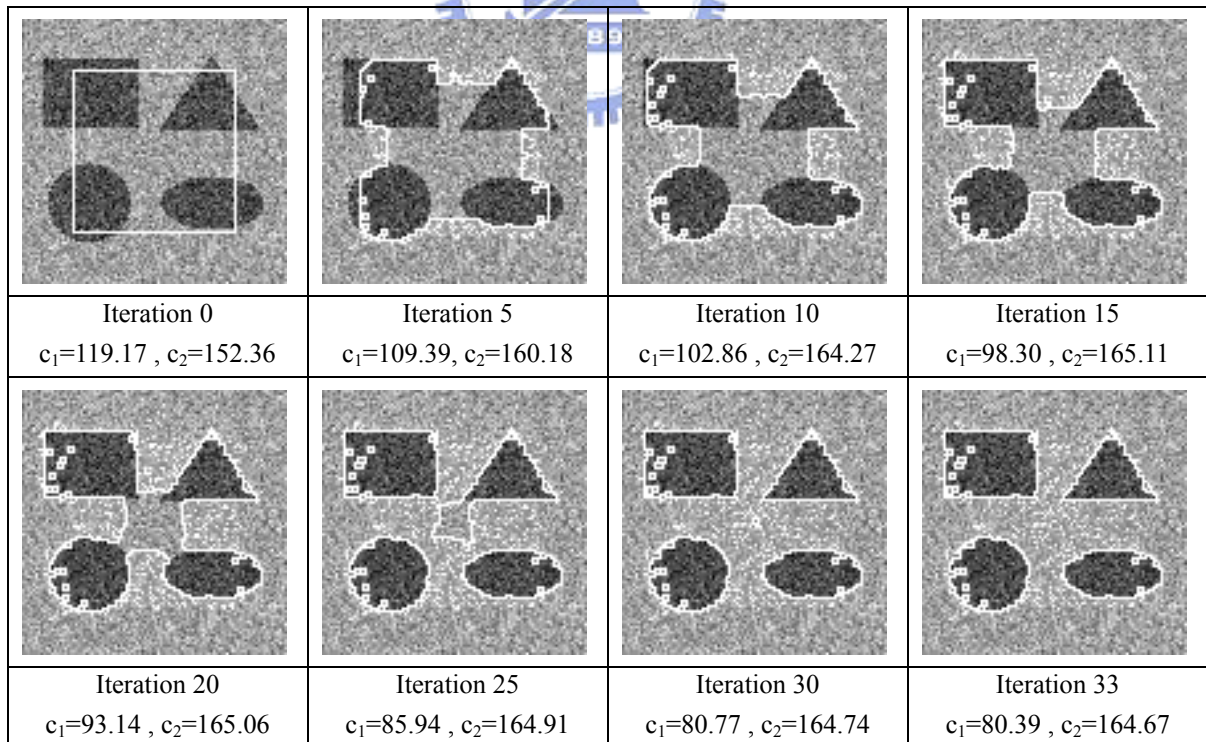
Because there are only two values in the image, the above simple case makes the energy function reach zero. When the image is blurred or added with some noise, the new model may still work well but the energy function reaches a non-zero minimum. Some non-trivial examples will be shown as follows.

Figure 3-4 shows the contour propagation in a blurred image. The blurred image is filtered by a Gaussian smoothing low-pass filter with the standard deviation 5.



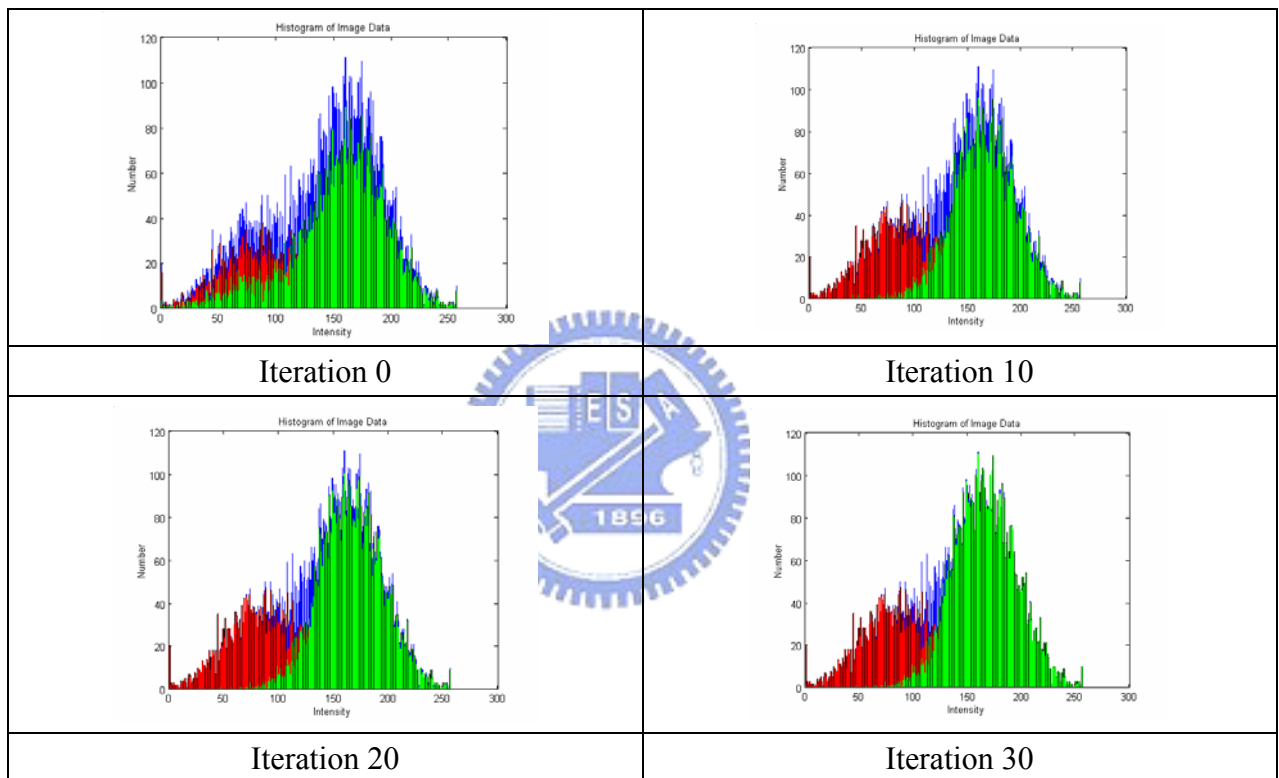


**Figure 3-4** Apply the “active contours without edges” method to the Gaussian smoothed image.  $c_1$  and  $c_2$  are the average values inside and outside the contours, respectively.



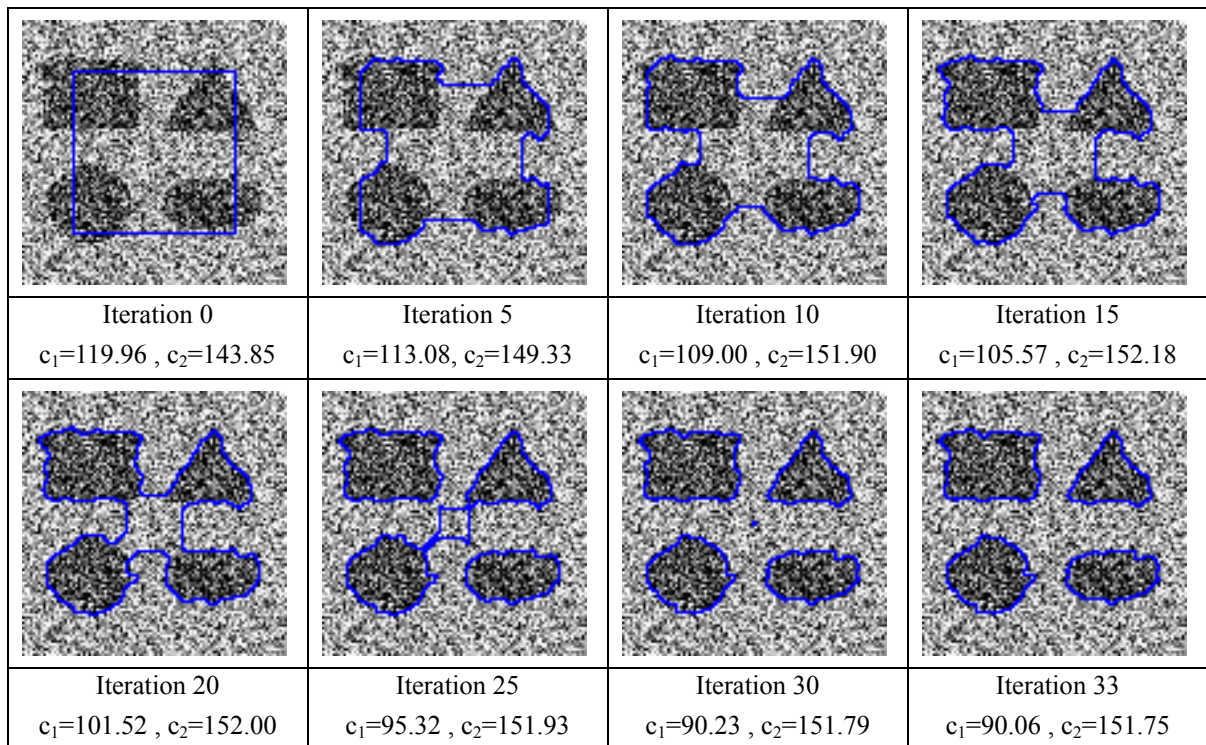
**Figure 3-5** Apply the “active contours without edges” approach to the image with Gaussian noise. The standard deviation of the noise is 30.

When this model is applied to a noisy image, some defects may appear as shown in Figure 3-5. Because the terms inside the integrals in (Eq. 3-1) are not continuous functions in the image spatial domain, some fragmented contours may be produced by the noise. The variation of the histogram inside and outside the contour is shown in Figure 3-6. It shows that the contours propagate toward the minimum of (Eq. 3-1) in the statistical sense. The blue bar is the histogram of the whole image, the red bar is the histogram of the inside region, and the green bar is the histogram of the outside region. The distributions of the inside and outside regions become more concentrated around the peak of the mean values.

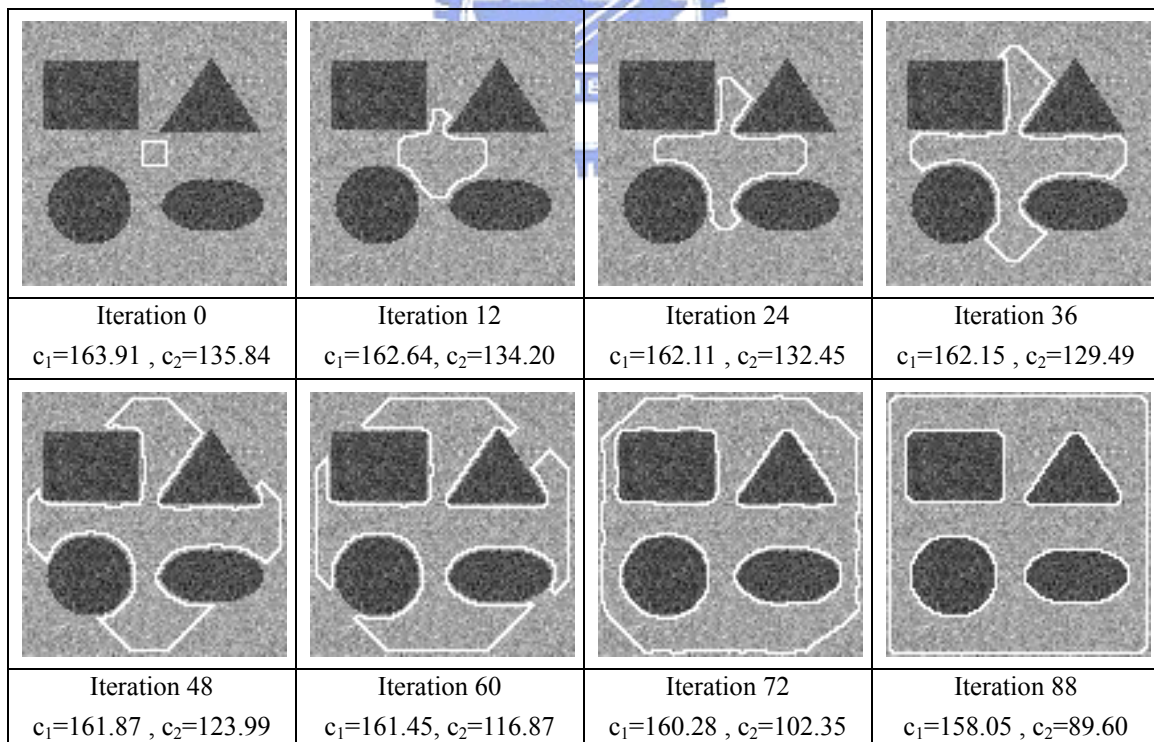


**Figure 3-6** Statistical histogram of Figure 3-5. The blue bar is the histogram of the whole image, the red bar is the histogram of the inside region, and the green bar is the histogram of the outside region.

To overcome the defects in Figure 3-5, the input image is pre-filtered by a Gaussian low-pass filter before the contour propagation starts. The result of the pre-filtered noise image is show in Figure 3-7. The standard deviation of the noise is now 80, which is much larger than the case in Figure 3-5.



**Figure 3-7** Apply the “active contours without edges” approach to the image with Gaussian noise. The standard deviation of the noise is 80.



**Figure 3-8** Apply the “active contours without edges” approach to the image with improper initialization. The standard deviation of the noise is 20. Note that  $c_1$  is larger than  $c_2$ . The region “inside” the contours is actually the background region.

Recall the active region model in Section 2.4. An undesirable result is produced if the initialization curve is too small to gather the information of the objects. Figure 3-8 shows the result based on the new model when the initial contour is very small. The new model successfully classifies the two regions in the image, but now the background region becomes the “inside” region while the objects are classified as the “outside” regions. Also note that the required iteration number increases because of the poor initialization.

### 3.1.2. Background Subtraction

Return to the case of Figure 3-1. The “active contours without edges” model is to be applied to the background subtracted image in Figure 3-1 (c). The difference values in the static regions are very small. Hence, the average value outside the contours approaches zero. The input images may contain some noise. Hence, a low-pass filtering process is applied to the difference image first.

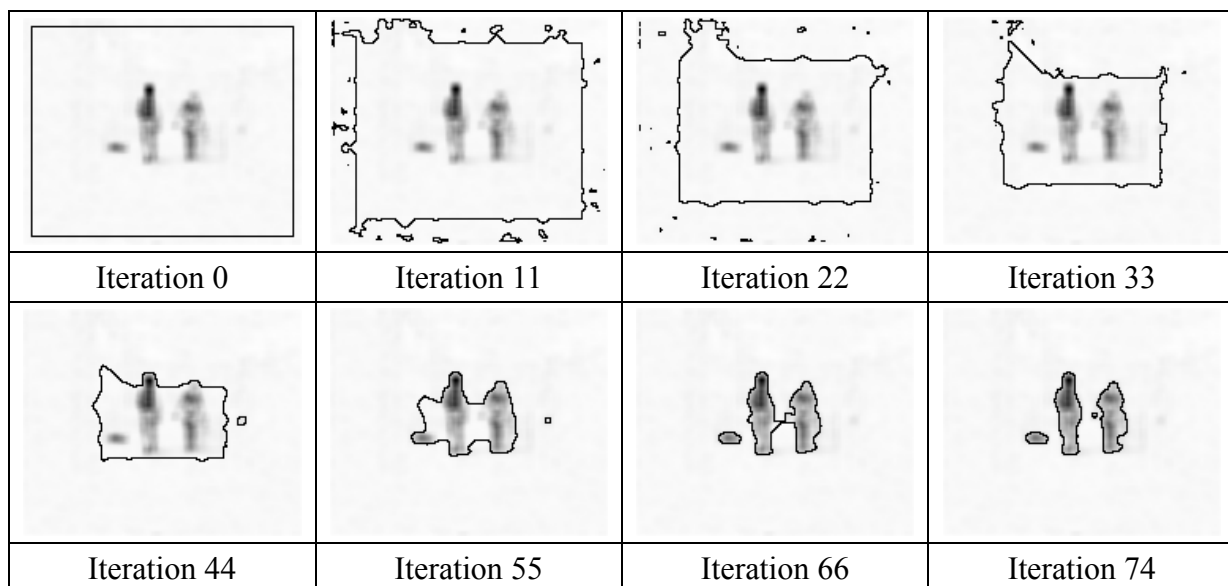
The evolution of level set theory is computationally expensive, and the computational time is proportional to the size of the level surface. Because very precise positions of the objects are not necessary in real applications, the image difference image can be processed block-wisely. Assume the block size is  $N \times N$ , the sum of absolute difference (*SAD*) of each block is obtained by

$$SAD_{ij}(t) = \sum_{p=1}^N \sum_{q=1}^N \left| u_t((i-1)N+p, (j-1)N+q) - B((i-1)N+p, (j-1)N+q) \right|,$$

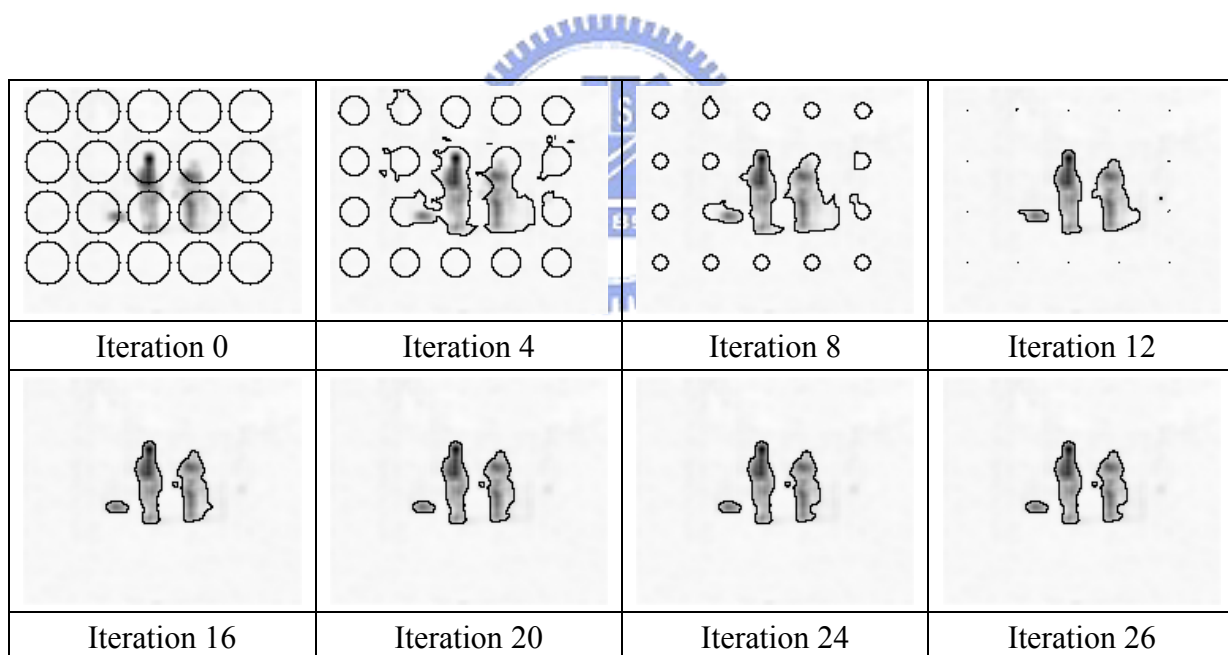
where  $B$  denotes the background image.

The “active contours without edges” model classifies the *SAD* data into two regions which maps to the motion and static parts of the image. The contour propagation is shown in Figure 3-9.

A large initial contour produces a large magnitude of the level set function in the center of the image. If the initial contours are split into small contours, the required iteration number decreases because the level set function may reach the convergence magnitudes more easily. Figure 3-10 shows the contour propagation with small contours initialization. The required iteration number decreases from 74 in Figure 3-9 to 26 in Figure 3-10.



**Figure 3-9** Classify the difference image data of Frame 180 and Frame 1 of the “Hall” test sequence.

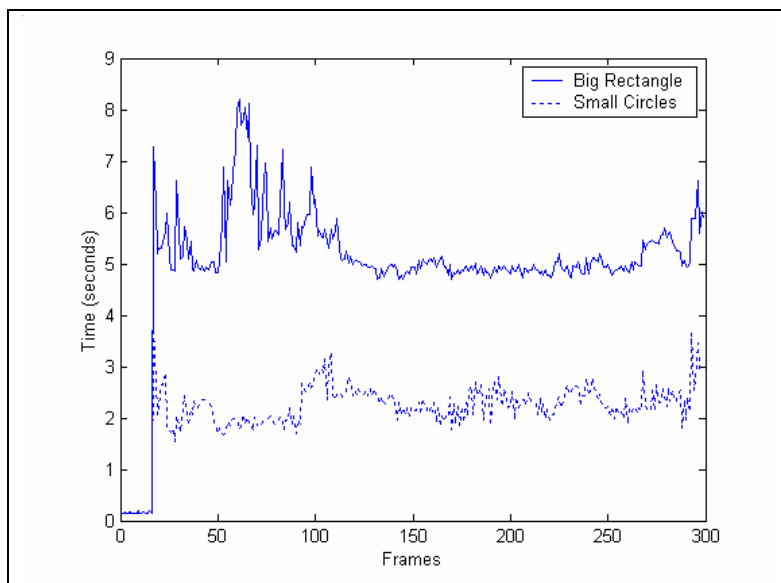


**Figure 3-10** Classify the difference image data of Frame 180 and Frame 1 of the “Hall” test sequence. The initial contour is replaced by a few small contours.

Figure 3-11 show the result which applies the “active contours without edges” model to the hall sequences. Frame 1 is used as the background. Figure 3-12 shows the comparison of the required time of two different initializations. There is no object in the first few frames. Hence, the level set evolution is not performed if the maximum SAD is less than a pre-selected threshold value.



**Figure 3-11** Apply the “active contours without edges” model to the hall sequences. Frame 1 is taken as the background. Multiple objects are successfully identified based on the level set approach.



**Figure 3-12** Comparison of required time of two different initializations. The solid line denotes the required time based on the large contour, while the dot line denotes the required time based on the set of small contours.

## 3.2 Region Tracking

When a camera is not static, both objects and background in the sequences change frame by frame. The background subtraction technique fails in this case because the static background is no longer available. Characteristics of the moving objects must be properly modeled in another way.

### 3.2.1. Region Tracking without Motion Computation

In [6] the regions are characterized by the probability models based on the Bayes' rule and MDL. Regions compete against each other and finally the image data within the same region will contain similar characteristics. Nikos Paragios and Rachid Deriche [7] implemented the region competition by using the level-set theory. Adbol-Reza Mansouri [8] further built another region tracking model via the use of the morphing concept so that the movement of the specific region can be properly tracked. Alper Yilmaz *et al* [9][10] applied the region tracking concept to the video surveillance sequences acquired by a mobile camera.

Assume the image domain is  $\Omega$  and a preset region  $R_0$  in the reference frame  $I^{n-1}$ . The goal is to find a region  $R_1$  which is very similar to  $R_0$  in the current frame  $I^n$  even if both translation and deformation of the object occur. The above statement is expressed as

$$\hat{R}_1 = \arg \max_{R \subset \Omega} P(R_1 = R | I^n, I^{n-1}, R_0). \quad (\text{Eq. 3-6})$$

This means that the desired region  $R_1$  corresponds to the maximum value of the well-defined probability model  $P$ . In order to define the probability model  $P$ , (Eq. 3-6) is rewritten as the following equation based on the Bayes' rule:

$$\hat{R}_1 = \arg \max_{R \subset \Omega} P(I^n | I^{n-1}, R_0, R_1 = R) P(R_1 = R | I^{n-1}, R_0). \quad (\text{Eq. 3-7})$$

The intensity probabilities inside and outside the region  $R$  are assumed to be independent. Then we define

$$P(I^n(\bar{x}) | I^{n-1}, R_0, R_1 = R) = \begin{cases} P_{in}(I^n(\bar{x}) | I^{n-1}, R_0), & \text{if } \bar{x} \in R \\ P_{out}(I^n(\bar{x}) | I^{n-1}, R_0), & \text{if } \bar{x} \in R^c \end{cases}$$

where  $R$  and  $R^c$  are the inside and outside regions in the reference frame  $I^{n-1}$ , respectively; and  $\bar{x}$  is the spatial position in the image. Assume again that the intensity values at different pixels are independent. (Eq. 3-7) then becomes

$$\hat{R}_1 = \arg \max_{R \subset \Omega} \left( \prod_{\bar{x} \in R} P_{in}(I^n(\bar{x}) | I^{n-1}, R_0) \right) \left( \prod_{\bar{x} \in R^c} P_{out}(I^n(\bar{x}) | I^{n-1}, R_0) \right) P(R_1 = R | I^{n-1}, R_0).$$

The negative log-likelihood function of  $P(R_1 = R | I^n, I^{n-1}, R_0)$  is

$$\begin{aligned} E(R_1 = R | I^n, I^{n-1}, R_0) = \\ - \int_R \log P_{in}(I^n(\bar{x}) | I^{n-1}, R_0) d\bar{x} - \int_{R^c} \log P_{out}(I^n(\bar{x}) | I^{n-1}, R_0) d\bar{x} - \log P(R_1 = R | I^{n-1}, R_0) \end{aligned} \quad (\text{Eq. 3-8})$$

and (Eq. 3-6) is now transformed to

$$\hat{R}_1 = \arg \min_{R \subset \Omega} E(R_1 = R | I^n, I^{n-1}, R_0).$$

This means that the optimal  $\hat{R}_1$  is obtained by minimizing  $E(R_1 = R | I^n, I^{n-1}, R_0)$ .

In the equation,  $P_{in}(I^n(\bar{x}) | I^{n-1}, R_0)$  is the probability of the intensity value inside the region  $R$  with the information of the reference frame  $I^{n-1}$  and the preset region  $R_0$ . If a morphing function  $\psi(I(\bar{x}))$  relates  $I^{n-1}$  and  $I^n$  as

$$I^n(\bar{x}) = I^{n-1}(\psi(\bar{x})) + \mu(\bar{x}), \quad (\text{Eq. 3-9})$$

where  $\mu(\bar{x})$  is the small error value after the morphing procedure. The probability model can be defined as

$$P_{in}(I^n(\bar{x}) | I^{n-1}, R_0) \equiv \sup_{\psi(\bar{x}) \in R_0} \exp\left(-\frac{(I^n(\bar{x}) - I^{n-1}(\psi(\bar{x})))^2}{2\sigma^2}\right), \quad (\text{Eq. 3-10})$$

where ‘‘sup’’ means the least upper bound of a set. Similarly,

$$P_{out}(I^n(\bar{x}) | I^{n-1}, R_0) \equiv \sup_{\psi(\bar{x}) \in R_0^c} \exp\left(-\frac{(I^n(\bar{x}) - I^{n-1}(\psi(\bar{x})))^2}{2\sigma^2}\right). \quad (\text{Eq. 3-11})$$



The probability model describes that the maximum probability occurs when the difference of the intensity values between the current frame and the morphing result of the reference (previous) frame is minimum. If considering color images, intensity difference can be replaced by color difference.

Now consider the third term of (Eq. 3-8). It can be defined as

$$-\log P(R_1 = R | I^{n-1}, R_0) = \lambda \oint_{\partial R} d\rho, \quad (\text{Eq. 3-12})$$

which describes the shape prior of the contours. The coefficient  $\lambda$  denotes the weight. The length of the contours tends to be as small as possible under the restriction of the region information defined in (Eq. 3-10) and (Eq. 3-11).

Use (Eq. 3-10), (Eq. 3-11), and (Eq. 3-12) to substitute the terms in (Eq. 3-8), the likelihood functional becomes

$$E(R_1 = R | I^n, I^{n-1}, R_0) = \int_R \inf_{\psi(\bar{x}) \in R_0} \left( \frac{(I^n(\bar{x}) - I^{n-1}(\psi(\bar{x})))^2}{2\sigma^2} \right) d\bar{x} + \int_{R^c} \inf_{\psi(\bar{x}) \in R_0^c} \left( \frac{(I^n(\bar{x}) - I^{n-1}(\psi(\bar{x})))^2}{2\sigma^2} \right) d\bar{x} + \lambda \oint_{\partial R} d\rho.$$

Note that the least upper bound “sup” is replaced by the greatest lower bound “inf” because of the negative log.

The morphing operator  $I(\psi(\bar{x}))$  can be simplified as “matching pixels” around a search range  $\delta$  in the reference frame. The constant  $\sigma$  can also be ignored. The above functional is now simplified as

$$E(R_1 = R | I^n, I^{n-1}, R_0) = \int_R \inf_{\|\bar{z}\| \leq \delta, \bar{x} + \bar{z} \in R_0} (I^n(\bar{x}) - I^{n-1}(\bar{x} + \bar{z}))^2 d\bar{x} + \int_{R^c} \inf_{\|\bar{z}\| \leq \delta, \bar{x} + \bar{z} \in R_0^c} (I^n(\bar{x}) - I^{n-1}(\bar{x} + \bar{z}))^2 d\bar{x} + \lambda \oint_{\partial R} d\rho.$$

Use the Euler-Lagrange derivation mentioned in previous sections, the likelihood functional is minimized by propagating the contour in the direction:

$$\frac{\partial \bar{C}}{\partial t} = \left( \inf_{\|\bar{z}\| \leq \delta, \bar{x} + \bar{z} \in R_0} (I^n(\bar{x}) - I^{n-1}(\bar{x} + \bar{z}))^2 - \inf_{\|\bar{z}\| \leq \delta, \bar{x} + \bar{z} \in R_0^c} (I^n(\bar{x}) - I^{n-1}(\bar{x} + \bar{z}))^2 + \lambda \kappa \right) \bar{N},$$

where  $\kappa$  is the curvature of the contour  $\bar{C}$ , and  $\bar{N}$  is the normal vector. Apply the level set theory described in Section 2.3, the corresponding updating equation of the

level surface is

$$\frac{\partial \varphi}{\partial t} = \left( \inf_{\|z\| \leq \delta, \bar{x} + \bar{z} \in R_0} \left( I^n(\bar{x}) - I^{n-1}(\bar{x} + \bar{z}) \right)^2 - \inf_{\|z\| \leq \delta, \bar{x} + \bar{z} \in R_0^c} \left( I^n(\bar{x}) - I^{n-1}(\bar{x} + \bar{z}) \right)^2 + \lambda \kappa \right) |\nabla \varphi| \quad (\text{Eq. 3-13})$$

(Eq. 3-13) means that the contour moves outward if the square error inside  $R_0$  is smaller than the square error outside  $R_0$ . If the pixel is far away from the boundary, one term in (Eq. 3-13) doesn't exist because of the limited search range. The inexistent term is set to be a very large value. Helpful implementation details can be found in [8].

Define the minimum square error found inside  $R_0$  as  $V_{in}$ :

$$V_{in} = \inf_{\|z\| \leq \delta, \bar{x} + \bar{z} \in R_0} \left( I^n(\bar{x}) - I^{n-1}(\bar{x} + \bar{z}) \right)^2 \quad (\text{Eq. 3-14})$$

and the minimum square error found outside  $R_0$  as  $V_{out}$

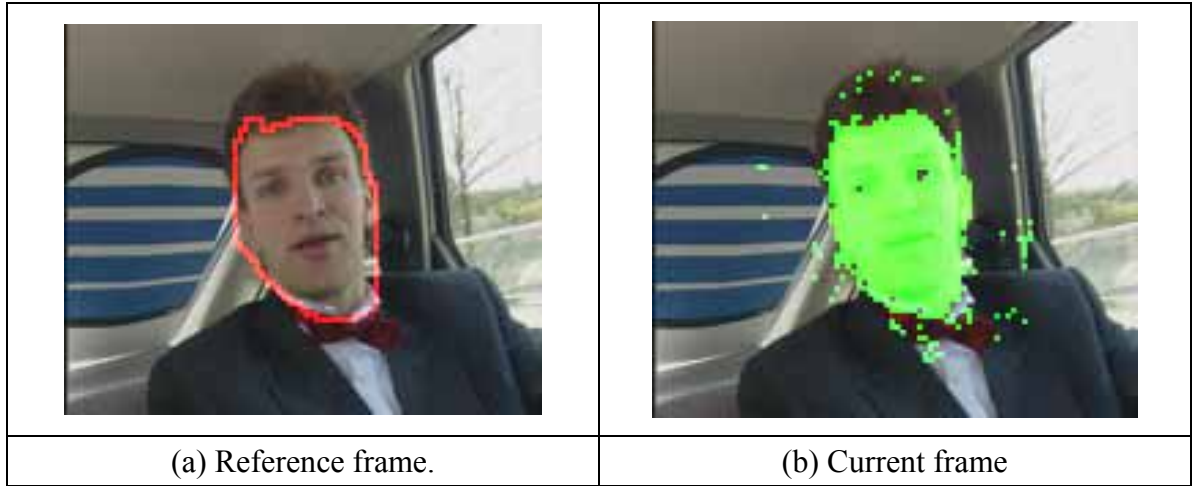
$$V_{out} = \inf_{\|z\| \leq \delta, \bar{x} + \bar{z} \in R_0^c} \left( I^n(\bar{x}) - I^{n-1}(\bar{x} + \bar{z}) \right)^2. \quad (\text{Eq. 3-15})$$

Replace the first two terms in (Eq. 3-13) by (Eq. 3-14) and (Eq. 3-15), the region tracking force  $F$  which update the level surface is

$$F = V_{in} - V_{out}. \quad (\text{Eq. 3-16})$$

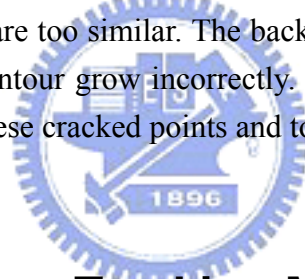
If  $F > 0$ , the level surface moves upward and the contour propagates inward. On the other hand,  $F < 0$  causes the contour to propagate outward.

Figure 3-13 contains two successive frames in the “car-phone” test sequences. The face region is pre-defined in the reference frame in Figure 3-13 (a). The pixels with  $F$  less than zero in the current frame are marked as green points in Figure 3-13 (b). The level surface moves down at the green points and moves up at the other locations. The level surface is updated by the force  $F$  until the final convergence is obtained. Note that even though the scene changes out of the car, the green region in the current frame may still properly estimate the face region in Figure 3-13 (a).



**Figure 3-13** The force  $F$  defined by the region tracking model. (a) The pre-defined region in the reference frame. (b) The pixels where  $F < 0$  in the current frame.

In Figure 3-13 (b), there are many cracked points outside the face region with  $F < 0$ . The false detection may cause problem if the intensities or the color values inside and outside the object are too similar. The background may be classified inside the contours and make the contour grow incorrectly. A new region tracking model is to be proposed to eliminate these cracked points and to increase the robustness.



### 3.2.2. A New Region Tracking Model

Because the frame rate is assumed to be very high, the inter-frame difference of the level surface should be very small. The new level surface can be roughly estimated by apply the morphing to the level surface of the reference frame. This procedure can be written as

$$\varphi^n(\bar{x}) = \varphi^{n-1}(\psi(\bar{x})) + \mu,$$

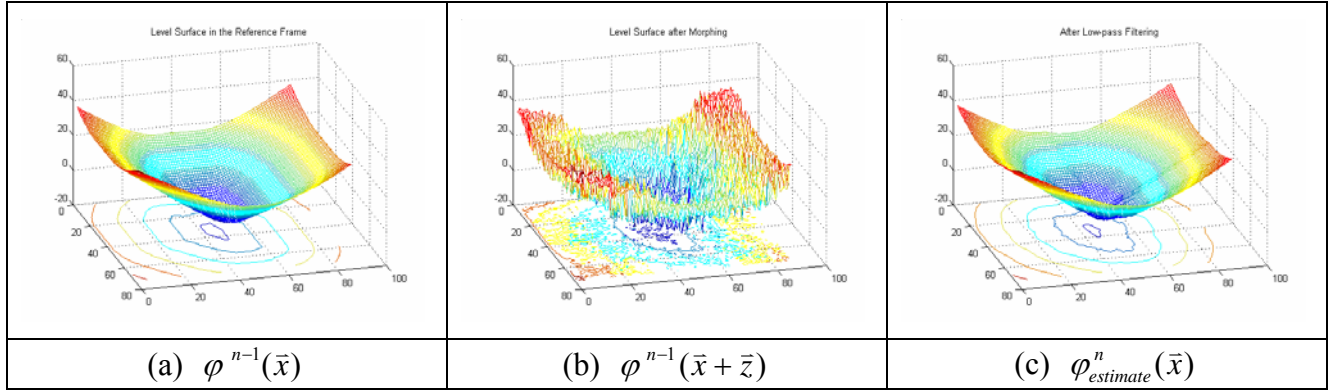
where  $\varphi$  is the level surface and  $\psi$  denotes the morphing operation. A pixel-wise matching can be used to simplify the morphing operation.  $\mu$  is the error value between the correct and estimated level surface. The morphing operation often results in cracked effects similar to Figure 3-13(b). The level surface produced by morphing is severely discontinuous as shown in Figure 3-14 (b). A simple low pass mean filter can alleviate this discontinuity and produce a smoother surface.

Besides the low-pass filtering, the re-initialization is used to make the surface

preserve the property  $|\nabla\varphi|=1$ . How to get the new surface is summarized in the following equation:

$$\varphi_{estimate}^n(\bar{x}) = \text{reinitialize} \left\{ \text{Lowpass Filtering} \left( \varphi^{n-1}(\bar{x} + \bar{z}) \right) \right\}. \quad (\text{Eq. 3-17})$$

The re-initialization step is introduced in Chapter 4.



**Figure 3-14** Apply the morphing operator to the level surface of the reference frame. (a) The level surface of the reference frame. (b) Morphing result. (c) Apply a low-pass filtering and the re-initialization to the surface in (b). (c) is used to estimate the new level surface in the current frame.

The difference between Figure 3-14 (b) and (c) provides the information about the reliability of the morphing operation. If the difference is very large, the morphing operation on the pixel is not very reliable because the inter-frame difference should be small. Now the probability models in (Eq. 3-10) and (Eq. 3-11) are modified as

$$P_{in}(I^n(\bar{x}) | I^{n-1}, R_0) \equiv \exp \left( -P_\varphi \cdot \inf_{\|\bar{z}\| \leq \delta, \bar{x} + \bar{z} \in R_0} \left( I^n(\bar{x}) - I^{n-1}(\bar{x} + \bar{z}) \right)^2 \right) \quad (\text{Eq. 3-18})$$

and

$$P_{out}(I^n(\bar{x}) | I^{n-1}, R_0) \equiv \exp \left( -P_\varphi \cdot \inf_{\|\bar{z}\| \leq \delta, \bar{x} + \bar{z} \in R_0^c} \left( I^n(\bar{x}) - I^{n-1}(\bar{x} + \bar{z}) \right)^2 \right), \quad (\text{Eq. 3-19})$$

where

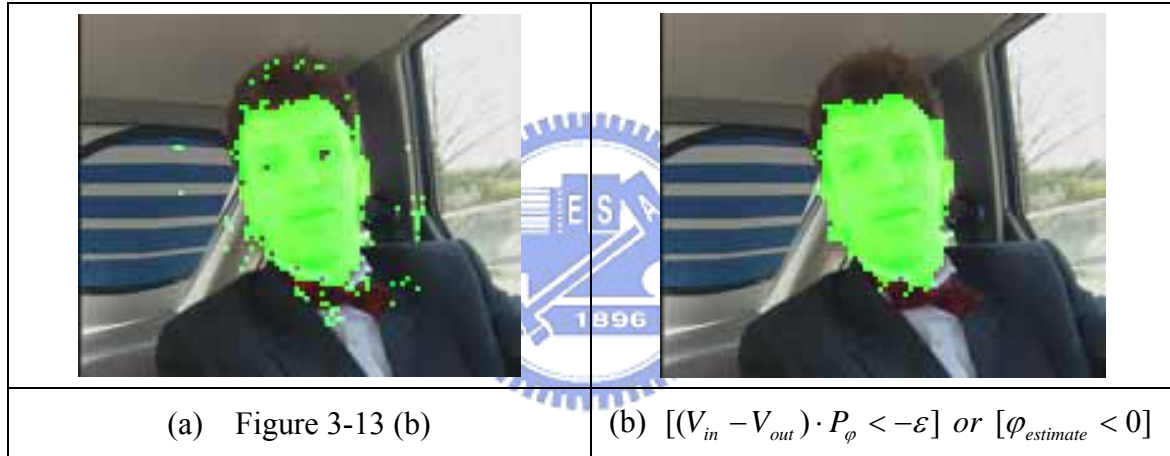
$$P_\varphi = \exp \left\{ - \frac{\left( \varphi_{estimate}^n(\bar{x}) - \varphi^{n-1}(\bar{x} + \bar{z}) \right)^2}{2\sigma^2} \right\}$$

is a weight which estimates the reliability of the morphing operation. Then the level surface updating equation in (Eq. 3-13) becomes

$$\frac{\partial \varphi}{\partial t} = \left[ \left( \inf_{\|\bar{z}\| \leq \delta, \bar{x} + \bar{z} \in R_0} (I^n(\bar{x}) - I^{n-1}(\bar{x} + \bar{z}))^2 - \inf_{\|\bar{z}\| \leq \delta, \bar{x} + \bar{z} \in R_0^c} (I^n(\bar{x}) - I^{n-1}(\bar{x} + \bar{z}))^2 \right) \cdot P_\varphi + \lambda \kappa \right] |\nabla \varphi|$$

$$= [(V_{in} - V_{out}) \cdot P_\varphi + \lambda \kappa] |\nabla \varphi|.$$

The additional weight  $P_\varphi$  suppresses the effect of  $(V_{in} - V_{out})$  if the square error  $(\varphi_{estimate}^n(\bar{x}) - \varphi^{n-1}(\bar{x} + \bar{z}))^2$  is large. Figure 3-15 (a) is the same image shown in Figure 3-14 (b). The green points in Figure 3-15 (b) are generated by the Boolean equation below the figure. The suppressed pixels are eliminated because of the negative threshold. The false detected points are obviously reduced in Figure 3-15 (b).

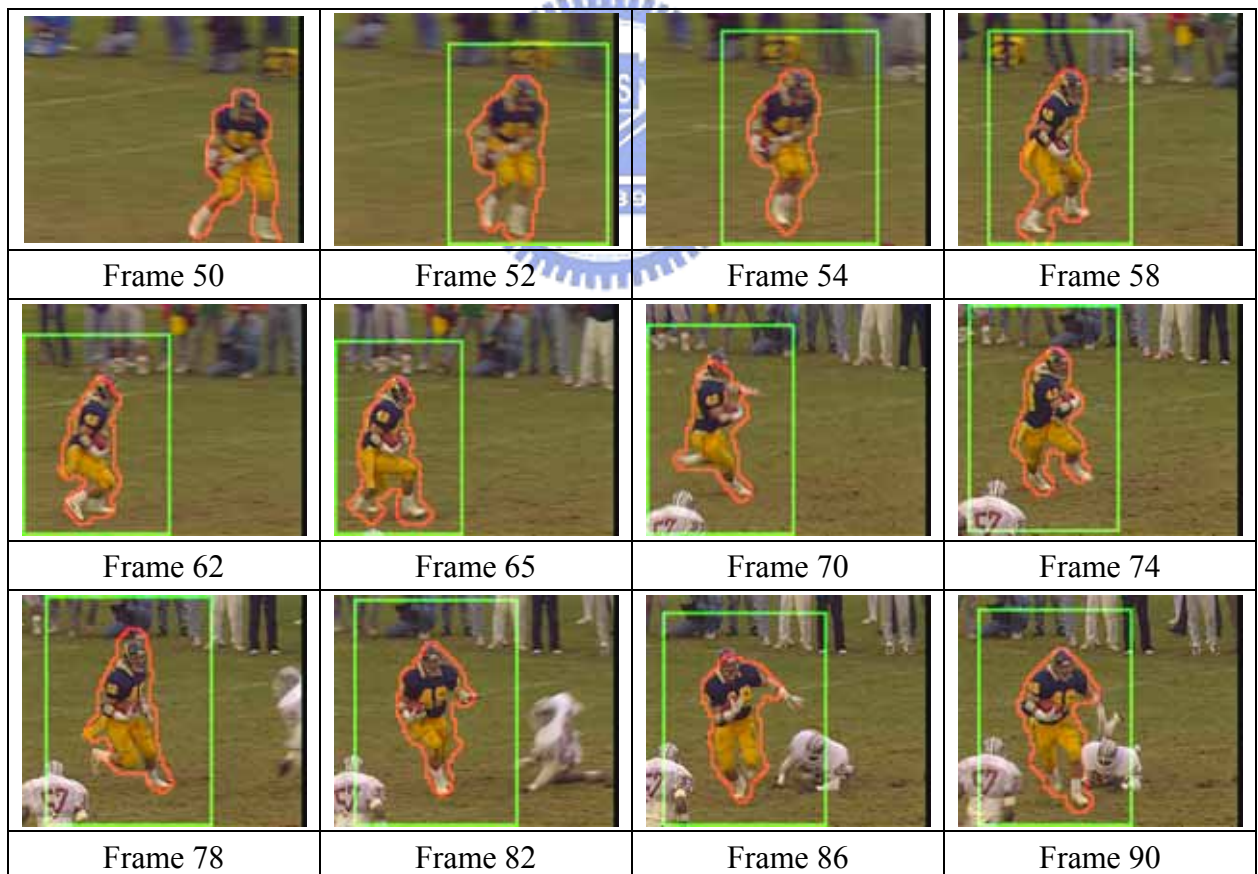


**Figure 3-15** The region force with and without the weight  $P_\varphi$ . The cracked false detected points are reduced by adding  $P_\varphi$ . The influence of the surrounding environment is reduced and makes the region tracking model more robust.

Figure 3-16 shows the region tracking results of the car-phone sequences. The level surface is only updated inside the green rectangle. Since the points far away from the object always find the points outside the  $R_0$  in the reference frame, the computation far away from the object is unnecessary. Because the level surface is only updated inside the rectangle, it results in discontinuous on the boundary of the rectangle. The discontinuity can be smoothed by the re-initialization at the end of the whole operation.

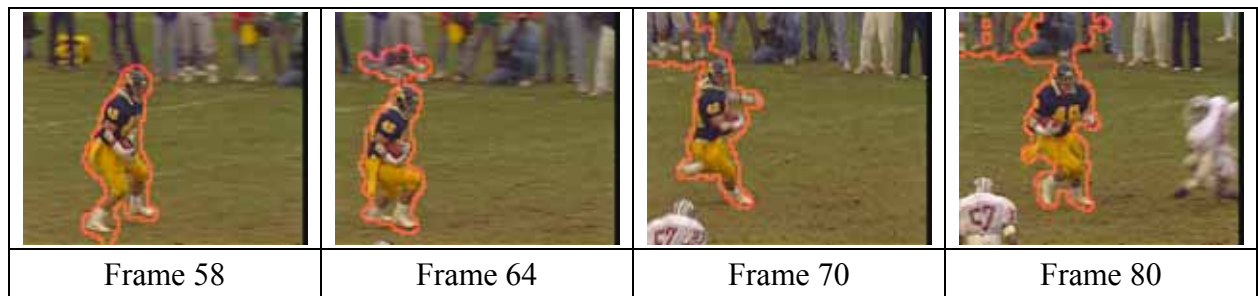


**Figure 3-16** The car-phone test sequences. The face region is defined in the first frame. The level surface is only updated inside the green rectangle. The face is successfully tracked even though the scene changes outside the car.

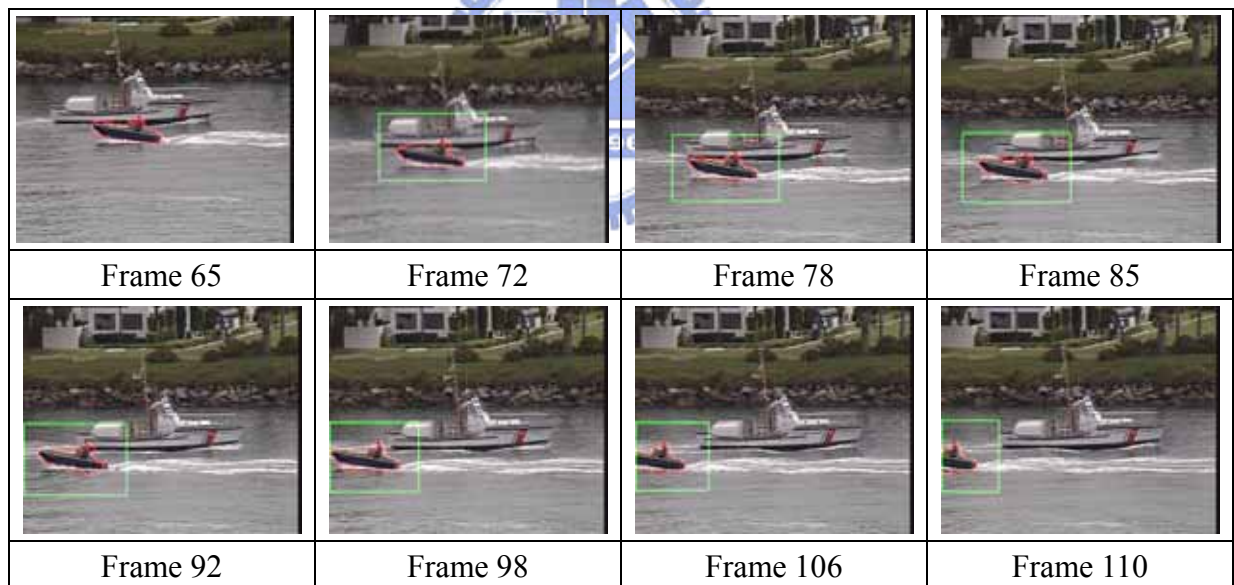


**Figure 3-17** Football sequences. The camera moves very quickly and the target is blurred in some frames. The contour does not grow incorrectly due to the use of  $\varphi_{estimate}^n(\bar{x})$  in (Eq. 3-17).

Figure 3-17 and Figure 3-18 shows the result of the football sequences. The sequences contain some interference of the clutter environment. Figure 3-17 is the result which applies the new model. The contour does not grow incorrectly because the new level surface is pre-estimated. The original tracking model produces the erroneous result shown as Figure 3-18.



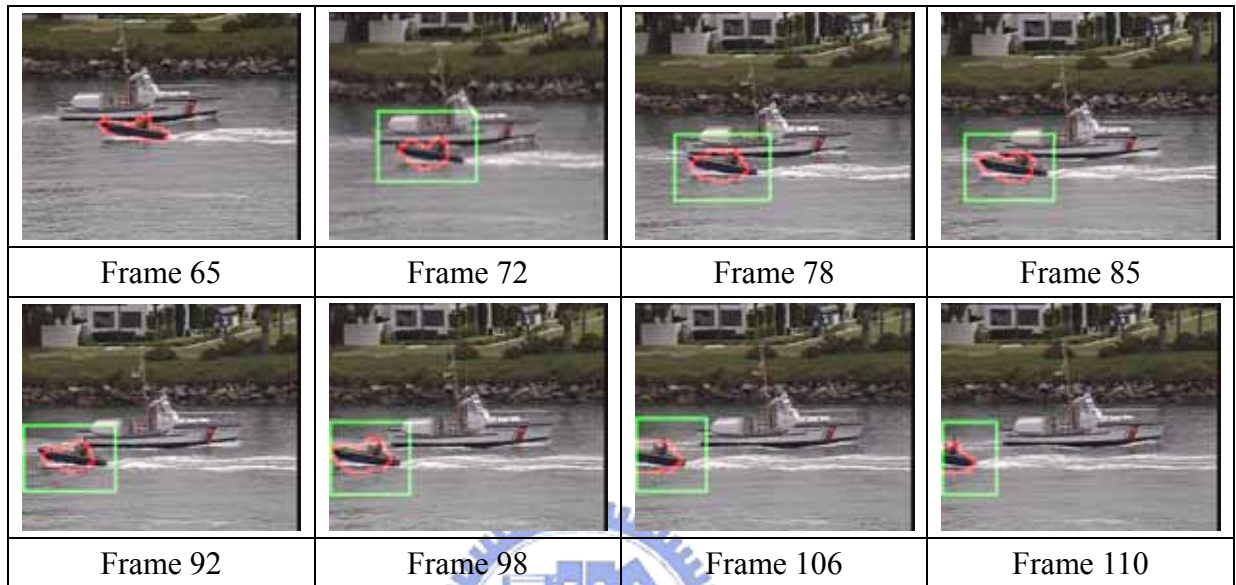
**Figure 3-18** Football sequences with original region tracking model. The contour grows incorrectly because of the interference of the environment clutter.



**Figure 3-19** Track a small target in the coastguard sequence. The level surface with higher resolution is only evaluated inside the green rectangle. More computational resources can be allotted to a specific region and produce a higher quality result.

Figure 3-19 shows the result of the coastguard sequences. Because the target is small, it is beneficial that the level surface updates only inside the small green rectangle. More computational resource can be reserved for updating the level surface with

higher resolution. A low resolution level surface is used in Figure 3-20. Although the tracker cannot get a fine boundary of the small boat, the essential position information is not lost. Selecting the scalability (or resolution) of the level surface is a tradeoff between the computational cost and the quality.



**Figure 3-20** Use a level surface with lower resolution. This reduces large computational cost. The essential position is successfully tracked but the fine boundary of the object is lost.

### 3.3 Probability Model

The region tracking model in the previous section builds the probability model based on the difference of intensity or color. Ideally every pixel in the current frame must search for the minimum-square-error point in the entire reference frame. A limited search range can reduce the computational cost. So far, only local characteristic is taken into account.

The original  $P_{in}$  and  $P_{out}$  in (Eq. 3-8) are modeled by (Eq. 3-10) and (Eq. 3-11) in [8]. In this section,  $P_{in}$  and  $P_{out}$  will be obtained by actually gathering the statistics of the image data. The searching procedure in (Eq. 3-10) and (Eq. 3-11) will be used as the prior model for the maximum a posteriori (MAP) method.



### 3.3.1. Kernel Density Estimation

Assume the intensity value is between 0 and 255. If the probability of every value from 0 to 255 is counted, no obvious “mode” or characteristics can be found from this density function because the accumulated number for each intensity value is too small. For color images, an impractical  $256^3$  storage space is required. It is better to classify the image points with similar intensity into “bins”. For example, if the value between 0 and 255 is divided into 8 “bins”, the points with value between 0 and 31 belong to the same bin and the characteristics of the data can be modeled by 8 counting numbers.

If the sampled data clusters near the boundaries of the bins, this may produce a poor estimation. A kernel may be chosen to “smooth” [14] the distribution in the statistical space. This technique is called Kernel Density Estimation (KDE). The kernel density estimator is expressed by

$$\hat{f}(\bar{x}) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\|\bar{x} - \bar{x}_i\|}{h}\right), \quad (\text{Eq. 3-20})$$

where  $n$  is the number of  $d$ -dimensional data points, and  $h$  is the bandwidth of the kernel  $K$ . The vector  $\bar{x}$  is a scalar for gray level image data. The Epanechnikov kernel is

$$K_E(\bar{x}) = \begin{cases} 0.5 c_d^{-1} (d+2) (1 - \|\bar{x}\|^2), & \|\bar{x}\| \leq 1 \\ 0, & \text{otherwise.} \end{cases}$$

and the multivariate normal kernel is

$$K_N(\bar{x}) = (2\pi)^{-d/2} \exp\left(-\frac{1}{2}\|\bar{x}\|^2\right).$$



**Figure 3-21**

The blue line is the histogram of the gray-level Lena image. The red line is the kernel density estimation result.

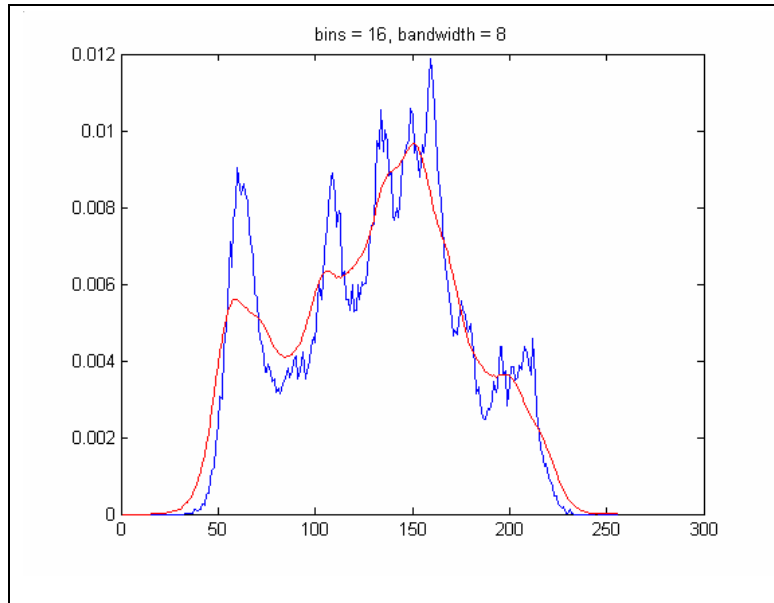
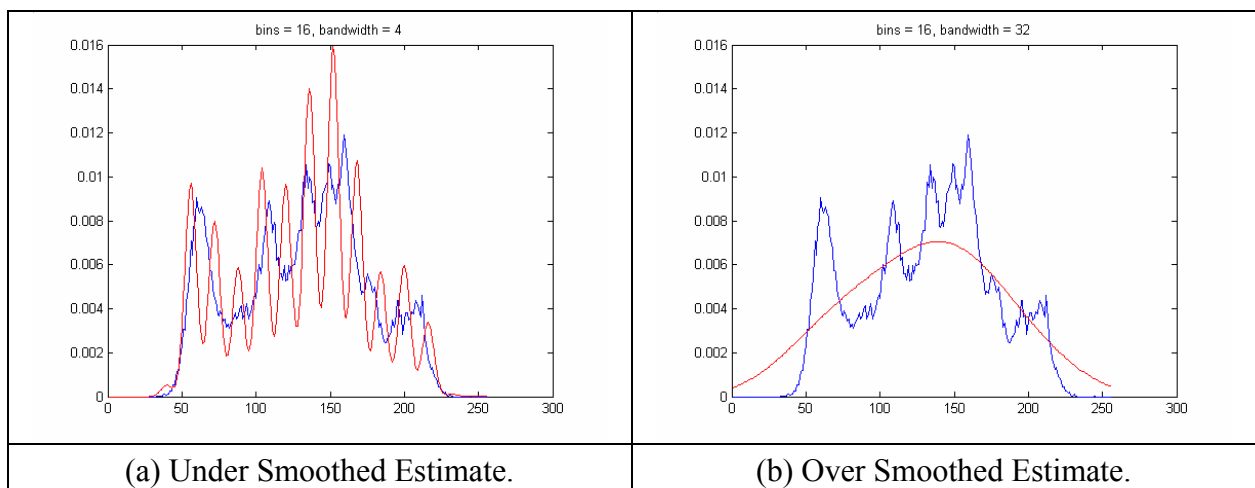


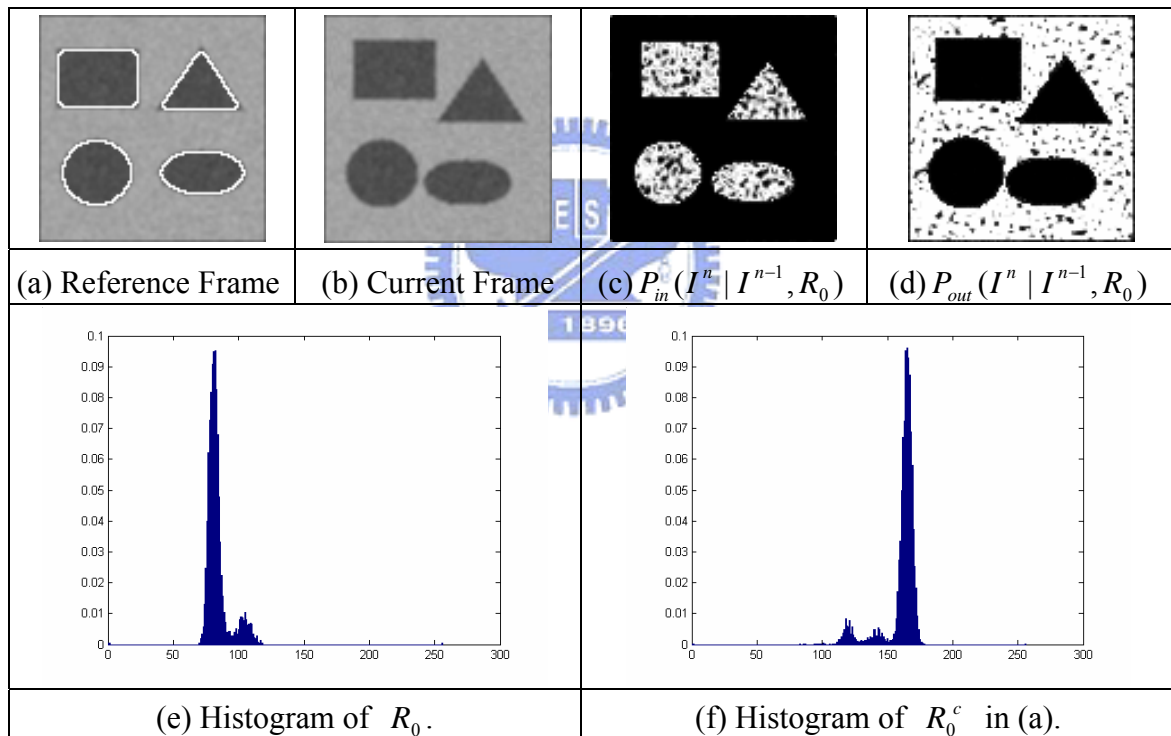
Figure 3-21 shows an example of kernel density estimation. The blue line is the histogram of the gray-level Lena image. The red line is the kernel density estimation result with 16 bins. The bandwidth of the normal kernel is 8. The bandwidth selection is a very important issue in kernel density estimation. Figure 3-22 (a) shows the result when the bandwidth is too small, it causes under-smoothed. If the bandwidth is too large, the over-smoothed result is shown in Figure 3-22 (b). A good bandwidth selection can be based on many criterions such as the mean integrated squared error (MISE) and the approximate mean integrated square error (AMISE).



**Figure 3-22** Improper kernel bandwidth selection. If the bandwidth is too small, it causes under-smoothed. If the bandwidth is too large, it causes over-smoothed.

Kernel density estimation is widely used in analyzing data distribution. The mean shift approach [15] is derived based on this probability model and may cluster the data into groups. Dorin Comaniciu *et al.* use the mean shift algorithm to deal with image segmentation [16] and object tracking [17] problem. The mean shift algorithm shifts the data in the direction of the greatest gradient of its distribution density function. Although the derivative result doesn't have to do the kernel density estimation, the mean shift tracking algorithm cannot track the boundary of the moving object. Now the probability model built by the kernel density estimation is going to be applied to the likelihood energy function shown in (Eq. 3-8).

### 3.3.2. ML and MAP Estimation



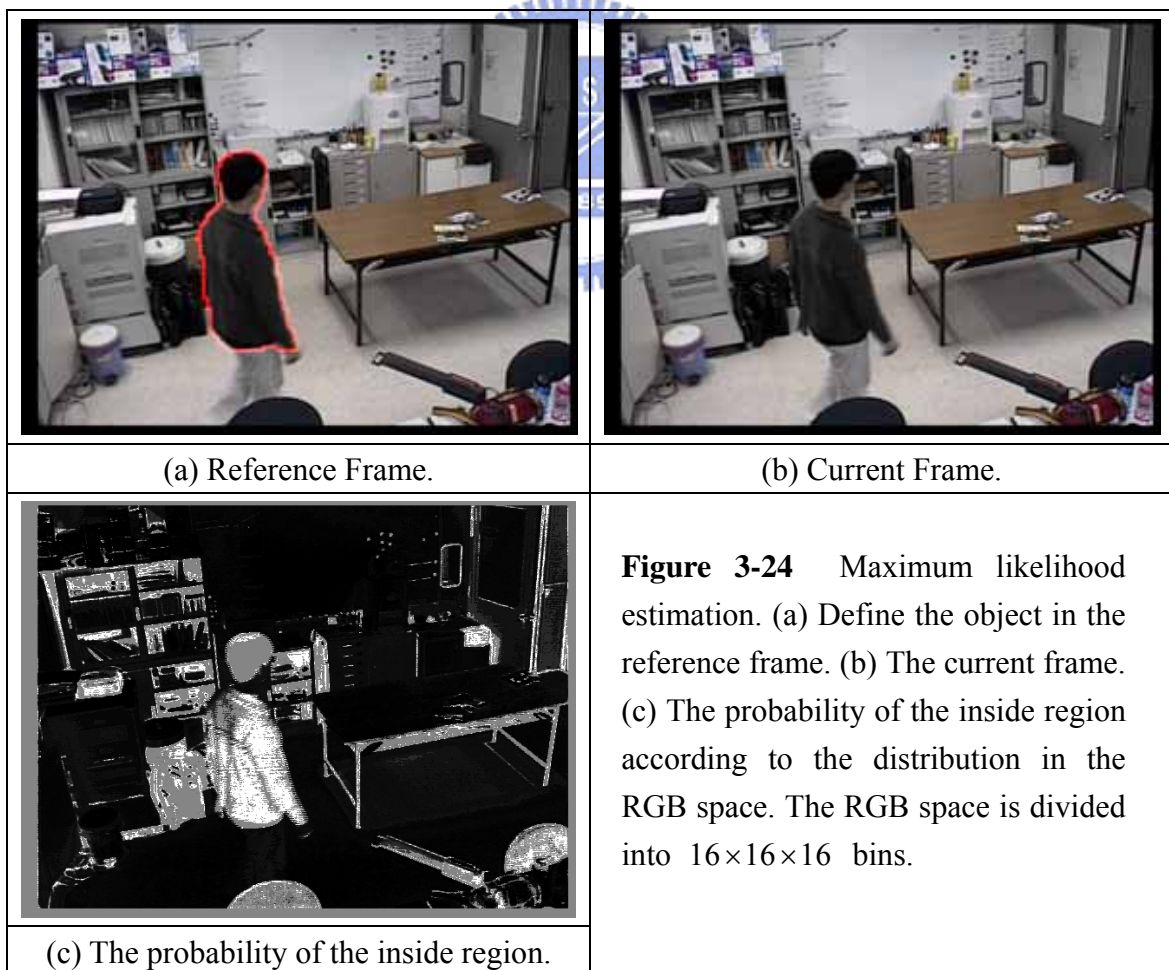
**Figure 3-23** Maximum likelihood. (a) The preset contours which define the objects in the reference frame. (b) The objects move in the current frame. (c) The probability of inside region in the current frame. (d) The probability of outside region in the current frame. (e) and (f) are the histograms inside and outside the contours in (a).

The maximum likelihood can be illustrated by Figure 3-23. (a) is the current frame with slight noise and the contours define the objects. The inside region is  $R_0$  and the outside region is  $R_0^c$ . The histograms of  $R_0$  and  $R_0^c$  are shown in (e) and (f),

respectively. The peak intensity value is 80 inside and 164 outside. Then in (b) the objects move. In order to estimate the objects' new positions, (c) shows the probability inside the objects in the current frame and (d) shows the outside probability.

Replacing  $P_{in}(I^n | I^{n-1}, R_0)$  and  $P_{out}(I^n | I^{n-1}, R_0)$  in (Eq. 3-8), the energy function becomes the active region model as expressed in (Eq. 2-25). This is called a Maximum Likelihood (ML) estimate.

For color images,  $P_{in}$  and  $P_{out}$  are built in the RGB space. Figure 3-24 shows the color image case. (a) and (b) are the reference and the current frame, respectively. The preset region is defined in the reference frame. Then the probability of the inside region in the current frame is shown in (c), which is similar to Figure 3-23 (c). The distribution is estimate by dividing the RGB space into  $16 \times 16 \times 16$  bins.



In Figure 3-24 (c), the background regions with color similar to the inside region also have high probability. If the moving object touches these regions, the ML estimator may misclassify the background region as a foreground region. A more reliable estimator with the maximum a posteriori (MAP) criterion will be proposed as follows.

Assume the inside probability in the current frame with  $I^n$ ,  $I^{n-1}$ , and  $R_0$  are given is represented as  $P(IN | I^n, I^{n-1}, R_0)$ .

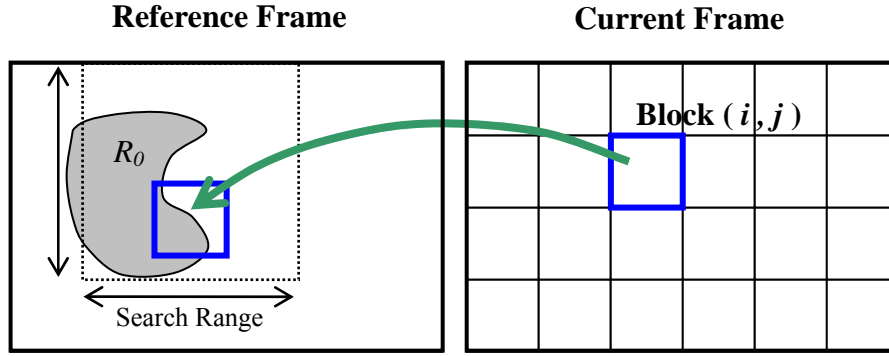
Based on the Bayes' Rule,

$$\begin{aligned}
 P(IN | I^n, I^{n-1}, R_0) &= \frac{P(IN)P_{in}(I^n | I^{n-1}, R_0)}{P(IN)P_{in}(I^n | I^{n-1}, R_0) + P(OUT)P_{out}(I^n | I^{n-1}, R_0)} \\
 &= \frac{P(IN)P_{in}(I^n | I^{n-1}, R_0)}{P(IN)P_{in}(I^n | I^{n-1}, R_0) + (1 - P(IN))P_{out}(I^n | I^{n-1}, R_0)} \quad \text{(Eq. 3-21)}
 \end{aligned}$$

Also note that  $P(OUT | I^n, I^{n-1}, R_0) = 1 - P(IN | I^n, I^{n-1}, R_0)$ .

The terms  $P_{in}(I^n | I^{n-1}, R_0)$  and  $P_{out}(I^n | I^{n-1}, R_0)$  can be obtained by the same way as Figure 3-23 (c) and (d). But now a 3-dimensional distribution from the kernel density estimation is considered for color images. The prior  $P(IN)$  is assumed to be  $1/Z$  for every pixel in the Active Region Model in Section 2.4.  $Z$  is the number of regions.  $Z$  is equal to 2 because now we only consider inside and outside probabilities.

Since the inside region is known in the reference frame, the prior part can be estimated more reasonable instead of setting a fixed number. The block matching is used as shown in Figure 3-25. The current frame is divided into many blocks. Every block finds the most matched region in the reference frame as the motion estimation does. Then the prior  $P(IN)$  of the Block( $i, j$ ) is estimated based on the proportion of  $R_0$ . All pixels in the block use the same prior value.



**Figure 3-25** Obtain the prior information. The current frame is divided into many blocks. Every block finds the most matched region within a search range in the reference frame. Then the proportion of  $R_0$  is used to estimate the prior  $P(IN)$  for all pixels of  $\text{Block}(i, j)$  in the current frame.

The above method is similar to the extended version in [8]. The prior  $P_{ij}(IN)$  of  $\text{Block}(i, j)$  is estimated to be

$$P_{ij}(IN) = \frac{\sum_{\bar{x} \in \text{Block}} R_0(\bar{x} + \bar{z})}{(\text{Block Size})^2}, \quad (\text{Eq. 3-22})$$

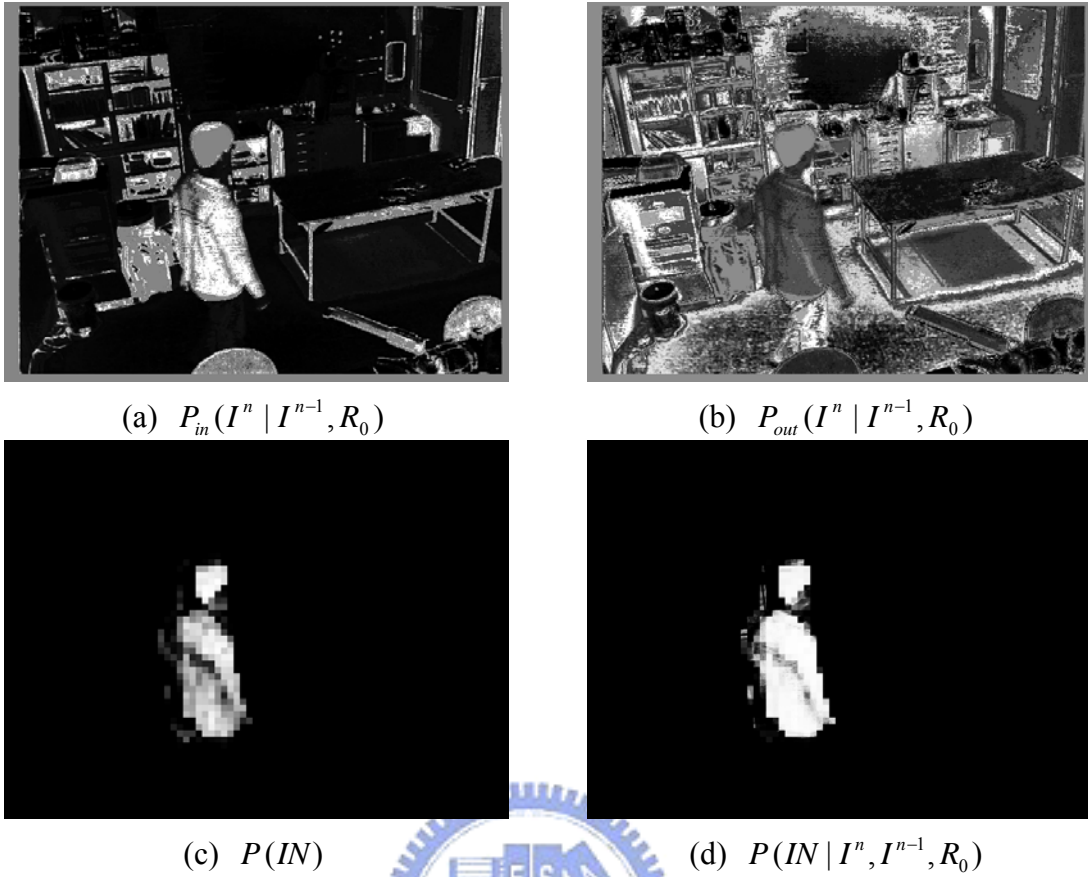
where

$$\bar{z} = \arg \min_{|\bar{z}| \leq \delta} \sum_{\bar{x} \in \text{Block}} (I^n(\bar{x}) - I^{n-1}(\bar{x} + \bar{z}))^2$$

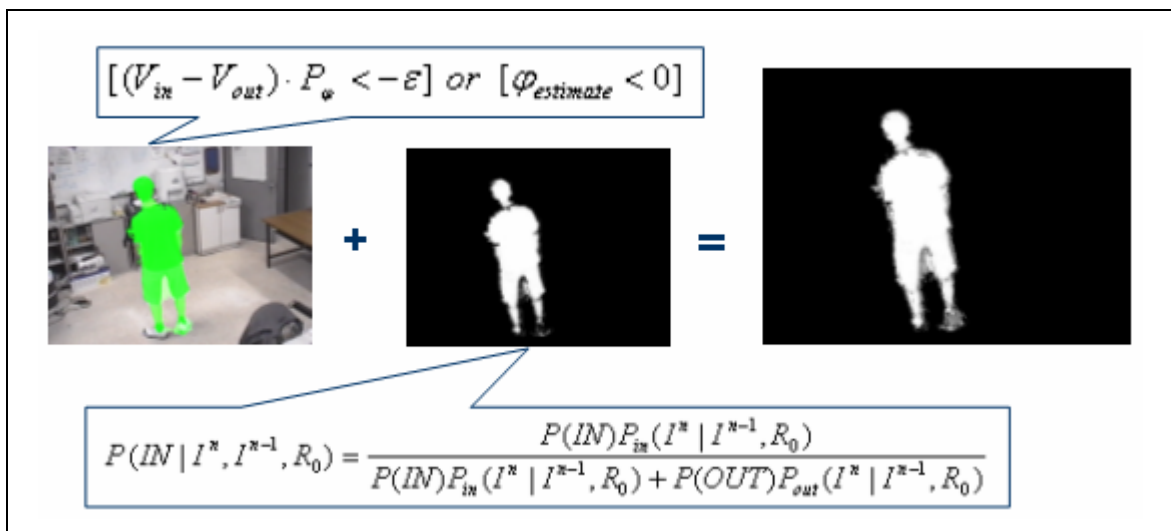
with  $R_0$  being 1 inside and 0 outside.

Figure 3-26 shows the individual terms for the MAP estimation. The predicted inside region is more accurate than the maximum likelihood model.

Figure 3-27 shows how to merge the inside information from the new tracking model and the probability model. The region tracking model can track the details of the inside region, but it produces false detection more easily. The probability model may fail if the colors inside and outside are too similar, but it offers robust information for regions with larger area. The total information can be obtained by merging these two components with different amounts of weight.



**Figure 3-26** The probability characteristic of the current frame in Figure 3-24 (b). (a) and (b) are the color version of Figure 3-23 (c) and (d). (c) is the prior information obtained by (Eq. 3-22). (d) is the MAP estimation from (Eq. 3-21).



**Figure 3-27** Merge the inside information of the new region tracking model and the probability model. The total inside information can be formed by using different amounts of weight for the two components.

Figure 3-28 shows the experimental result of real sequences. The camera moves with the person and the background changes. The region tracking model is applied and successfully tracks the person. The tracked region does not get affected by the surrounding clutters.



**Figure 3-28** The experimental result of real sequences acquired in the laboratory. The contour of the person is preset in the first frame.

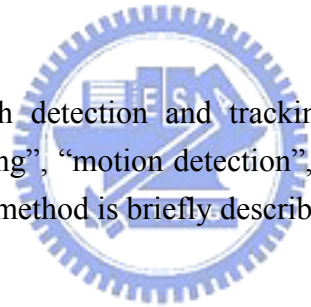


# Chapter 4.

## Implementation Issues and Experimental Results

This chapter discusses some implementation issues. First we show how to create a signed distance function. The level set can be evaluated more easily when the level surface is a signed distance function. Then the partial differential equations (PDE) are implemented based on the essentially non-oscillatory (ENO) polynomial interpolation scheme. Finally the re-initialization step makes the propagation of the contours more robust.

A complete system with detection and tracking is constructed. The system contains “background modeling”, “motion detection”, and “region tracking” steps. A simple background modeling method is briefly described in this chapter.



### 4.1 Implementation Issues

#### 4.1.1. Distance Transform

A binary image which represents the objects will be transformed into a distance function. Generating a distance function is very important when the level set theory is performed. Figure 4-1 (a) is a binary input image. The value 1 represents the object. The binary map in Figure 4-1 (a) will be transformed to (b). The values in (b) outside the object depend on their distance away from the object. The distance values inside the object are zero.

0	0	0	0	0	0	$\sqrt{5}$	$\sqrt{2}$	1	1	$\sqrt{2}$	$\sqrt{5}$
0	0	1	1	0	0	$\sqrt{2}$	1	0	0	1	2
0	1	1	1	0	0	1	0	0	0	1	$\sqrt{2}$
0	1	1	1	1	0	1	0	0	0	0	1
0	1	1	1	0	0	1	0	0	0	1	$\sqrt{2}$
0	0	0	0	0	0	$\sqrt{2}$	1	1	1	$\sqrt{2}$	$\sqrt{5}$
(a) Input binary image.						(b) Distance Transform of (a).					

**Figure 4-1** Distance Transform. In (b), the values outside the object depend on their distance away from the object.

A simple 1-D example is shown in Table 4-1. A binary map  $b(x)$  is 1 inside the object and 0 outside the object.

Step 1: The object region is set to be zero while the background region is set to be an “as large as possible” value.

Step 2: Use a three-tap mask  $M_1 = [m_1, m_2, m_3] = [1, 0, Inf.]$  to scan from left to right.

$$d(x) = \min\{b(x-1) + m_1, b(x) + m_2, b(x+1) + m_3\}$$

Step 3: Use a three-tap mask  $M_2 = [Inf., 0, 1]$  to scan from right to left similar as Step 2.

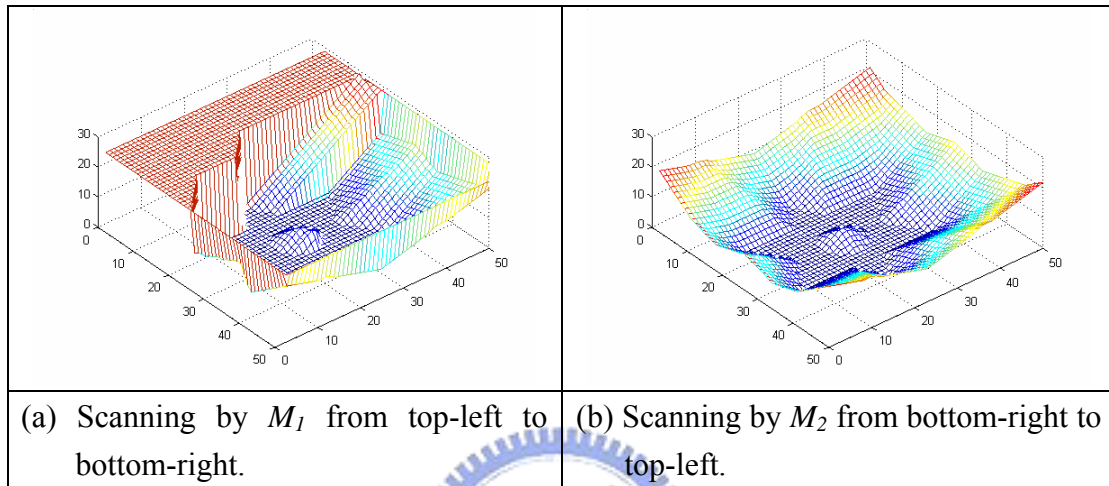
**Table 4-1** Create the 1D distance function.

x	1	2	3	4	5	6	7	8	9	10	11
b(x)	0	0	0	0	1	1	1	0	0	0	0
Step 1	Inf.	Inf.	Inf.	Inf.	0	0	0	Inf.	Inf.	Inf.	Inf.
Step 2	Inf.	Inf.	Inf.	Inf.	0	0	0	1	2	3	4
Step 3	4	3	2	1	0	0	0	1	2	3	4

The above evolution is called a “Two-pass Algorithm.” The distance value is accumulated when  $M_1$  scans in one direction, and then  $M_2$  scans in the other direction. In the two dimensional case, the masks are:

$$M_1 = \begin{bmatrix} \sqrt{2} & 1 & \sqrt{2} \\ 1 & 0 & Inf \\ Inf & Inf & Inf \end{bmatrix}, \quad M_2 = \begin{bmatrix} Inf & Inf & Inf \\ Inf & 0 & 1 \\ \sqrt{2} & 1 & \sqrt{2} \end{bmatrix}$$

$M_1$  scans from top-left to bottom-right, while  $M_2$  scans from bottom-right to top-left. The  $Inf$  is a preset large value. The 2D example is shown in Figure 4-2.



**Figure 4-2** Two-pass algorithm in 2D. The binary input image is shown as the right star image. The first pass accumulates the distance value from top-left to bottom-right, while the second pass accumulates the distance value from bottom-right to top-left.



There are many different distance measures that create different distance functions. Four popular measures, Euclidean, City Block, Chessboard, and Quasi-Euclidean, are shown in Figure 4-3. The center point denotes the location of the object, so the center value is zero. The values of the City Block and the Chessboard measure are integers and are suitable for hardware implementation. The Euclidean measure is more precise but it contains floating point numbers. The Quasi-Euclidean measure can compute more points at one time and speed up the computation. These matrices can be decomposed into  $M_1$  and  $M_2$  for the two-pass algorithm.

<table border="1"> <tbody> <tr><td><math>\sqrt{2}</math></td><td>1</td><td><math>\sqrt{2}</math></td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td><math>\sqrt{2}</math></td><td>1</td><td><math>\sqrt{2}</math></td></tr> </tbody> </table>	$\sqrt{2}$	1	$\sqrt{2}$	1	0	1	$\sqrt{2}$	1	$\sqrt{2}$	<table border="1"> <tbody> <tr><td>2</td><td>1</td><td>2</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>2</td><td>1</td><td>2</td></tr> </tbody> </table>	2	1	2	1	0	1	2	1	2	<table border="1"> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	1	1	1	1	0	1	1	1	1	<table border="1"> <tbody> <tr><td><math>\sqrt{8}</math></td><td><math>\sqrt{5}</math></td><td>2</td><td><math>\sqrt{5}</math></td><td><math>\sqrt{8}</math></td></tr> <tr><td><math>\sqrt{5}</math></td><td><math>\sqrt{2}</math></td><td>1</td><td><math>\sqrt{2}</math></td><td><math>\sqrt{5}</math></td></tr> <tr><td>2</td><td>1</td><td>0</td><td>1</td><td>2</td></tr> <tr><td><math>\sqrt{5}</math></td><td><math>\sqrt{2}</math></td><td>1</td><td><math>\sqrt{2}</math></td><td><math>\sqrt{5}</math></td></tr> <tr><td><math>\sqrt{8}</math></td><td><math>\sqrt{5}</math></td><td>2</td><td><math>\sqrt{5}</math></td><td><math>\sqrt{8}</math></td></tr> </tbody> </table>	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	2	1	0	1	2	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$
$\sqrt{2}$	1	$\sqrt{2}$																																																					
1	0	1																																																					
$\sqrt{2}$	1	$\sqrt{2}$																																																					
2	1	2																																																					
1	0	1																																																					
2	1	2																																																					
1	1	1																																																					
1	0	1																																																					
1	1	1																																																					
$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$																																																			
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$																																																			
2	1	0	1	2																																																			
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$																																																			
$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$																																																			
Euclidean	City Block	Chessboard	Quasi-Euclidean																																																				

**Figure 4-3** Four distance measures for creating a distance function.

A signed distance function is required in the level set evolution. The function value is negative inside the object and is positive outside the object. The above method only produces positive distance values outside the object. Negative distance values inside the object can be obtained by inverting the input binary image before performing the two pass algorithm and changing the sign of the final result. The surface near the boundary of the object is not smooth because both inside and outside distance values start from 1. Hence, the distance values outside the object are decreased by 0.5 and the values inside the object are increased by 0.5. This makes the absolute difference of two points “across” the zero-value boundary equal to 1.

Assume  $\varphi_{in}$  is set to be an arbitrary negative value if inside the object and an arbitrary positive value if outside the object. The transformation  $Dist(x)$  denotes the distance transform of the binary image  $x$ . The signed distance function  $\varphi$  is generated by

$$\varphi = [Dist(u(-\varphi_{in})) - 0.5] \cdot u(\varphi_{in}) - [Dist(u(\varphi_{in})) - 0.5] \cdot u(-\varphi_{in})$$

where  $u(x)$  is a Heaviside function:

$$u(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases}$$

## 4.1.2. Solve PDE Numerically

The partial differential equation (PDE) of the level set theory has the form

$$\varphi_t = F |\nabla \varphi|, \quad (\text{Eq. 4-1})$$

where  $\varphi$  is the time varying level set function according to the velocity function  $F$  and the gradient value of  $\varphi$ . The velocity function  $F$  depends on the curvature of the contour or whatever contour propagation model defined in the previous chapters.

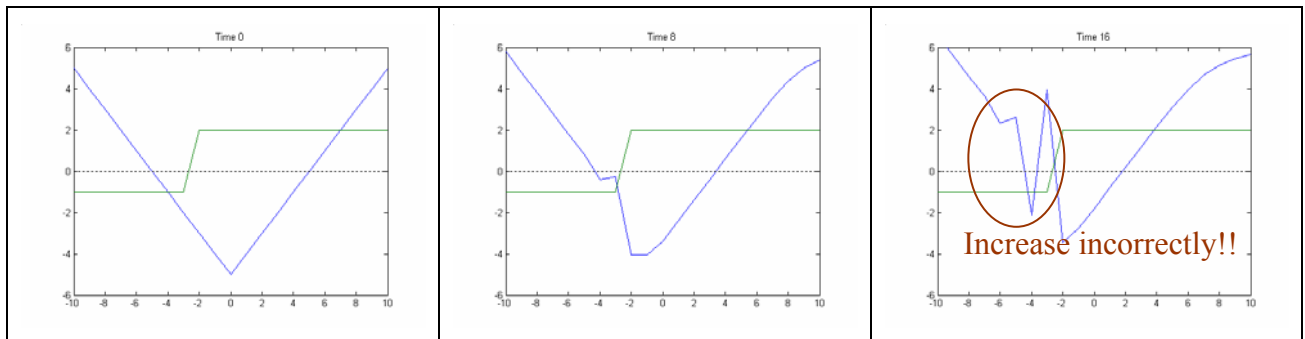
Consider a 1-D signed distance level set function shown as the solid blue line in Figure 4-4. The object locates in the interval  $[-5, 5]$ . The boundary of the object is at  $-5$  and  $5$  where the value of the level set function is zero. The green line is the velocity function  $F$  which is  $-1$  in  $[-10, -4]$  and  $2$  in  $[-3, 10]$ . When (Eq. 4-1) is implemented in the discrete time domain, there are two choices to approximate  $\nabla \varphi$ :

$$D_i^+ \varphi = \varphi_{i+1} - \varphi_i \quad (\text{Eq. 4-2})$$

or

$$D_i^- \varphi = \varphi_i - \varphi_{i-1}. \quad (\text{Eq. 4-3})$$

If we only choose (Eq. 4-2) or (Eq. 4-3) to approximate  $\nabla \varphi$ , the differentiation information will propagate in one direction and the level surface erroneously drifts one pixel after each time step. This error propagation is shown in Figure 4-4. If only  $D_i^+ \varphi$  is used to approximate  $\nabla \varphi$ , the level set function disobeys the original meaning of the PDE described in (Eq. 4-1).



**Figure 4-4** Error propagation result due to improper approximation of  $\nabla \varphi$ .

Equation with the form of (Eq. 4-1) is called a Hamilton-Jacobi Equation. The essentially non-oscillatory (ENO) polynomial interpolation is introduced in [11] to solve the PDE numerically. [12] [13] and [4] contain some practical applications. Here, (Eq. 4-1) is simplified as a one dimensional problem and is implemented via the following equations:

$$\varphi^{n+1} = \varphi^n + \Delta t [\max(F,0) \cdot \nabla^+ + \min(F,0) \cdot \nabla^-], \quad \text{(Eq. 4-4)}$$

where

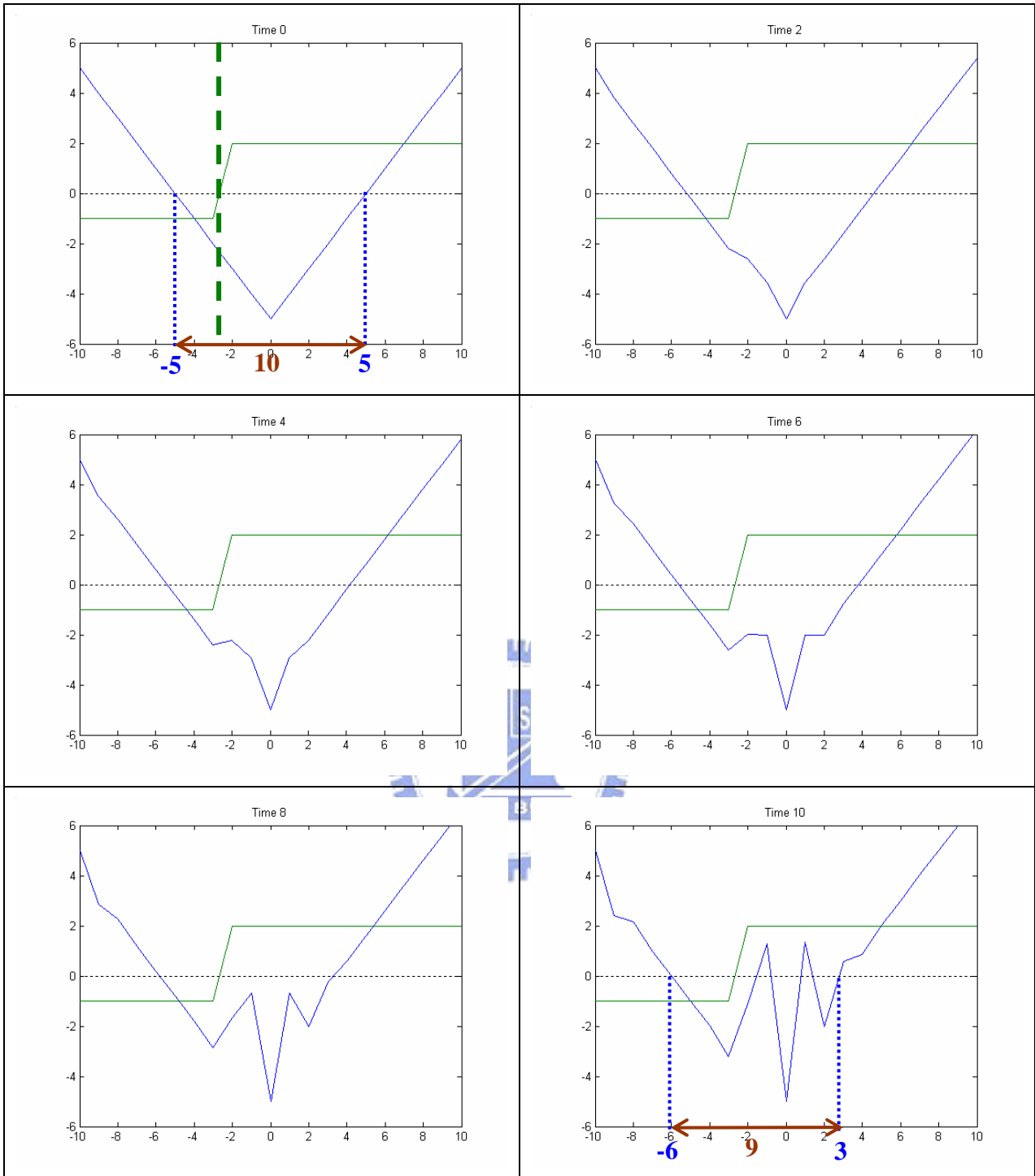
$$\nabla^+ = [\max(D_i^- \varphi, 0)^2 + \min(D_i^+ \varphi, 0)^2]^{1/2} \quad \text{(Eq. 4-5)}$$

$$\nabla^- = [\max(D_i^+ \varphi, 0)^2 + \min(D_i^- \varphi, 0)^2]^{1/2} \quad \text{(Eq. 4-6)}$$

and  $D_i^+ \varphi$ ,  $D_i^- \varphi$  are defined in (Eq. 4-2) and (Eq. 4-3), respectively. The time step  $\Delta t$  controls the propagation speed. The choice of  $\Delta t$  is a tradeoff between the propagation speed and the stability. With this scheme, a positive  $F$  is multiplied with  $\nabla^+$  while a negative  $F$  is multiplied with  $\nabla^-$ .

The level set function is updated by (Eq. 4-4) and the results of ten iterations with the ENO scheme are shown in Figure 4-5. The level set function propagates properly with respect to the velocity function  $F$ . The left zero surface value in Time 0 appears at -5 with velocity  $F = -1$ . The zero value propagates toward -6 after ten iterations with time step 0.1. In the same way, the right zero surface value propagate from 5 to 3 with  $F = 2$ .

However, the glitches of the level set function occur at the discontinuous points of  $F$  and the level surface. The value of the level set function inside the object finally “breaks” because of the glitch effects. These glitch effects can be eliminated by the re-initialization process that is to be introduced in the next section.



**Figure 4-5** The propagation of the 1-D level set function. The solid blue line is the level set function and the green line is the value of the velocity function.

### 4.1.3. Re-initialization

In order to eliminate the glitch effects and increase the robustness, frequent re-initializations should be applied. Re-initialization creates a new signed distance function according to the sign property of the level set function. This procedure eliminates the glitch effects and makes the contour propagation more stable.

The computation of a signed distance function is very expensive. If the time step is small, re-initialization is not necessary before the glitch effect “breaks” the negative region of the level set function. However, the penalty is that more iterations are required to propagate the level set function to the solution of the partial differential equation.

The re-initialization problem can be defined by another differential equation:

$$\begin{cases} \varphi_t + |\nabla\varphi| = 1, & \text{if } \varphi > 0 \\ \varphi_t - |\nabla\varphi| = -1, & \text{if } \varphi < 0 \end{cases} \quad \text{(Eq. 4-7)}$$

The solution of this PDE is constrained by the preset initial region  $R$  in the 2-D image space. The region inside the object is set to be an arbitrary negative value while the region outside the object is set to be an arbitrary positive value.

(Eq. 4-7) means that if  $|\nabla\varphi|$  is greater than 1 when  $\varphi > 0$ , we decrease the level function by  $(|\nabla\varphi| - 1)$ . If  $|\nabla\varphi|$  is greater than 1 when  $\varphi < 0$ , we increase the level function by  $(|\nabla\varphi| - 1)$ . When the level function doesn't change with respect to time, that is,  $\varphi_t = 0$ , then  $|\nabla\varphi| = 1$  and a signed distance level function is obtained.

(Eq. 4-7) can be integrated into one equation with a similar form like (Eq. 4-1) for implementation:

$$\varphi_t + S(\varphi)(|\nabla\varphi| - 1) = 0, \quad \text{(Eq. 4-8)}$$

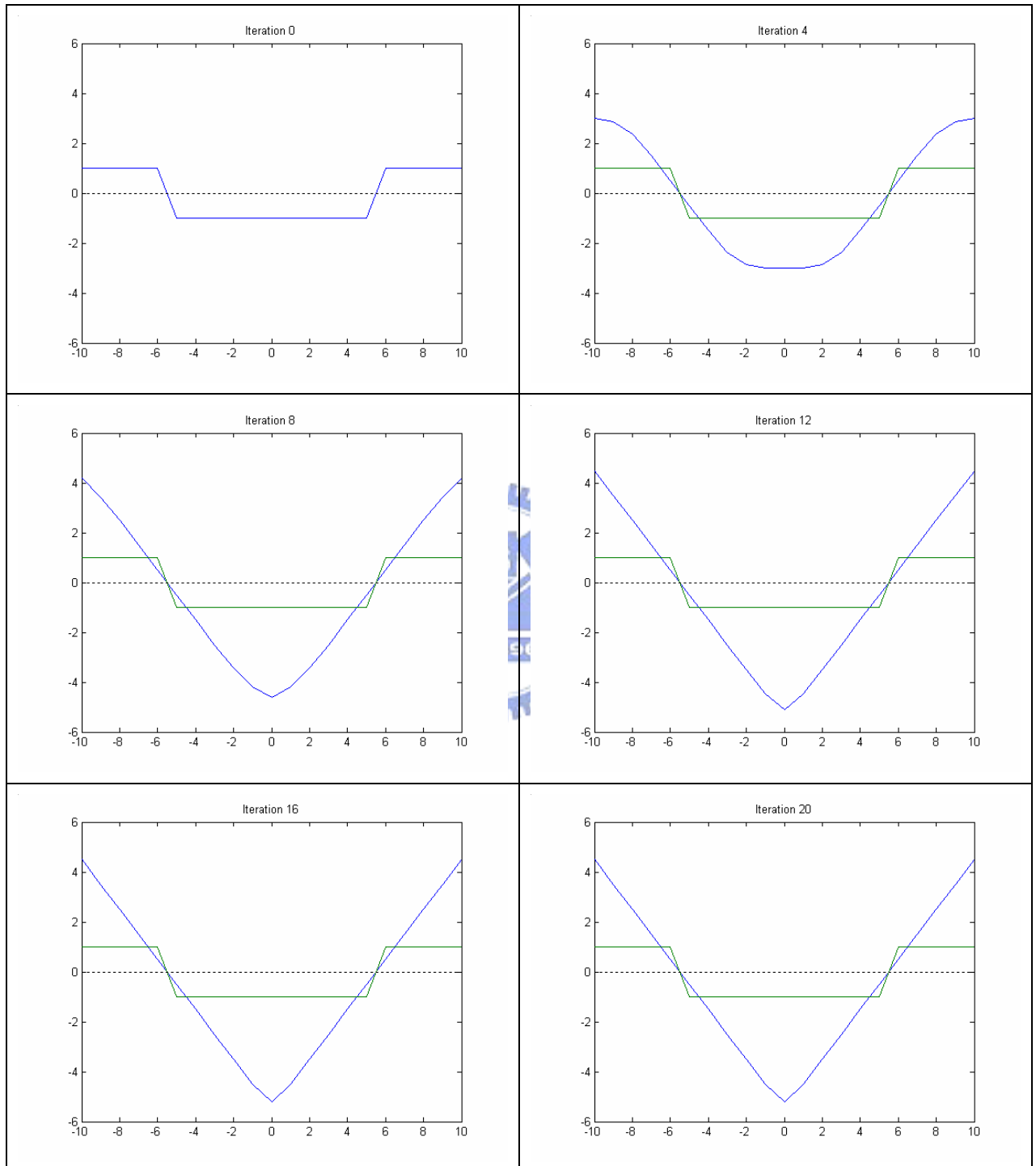
where  $S(\varphi)$  is a sign function defined as

$$S(\varphi) = \begin{cases} 1, & \text{if } \varphi > 0 \\ -1, & \text{if } \varphi < 0 \end{cases}$$

Because the difference of the level set function between each time step is not very large, the re-initialization scheme in (Eq. 4-8) can efficiently approximate a signed distance function for the next time step. The demonstration of (Eq. 4-8) is shown in Figure 4-6. The green line is the preset level set function and the blue line

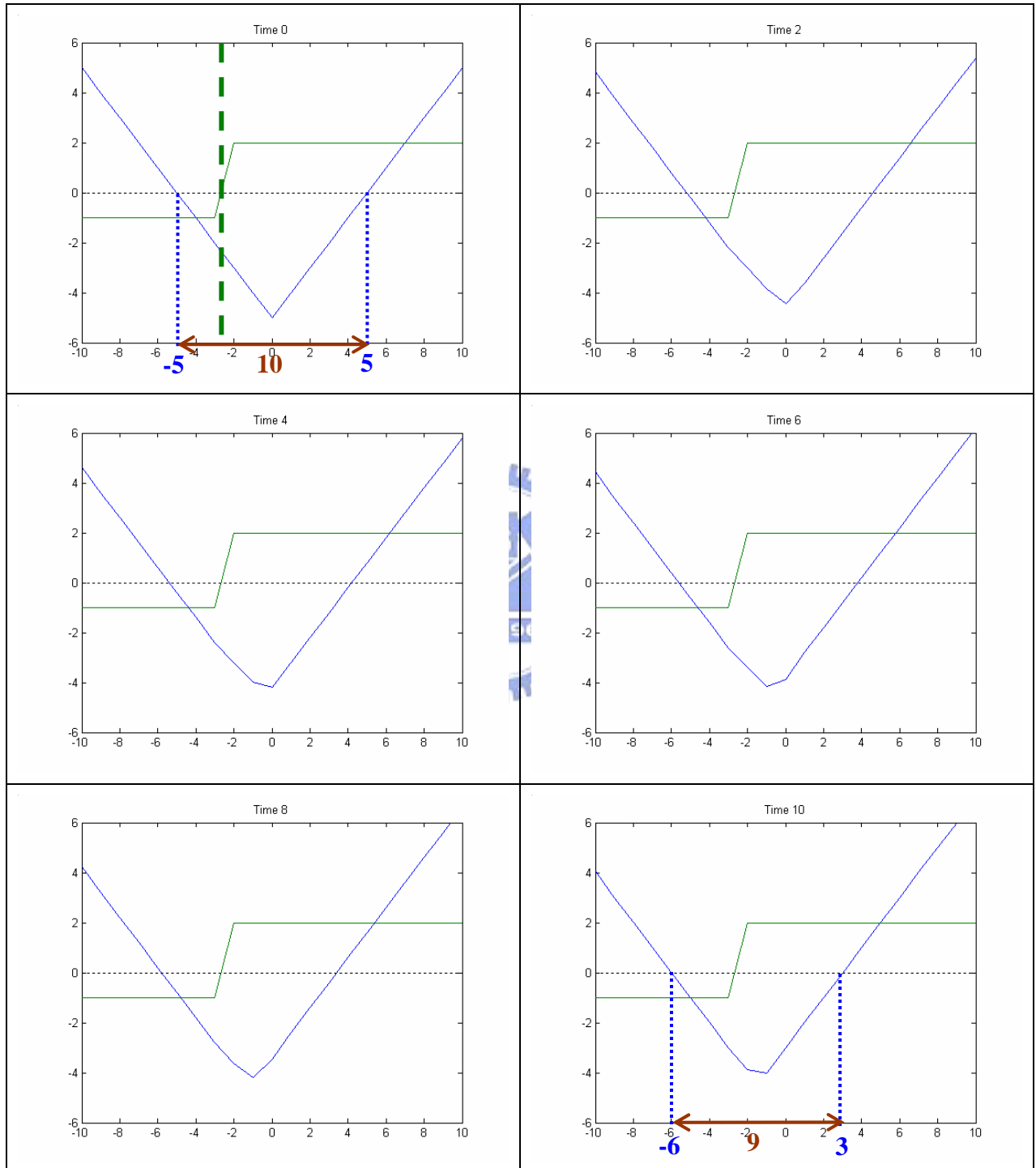


shows the propagation of the level set function according to the re-initialization procedure.



**Figure 4-6** Re-initialization demonstration of (Eq. 4-8). The green line is the preset level function and the blue line shows the variation of the level function according to the re-initialization procedure. The time step is set to be 0.5.

Now the re-initialization procedure is applied to the propagation of the level function with the same velocity force shown in Figure 4-5. The result is in Figure 4-7. Note that the glitches in Figure 4-5 are eliminated and the object region with negative values is not broken.



**Figure 4-7** The propagation of the level set function with velocity  $F$  in Figure 4-5 and adopt the re-initialization at the end of each time step. The glitches in Figure 4-5 are eliminated.

The above level set function propagation and re-initialization scheme can be extended to a two-dimensional version for image processing. The term  $\nabla^+$  and  $\nabla^-$  which approximate  $\nabla\varphi$  in (Eq. 4-5) and (Eq. 4-6) are modified as

$$\nabla^+ = [\max(D_{ij}^{-x}\varphi, 0)^2 + \min(D_{ij}^{+x}\varphi, 0)^2 + \max(D_{ij}^{-y}\varphi, 0)^2 + \min(D_{ij}^{+y}\varphi, 0)^2]^{1/2} \quad (\text{Eq. 4-9})$$

and

$$\nabla^- = [\max(D_{ij}^{+x}\varphi, 0)^2 + \min(D_{ij}^{-x}\varphi, 0)^2 + \max(D_{ij}^{+y}\varphi, 0)^2 + \min(D_{ij}^{-y}\varphi, 0)^2]^{1/2}, \quad (\text{Eq. 4-10})$$

where

$$D_{ij}^{+x}\varphi = \varphi_{i+1,j} - \varphi_{i,j}$$

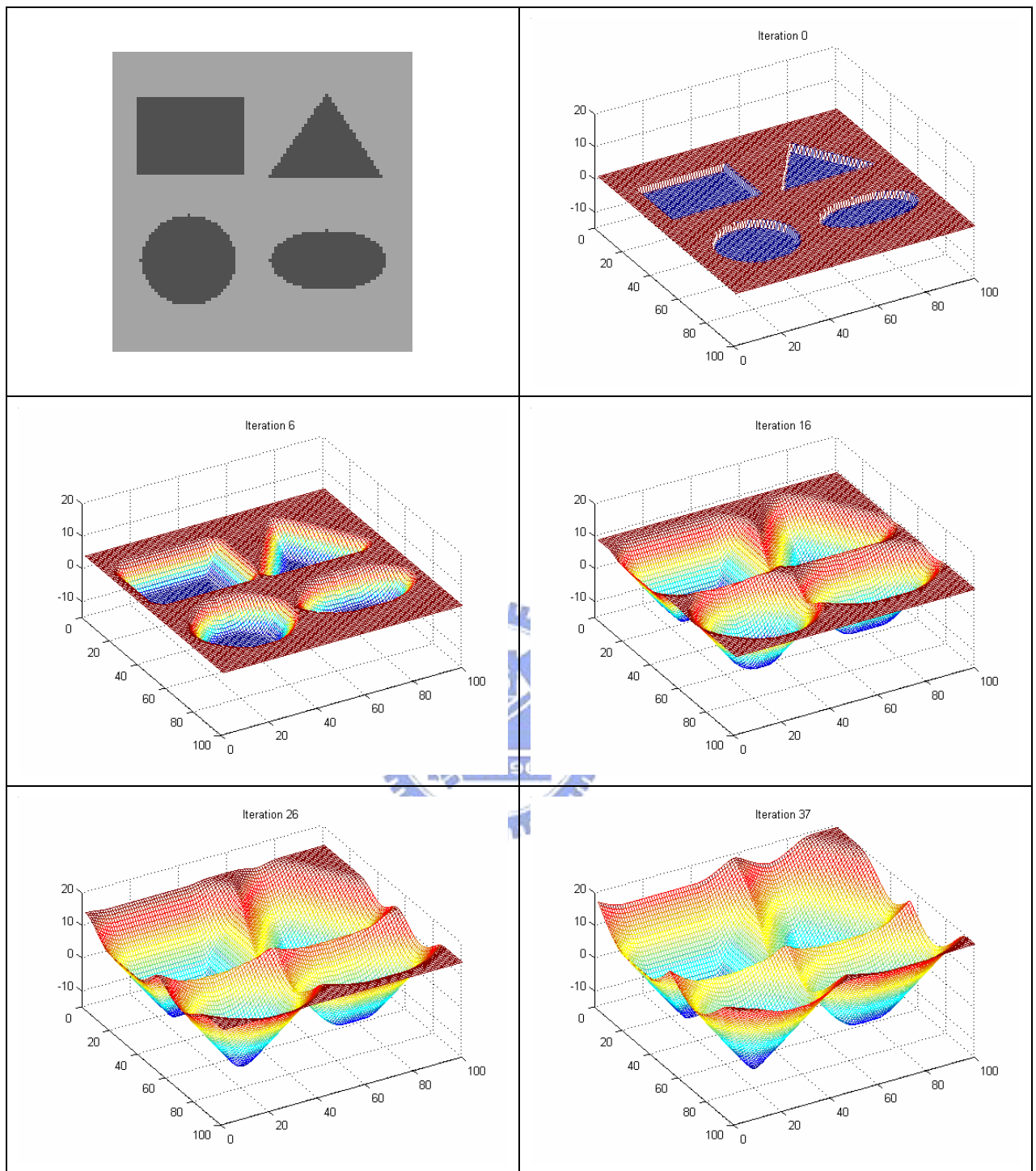
$$D_{ij}^{-x}\varphi = \varphi_{i,j} - \varphi_{i-1,j}$$

$$D_{ij}^{+y}\varphi = \varphi_{i,j+1} - \varphi_{i,j}$$

$$D_{ij}^{-y}\varphi = \varphi_{i,j} - \varphi_{i,j-1}.$$

Then, (Eq. 4-4) can be generalized into the 2-D space. Figure 4-8 demonstrates the re-initialization scheme in 2-D space. The level set function is initially set to be -1 inside the objects and 1 outside the objects. The level set function is updated by using (Eq. 4-8). The time step is set to be 0.5. The iteration stops when the maximum absolute difference of the level function between two iterations is less than ten times smaller than the preset time step.

Here is a short summary. In order to solve the partial differential equation (Eq. 4-1), the essentially non-oscillatory (ENO) polynomial interpolation is used. The spatial gradient of the level set function  $\nabla\varphi$  is approximated by (Eq. 4-9) and (Eq. 4-10). When the level function propagates, glitches are produced because of the discontinuity of the level function  $\varphi$  and the velocity function  $F$ . The re-initialization procedure is applied at the end of each time step. The re-initialization can be viewed as solving another PDE as (Eq. 4-8). Because the level set surfaces before and after the evolution of (Eq. 4-1) in one time step are very close, using (Eq. 4-8) to re-initialize the surface is more efficient than to create a new signed distance function again.



**Figure 4-8** The re-initialization in two dimension space. The level function is initially set to be -1 inside the objects and 1 outside the objects.

## 4.2 Build Background

In real surveillance system, the background image must be modeled. Even if the camera is static, the first acquired image can't be used as the background because it may contain moving objects. The background is obtained by analyzing successive images after a period of time. This is called background modeling.

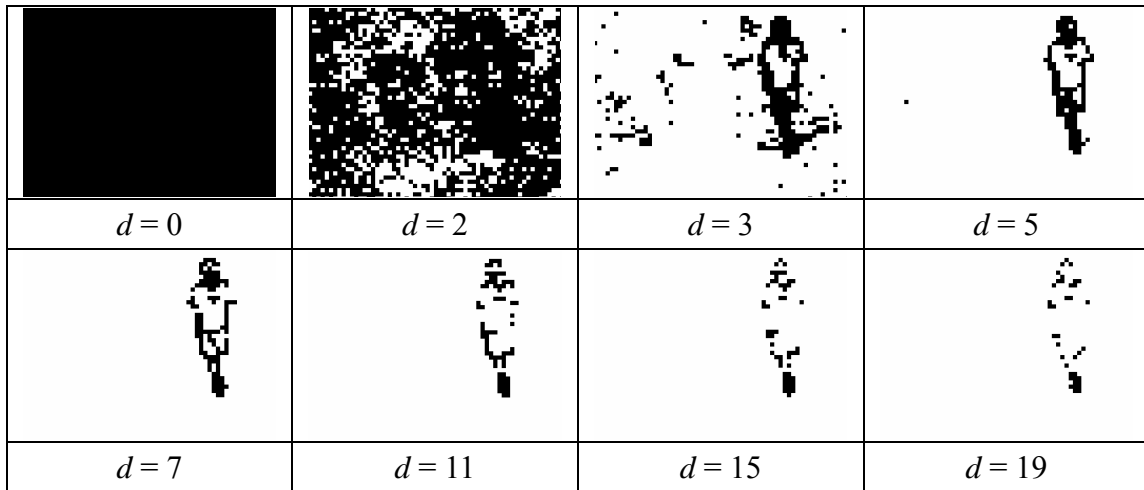
Ahmed Elgammal *et al* [18] proposed a way to model the foreground and background by using nonparametric kernel density functions. Although the Gaussian form of the density function can be implemented by a lookup table, the modeling method in [18] needs many sample frames to make the model more reliable. The storage and the computational time increase simultaneously. In this thesis, a simpler background modeling procedure will be introduced in this section.

### 4.2.1. Motion Detection without Background

The motion part in sequences can be extracted more easily by inter-frame difference. Boundaries of the moving objects are identified by setting a threshold on the absolute difference result.

Manipulating the image data pixel by pixel is very sensitive to noise. Hence, the inter-frame difference image is divided into many blocks and the sum of the absolute difference (SAD) within each block is used to identify the variation property. If the SAD of the block is above a predefined threshold, the block is classified as a motion block; otherwise, the block may be located on the background or the interior region of a moving object.

Experiments of different threshold value are shown in Figure 4-9. The value  $d$  represents the average difference per pixel in the block. If the SAD of the block is beyond  $d \times (\text{block size})^2$ , the block is marked as a motion block. The "black" pixels in Figure 4-9 indicate the motion blocks. The block size is  $8 \times 8$ . When  $d = 0$ , every block in the image is detected as motion block. As the  $d$  value increases, the number of detected blocks decreases. From the experiment result,  $d = 5$  is chosen empirically. In this example with the  $8 \times 8$  block size, a block is classified as a motion block if the sum of absolute difference  $SAD > 5 \times (8)^2 = 320$ .



**Figure 4-9** Motion Detection by thresholding the inter-frame difference image. The black pixels represents the motion block with the measurement  $d$ .

The size of block affects the storage requirement of the detection result. A smaller block size produces more blocks and the required storage increases. A smaller block is also more sensitive to noise. However, a smaller block size provides a better precision for the positioning of moving objects.

In the case  $d = 5$  in Figure 4-9, only the boundary of the moving object is detected. A more solid result is expected; otherwise, the center of the moving object may become the background region. The property of the level surface evolution in the level set theory may help classifying these cracked points. The level surface is updated by the following equation:

$$\frac{\partial \varphi_{ij}}{\partial t} = u\left(\left(d \times (\text{motion block})^2\right) - SAD_{ij}\right) |\nabla \varphi|, \quad (\text{Eq. 4-11})$$

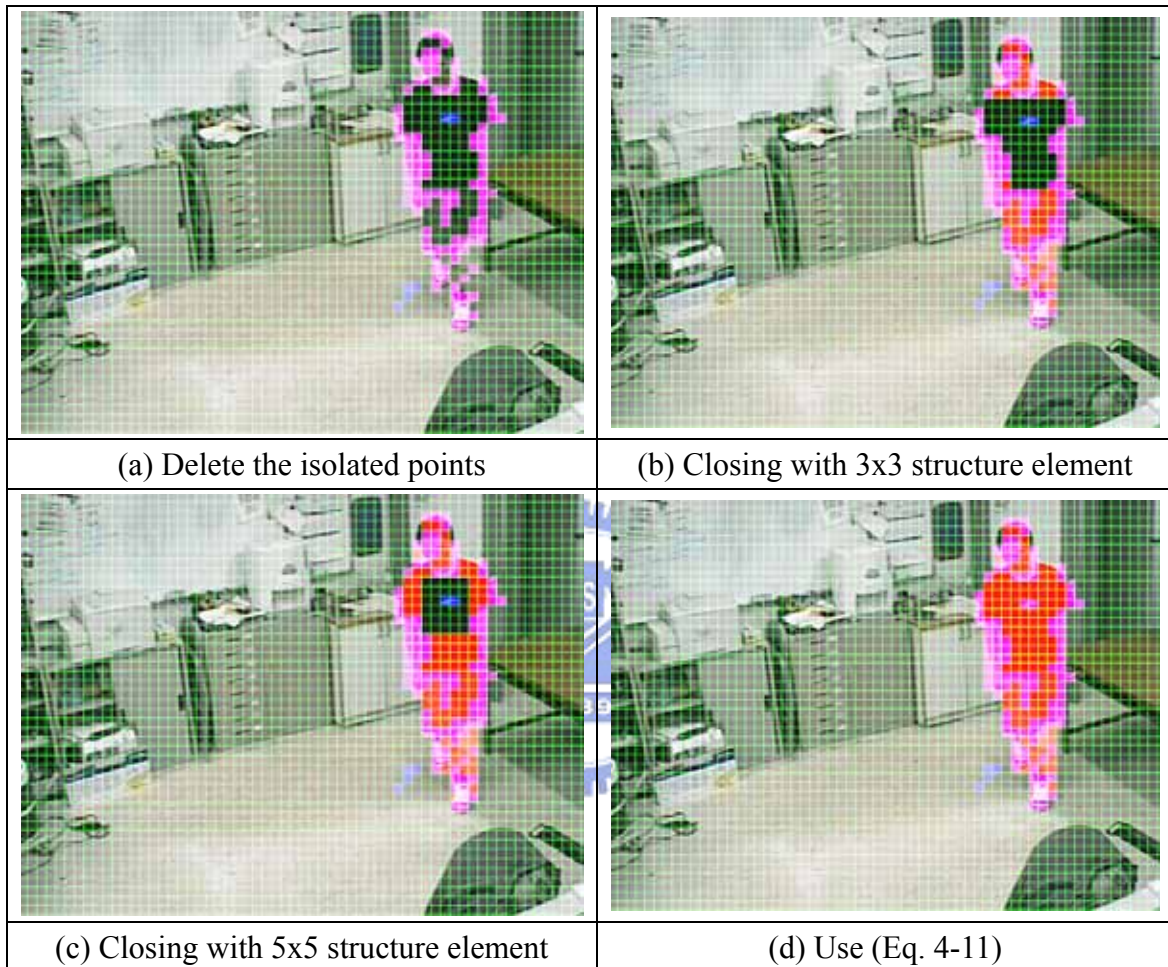
where  $u(x)$  is the unit step function and

$$SAD_{ij} = \sum_{p=1}^N \sum_{q=1}^N \left| I_t((i-1)N + p, (j-1)N + q) - I_{t-1}((i-1)N + p, (j-1)N + q) \right|. \quad (\text{Eq. 4-12})$$

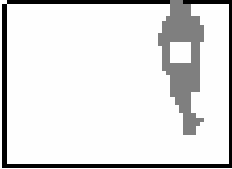
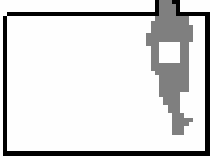
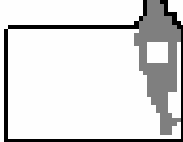
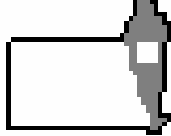
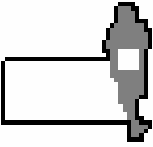
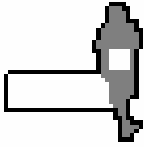
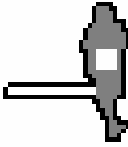

Here,  $N$  is the block size. The level surfaces moves upward if the  $SAD$  is less than  $d \times (\text{motion block})^2$ .

Dilation and erosion are used to link the boundary of the moving objects. In Figure 4-10 (b) and Figure 4-10 (c), 3x3 and 5x5 structuring elements are applied,

respectively. The result with the 3x3 structuring element still contains discontinuity on the boundary of the moving object. The result of the 5x5 structuring element is better. Figure 4-10 (d) is the result which applies (Eq. 4-11). The contour propagation is shown in Figure 4-11.



**Figure 4-10** Make a solid detection result. (a) The blue blocks show the original SAD detection result. (b) Morphological closing with a 3x3 structuring element. (c) Morphological closing with a 5x5 structuring element. (d) Fill the hole inside the moving object via (Eq. 4-11)

			
Iteration 1	Iteration 4	Iteration 7	Iteration 10
			
Iteration 13	Iteration 16	Iteration 19	Iteration 22

**Figure 4-11** Contour propagation using (Eq. 4-11).

## 4.2.2. Background Modeling

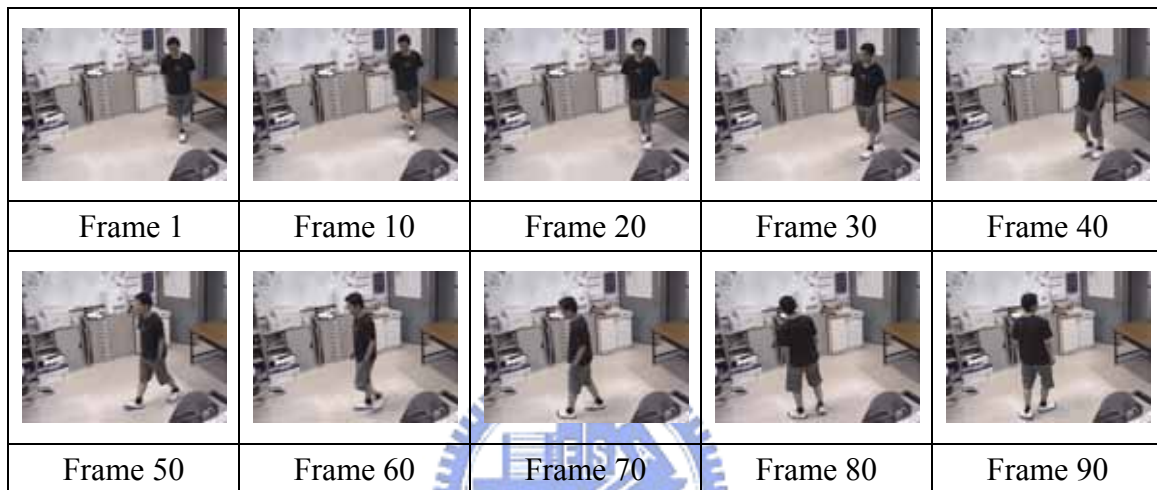
If the motion regions are extracted, the background can be built by accumulating the static image data. In the Video Surveillance and Monitoring (VSAM) project [19] at Carnegie Mellon University, a layered detection method is proposed. Every pixel in the image has three states: transient, static, and background. Pixels with high inter-frame difference value are defined as transient pixels. Otherwise, their states are defined to be static. Static pixels become background pixels if they keep their “static” state for a long enough time. The relationship between these three states is defined graphically. Moreover, the motion regions detected at different times are store in different layered maps so this method may deal with the occlusion problem.

If the initial background information is not available, the information has to be extracted from the sequences which contain moving objects. A preset background model is not adequate in a practical surveillance system because the background may change over time. Two types of blocks are defined using the graph concept in [19]. One is the “background” block, while the other is the “motion” block. The “static” block is not necessary here because the static region inside the moving objects is included in the motion region based on the method mentioned in Section 4.2.1. If a block is defined as a non-motion block in successive  $n$  frames, the averaged RGB values are defined as background. If the block remains static less than  $n$  frames, then we reset the counter and clear the accumulated value in the buffer.

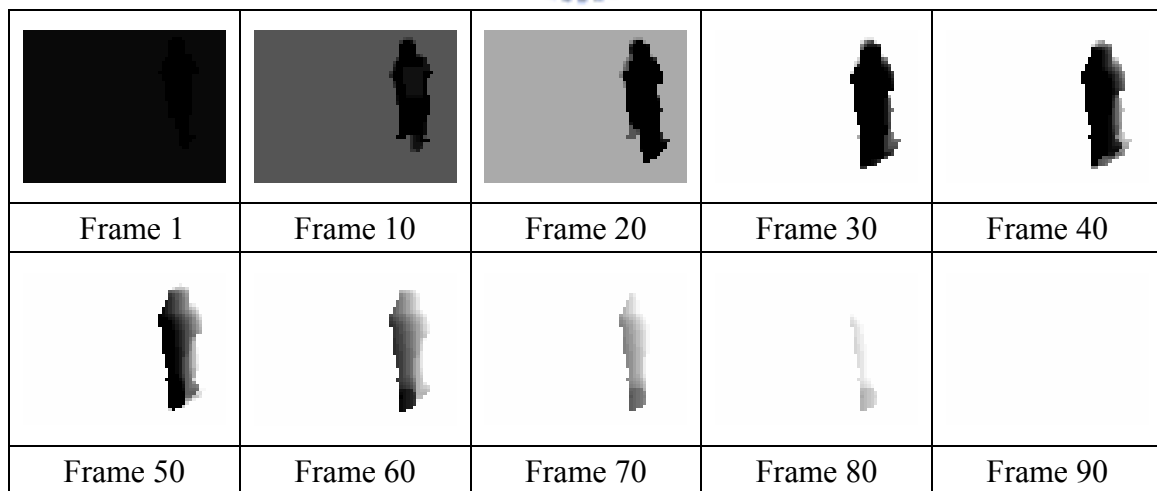
The test sequences are shown in Figure 4-12. The accumulation map is shown in Figure 4-13. The intensity denotes the accumulated number from 0 to  $n$ . The motion



region in the map is black because the accumulation number is reset to zero. When the person leaves its original position, the static background information occupied by the person is accumulated. Eventually the values of the entire accumulation map are  $n$  and the background modeling is completed. In this case,  $n$  is set to be 30. The background is built after 90 frames or about 3 seconds. The background modeling time strongly depends on the displacement of the moving object. If the object keeps moving in the same position, the background can't be built until the object leaves the position.



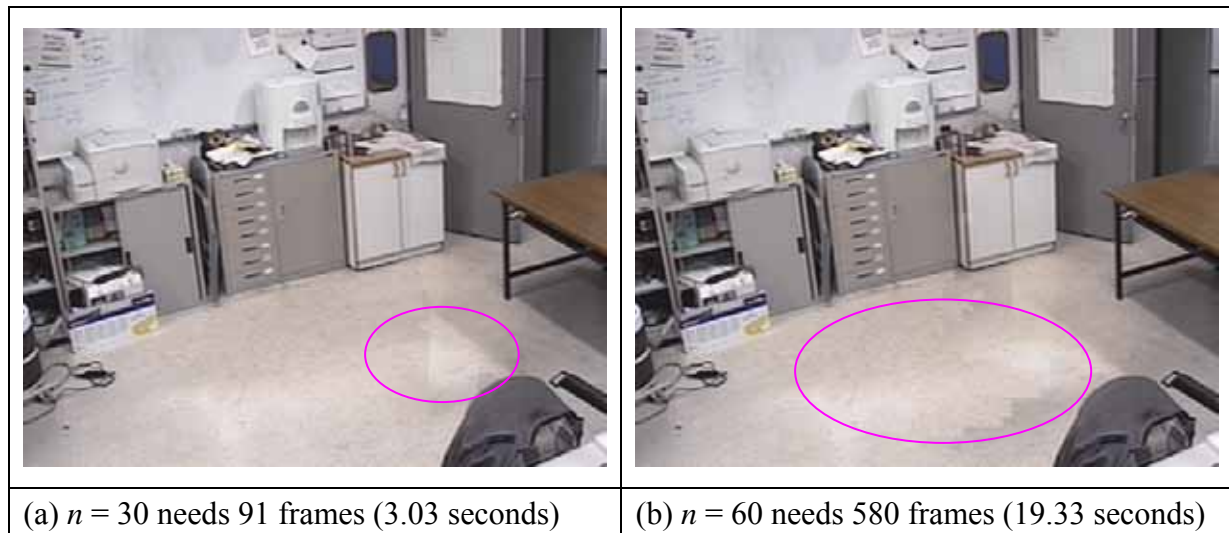
**Figure 4-12** Test sequences for background modeling.



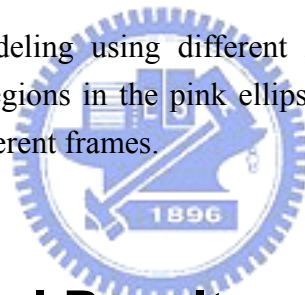
**Figure 4-13** Background modeling process with  $n = 30$ . The white regions are the background region.

When  $n$  increases, the required number of frames increases. Figure 4-14 (a) and (b) show the results of  $n = 30$  and  $n = 60$ , respectively. The discontinuous regions in

the pink ellipses are due to the different intensity values from different frames. The discontinuous defects can be tolerated because the subtraction data will be classified by the “active contours without edges” model. A larger  $n$  costs extra consuming time and does not necessarily produce a better result.



**Figure 4-14** Background modeling using different  $n$ 's. (a)  $n = 30$ , (b)  $n = 60$ . The discontinuous regions in the pink ellipses are due to the different intensity values from different frames.

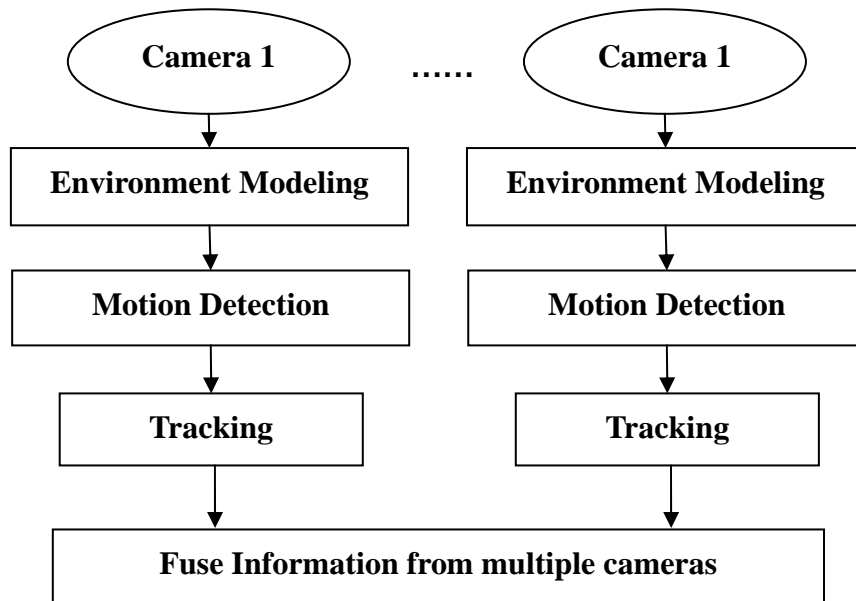


## 4.3 Experimental Results

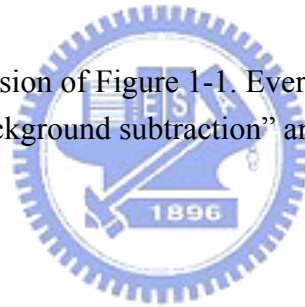
A simplified version of Figure 1-1 is shown in Figure 4-15. Every camera system contains “background modeling”, “background subtraction” and “region tracking” processes. Figure 4-16 shows the flow chart about how to control these three processes.

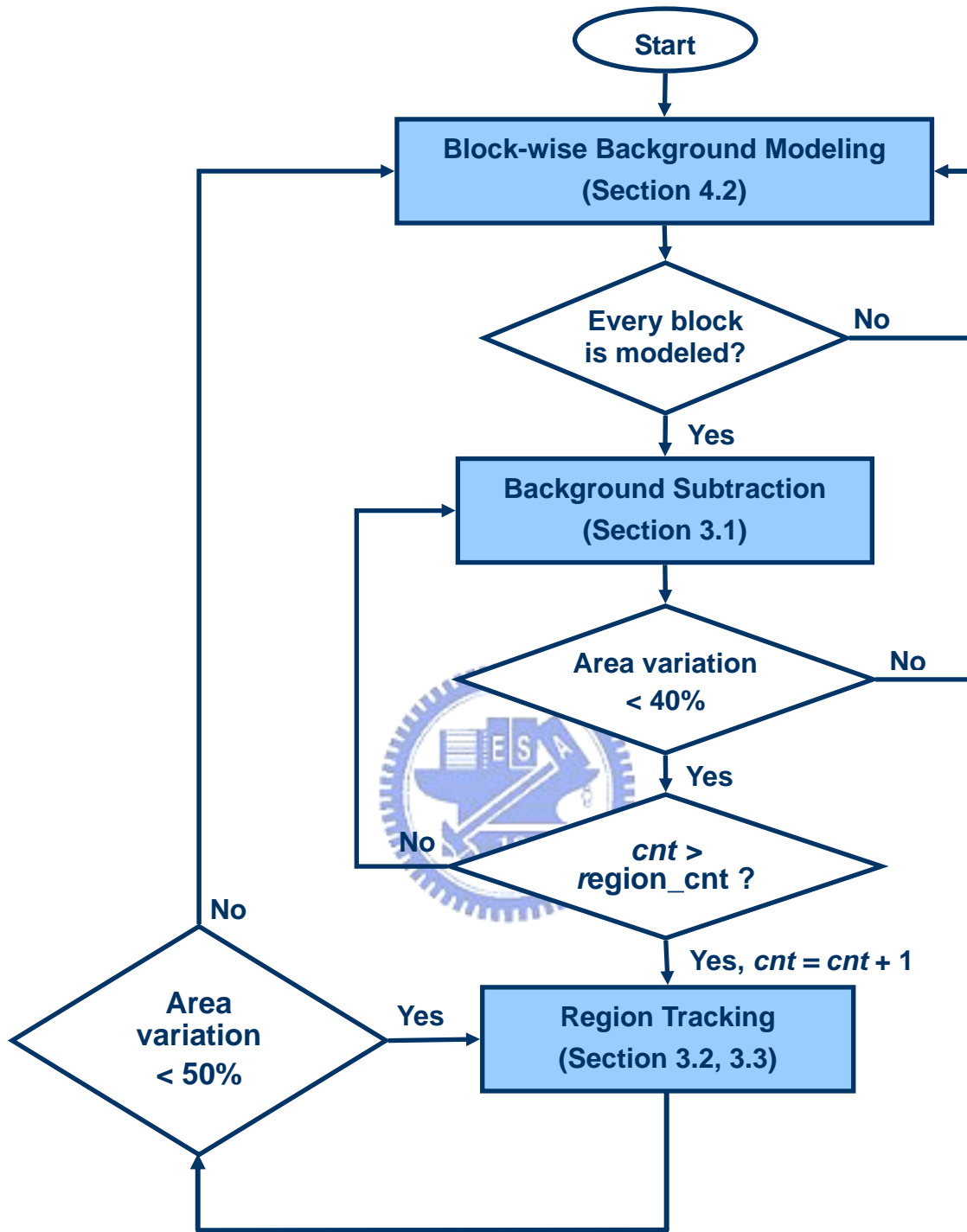
The camera is kept static at the beginning. Both background and moving objects may appear in the camera. The background information must be accumulated for a period of time. Once the background is built, the background subtraction method can be used. The motion region is detected for several frames to ensure reliability of the background information. If the area of the motion region is not stable, then the background must be remodeled again. The system enters the “region tracking” state when the motion region is defined. The area of the tracked region may unreasonably shirk or expand because of the imperfection of the region tracking process. If the area

changes more than 50% of the original region, the systems restarts the background modeling again.



**Figure 4-15** A simplified version of Figure 1-1. Every camera contains “background modeling”, “background subtraction” and “region tracking” processes.





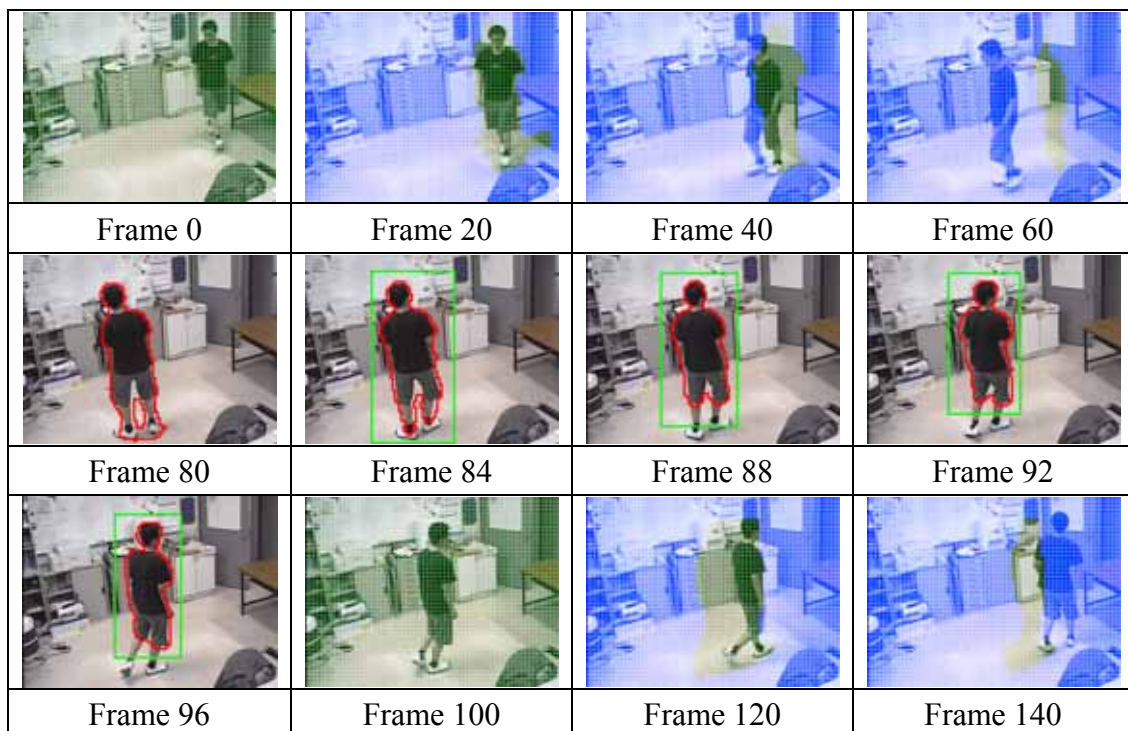
**Figure 4-16** Flow chart of a single camera's system. This figure shows the details about how to control the process of every camera in Figure 4-15.

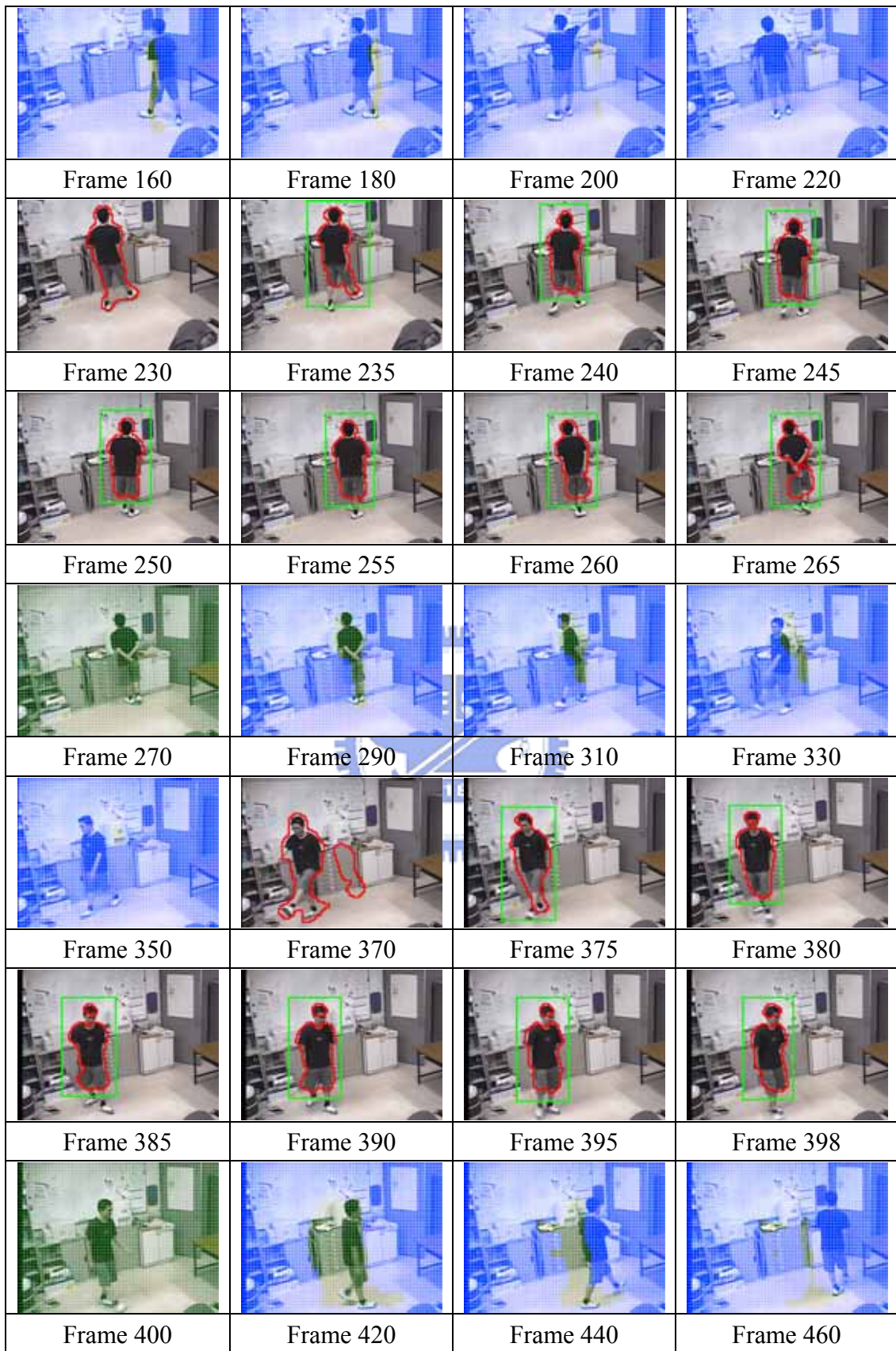
Figure 4-17 is the simulation result of Figure 4-16. The images with green grids and blue masks are in the background modeling state. When the background is modeled in all blocks, a background subtraction result like Frame 80 is obtained. Then the frames with a green rectangle represent the region tracking results. The camera changes its pan and tilt angles to locate the moving object in the center of the image. If the area of the tracked region varies too much, as shown in Frame 96 or Frame 265, the system restarts the background modeling process.





























In Frame 370, two motion regions are detected. This is because the person had stayed in front of the white board for a long enough time. Some part of the person is modeled as background. The system only tracks the bigger region.

From Frame 640 to Frame 680, the person stands at the same position for a long time. The person is modeled as a part of the background. No motion region is detected and the camera remains static. The new motion region is defined when the person turns around. The system tracks the new motion region and locates it in the center of the image.

These  $480 \times 320$  images are actually extracted from a sequence of  $640 \times 480$  images, which are acquired by a static camera, to simulate the images acquired by a moving camera. The system is simulated by using MATLAB on a Pentium IV 3.0 GHz computer. The required computational time is plotted in Figure 4-18.

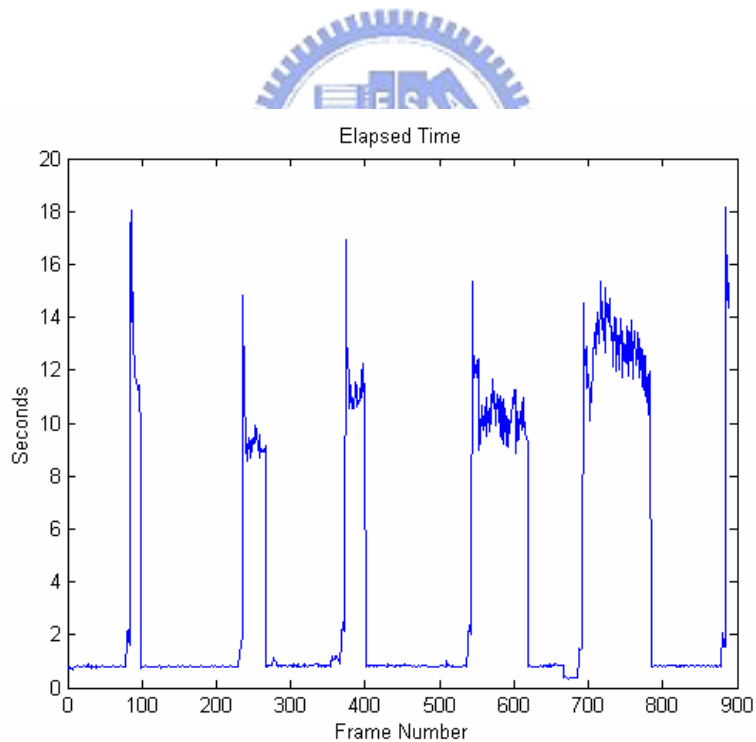




			
Frame 480	Frame 500	Frame 520	Frame 540
			
Frame 545	Frame 553	Frame 561	Frame 570
			
Frame 578	Frame 586	Frame 594	Frame 602
			
Frame 610	Frame 618	Frame 620	Frame 640
			
Frame 665	Frame 670	Frame 680	Frame 690
			
Frame 700	Frame 710	Frame 720	Frame 730
			
Frame 740	Frame 750	Frame 760	Frame 770



**Figure 4-17** The simulation result of Figure 4-16. The images with green grid and blue mask are in the background modeling status. When the background is modeled in all blocks, a background subtraction result is obtained. The frames with a green rectangle represent the tracking results.



**Figure 4-18** The consuming time of processing the sequence in Figure 4-17. The frames with small time consuming are in the status of background modeling or background subtraction. The frames with a relatively large time consuming are in the status of region tracking.



# Chapter 5. Conclusions

In this thesis, different active contour models are applied to dealing with the motion detection and region tracking problems. The information of the moving objects is obtained and classified based on the “active contour without edges” model. Some previous works about contour modeling is introduced in Chapter 2. In this thesis, contour models are all implemented based on the level set theory.

When the background is static, the motion region can be obtained by subtracting the background. The absolute difference data can be classified by the “active contour without edges” model. In order to locate the moving objects in the center of the image, the camera must change its pan and tilt angles. As the camera moves, the background subtraction can no longer be used. In this thesis, we adopt the region tracking model to identify moving objects.

The region tracking model is used to find in the current image the region which is similar to the region defined in the previous frame. The original region tracking model has some problems when the color of the moving object is very similar to that of the surrounding clutter. According to the level surface constructed from the previous frame, a reliability weight is introduced to suppress erroneous estimations. This is because it is reasonable to assume that the unknown level surface in the new frame should be very similar to that in the previous frame.

The statistical property is also taken into consideration in this thesis. A method that can estimate the prior probability is proposed for the maximum a posteriori (MAP) estimation. The probability model can strengthen the information of the inside region and eliminate the regions which should be outside.

A background modeling method is proposed and the static region is accumulated. The motion region is extracted by collecting the inter-frame difference results. A simple model is proposed to update the level surface.

A surveillance system is built based on background modeling, motion detection, and region tracking. The system restarts the background modeling if the background model is not perfect or if the area of the tracked region varies too dramatically. Because the region tracking process does not need the background information, the camera can be moved to always locate the moving objects in the center of the image.

In order to reduce the computational cost, a low-resolution level surface is used in this thesis. That is, the input image data is smoothed and down sampled before being processed. This results in problems in thin regions, such as the neck and limbs of a person. This is because the magnitude of the level surface in these thin regions is too small. To deal with this problem, the input image can be interpolated to increase the areas of thin regions. Then, we may use an associated high-resolution level surface to accomplish the region tracking task. The evolution will become more stable if the maximum magnitude of the level surface becomes larger inside the objects.



# References

- [1] Kass, M., Witkin, A., and Terzopoulos, D. “*Snakes: Active contour models.*” International Journal of Computer Vision, 1:321–331. 1988.
- [2] V. Caselles, R. Kimmel, and G. Sapiro, “*Geodesic Active Contours,*” International Journal of Computer Vision 22(1), 61-79, 1997
- [3] Dubrovin, B.A., Fomenko, A.T., and Novikov, S.P. “*Modern Geometry — Methods and Applications I,*” Springer-Verlag: New York.1984.
- [4] Gilles Aubert, Pierre Kornprobst. “*Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations,*” Springer 2001.
- [5] Tony F. Chan, Luminita A. Vese, “*Active Contours Without Edges,*” IEEE Transactions on Image Processing, Vol. 10, No. 2, pp.266-277, February 2001.
- [6] Song Chun Zhu and Alan Yuille, “*Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation,*” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 9, pp.884-900, September 1996.
- [7] Nikos Paragios and Rachid Deriche, “*Coupled Geodesic Active Regions for Image Segmentation,*” European Conference in Computer Vision, Nov. 1999.
- [8] Abdol-Reza Mansouri, “*Region Tracking via Level Set PDEs without Motion Computation,*” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No.7, 947-961, July 2002.
- [9] Alper Yilmaz, Xin Li, Mubarak Shah, “*Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras,*” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 11, 1531-1536, November 2004.
- [10] Alper Yilmaz, Xin Li, Mubarak Shah, “*Object Contour Tracking Using Level Sets,*” Proceedings of Asian Conf. on Computer Vision (ACCV), South Korea, 2004.
- [11] Osher, S. and Sethian, J.A., “*Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations,*” Journal of Computational Physics, 79, pp.12-49, (1988)
- [12] Stanley Osher, Ronald Fedkiw, “*Level Set Methods and Dynamic Implicit*

*Surfaces*,” Sprigner, 2003.

- [13] Sethian, James Albert., “**Level Set Methods and Fast Marching Methods, Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science**,” 2<sup>nd</sup> ed., Cambridge University Press, 1999. New York.
- [14] M.P. Wand and M. Jones, “**Kernel Smoothing**,” Chapman and Hall, 1995
- [15] Yizong Cheng, “**Mean Shift, Mode Seeking, and Clustering**,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.17, No. 8, pp.790-799, August 1995.
- [16] Dorin Comaniciu, Peter Meer, “**Mean Shift: A Robust Approach toward Feature Space Analysis**,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 5, pp.1-18, May 2002.
- [17] Dorin Comaniciu, Visvanathan Ramesh, Peter Meer, “**Kernel-Based Object Tracking**,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.25, No.5, pp.564-577, May 2003.
- [18] Ahmed Elgammal, Ramani Duraiswami, David Harwood, Larry S. Davis, “**Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance**,” Proceedings of the IEEE, Vol. 90, No.7, 1151-1163, July 2002.
- [19] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and .Wixson, “**A system for video surveillance and monitoring**,” Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep., MU-RI-TR-00-12, 2000.
- [20] Weiming Hu, Tieniu Tan, Liang Wang, Steve Maybank, “**A Survey on Visual Surveillance of Object Motion and Behaviors**,” IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews, Vol.34, No.3, pp.334-351, August 2004.
- [21] Ying-li Tian, Arun Hampapur, “**Robust Salient Motion Detection with Complex Background for Real-time Video Surveillance**,” IEEE Computer Society Workshop on Motion and Video Computing, Breckenridge, Colorado, January 5 and 6, 2005
- [22] Micheal Isard and Andrew Blake, “**CONDENSATION Conditional Density Propagation for Visual Tracking**,” International Journal of Computer Vision

29(1), pp.5-28, 1998.

- [23] Nikos Paragios, Rachid Deriche, "***Geodesic active Contours and Level Sets for the Detection and Tracking of Moving Objects,***" IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 3, pp.266-280, March 2000.
- [24] Nikos K. Paragios, "***Geodesic Active Regions and Level Set Methods: Contributions and Applications in Artificial Vision,***" PhD Thesis in University of Nice Sophia Antipolis School of Computer Engineering, January 2000.
- [25] Rosario Feghali, Amar Mitiche, "***Spatiotemporal Motion Boundary Detection and Motion Boundary Velocity Estimation for Tracking Moving Objects With a Moving Camera: A Level Sets PDEs Approach With Concurrent Camera Motion Compensation,***" IEEE Transactions on Image Processing, Vol. 13, No. 11, pp.1473-1490, November 2004.
- [26] Roman Goldenberg, Ron Kimmel, Ehud Rivlin, Michael Rudzsky, "***Fast Geodesic Active Contours,***" IEEE Transactions on Image Processing, Vol. 10, No. 10, pp.1467-1475, October 2001.
- [27] Yonggang Shi, William Clem Karl, "***A Fast Level Set without Solving PDEs,***" Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on, Vol. 2, pp.97- 100, March 2005.
- [28] Yonggang Shi, William Clem Karl, "***Real-time Tracking Using Level Sets,***" International Conference on Computer Vision and Pattern Recognition (CVPR) June 2005.



## 作者簡歷

蔡孟修，西元 1982 年 6 月 11 日出生於台中縣。西元 2000 年畢業於國立台中第二高級中學，西元 2004 年畢業於國立交通大學電機與控制工程學系。之後進入國立交通大學電子研究所攻讀碩士學位。碩士論文題目為「基於等位函數法之運動物體偵測與追蹤」。

聯絡地址：台中市西屯區大聖街 455 號

電子信箱：[manshow@archer.ee.nctu.edu.tw](mailto:manshow@archer.ee.nctu.edu.tw)

