國立交通大學

資訊工程學系

碩士論文

利用等位函數法結合形狀資訊做影像序列的物體追蹤

Object Tracking via the Level Set Method Integrated with Prior Shape

研究生：徐錦祥

指導教授：陳稔 博士

中華民國九十四年七月

利用等位函數法結合形狀資訊做影像序列的物體追踪

Object Tracking via the Level Set Method Integrated with Prior Shape

研究生：徐錦祥　　Student：Chin-Hsiang Hsu

指導教授：陳　稔　　Advisor：Zen Chen

國立交通大學

資訊工程學系

碩士論文

A Thesis
Submitted to Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Master
in Computer Science and Information Engineering

July 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年七月

利用等位函數法結合形狀資訊做影像序列的物體追踪

學生：徐錦祥　　　　　　　　　　　指導教授：陳　稔 博士

國立交通大學資訊工程學系

摘要

物體追踪在電腦視覺、監測系統、交通監測站等，都有很大的應用，並且提供了高階影像處理的前置工作。本論文提出一個對影像序列的物體輪廓線追踪方法。主要是結合物體形狀資訊來改進 Abdol-Reza Mansouri 的方法。

在 Abdol-Reza 的方法中，追踪演算法主要是透過最小化一個由貝氏定理所推導出的能量函數，來變動所要用來追踪的輪廓線，使其往目標物逼進。此方法有三個優點。第一，不需要做物體運動估測。第二，物體允許任意變形，包括拓樸的改變。第三，物體的輪廓梯度不需要很大。而它所面臨的缺點是，當物體的附近有跟其相近的背景時，追踪的結果將會受到很大的影響。我們希望透過結合物體的形狀資訊來改進這個缺點。當被追踪物體有部分跟背景相似時，明確的部分由原來的方法追踪，並將其結果與形狀資訊對齊，接著由形狀資訊將不明確的部分補足。如果物體本身就整個能與背景分開時，形狀資訊的權重將會自動被付予很小的值，以保有物體允許隨意變形的特性。這樣的整合方式，也將使我們的方法更具有彈性。

Object Tracking via the Level Set Method Integrated with Prior Shape

student：Chin-Hsiang Hsu Advisor：Dr. Zen Chen

Department of Computer Science and Information Engineering
National Chiao Tung University

Abstract

Detecting and tracking moving objects has a wide variety of applications in computer vision such as computer vision, video surveillance, traffic monitoring, etc. Additionally, it provides input to higher level vision tasks. This thesis presents an approach to tracking a moving object over a sequence of images. In particular, we improve the Abdol-Reza's model by coupling with shape prior knowledge for shape perseverance in case of ambiguity.

In the model of Abdol-Reza, tracking is achieved by evolving the contour from frame to frame by minimizing an energy functional evaluated by Bayesian theory. There are three two favorable features in this model. First, no motion field or parameters needed to be computed. Second, deformable shapes of the object are allowed and the topology of the boundary is not constrained. Third, no assumption is made on the strength of the edge gradient. However, it also suffers from the constraints imposed on a degree of dissimilarity between the object and the background. A background region similar to the object might corrupt the contour evolution. We want to overcome this drawback by coupling with a shape prior in the associated energy functional. When the object is partially involved in a similar background, the original tracking term in the functional will dominate the result in the unambiguous background part, and the prior shape will guide the object movement in the ambiguous part. If the object is entirely distinguishable from the background, the weight of the shape prior is set low and thus allowing free deformation of the object. Compared to other tracking methods embedded with shape priors, the presented approach is more flexible, retaining the advantage of suffering little constraints on the deformable shape of the tracked object in many cases.

誌　　謝

首先要感謝　陳　稔教授兩年來的悉心教導，這段期間培養出我獨立思考的能力，在學習態度、研究方法上也得到長足進步。每每在研究過程中遭遇困難時，老師總能積極給予幫助。因為有來自老師的援助，這篇論文才得以順利完成，在此向老師獻上無盡的謝意。

其次我要感謝許志宇學長在這期間曾經給我的指導。您犧牲晚上及假日休閒的時間給我細心的建議及教導，使我得以一窺等位函數法的堂奧，並在研究方法上有所突破。這篇論文的完成，您真的給我很大的幫助，在此我要由衷的感激您!

感謝實驗室全體的成員，方彥翔學長，顧正偉同學、朱佑吾同學、紀文原同學、蘇永迪同學、陳柏聰學弟、鄭龍凱學弟、吳昶浩學弟。您們都曾在研究期間給予意見，並給予我許許多多的幫助，謝謝你們陪我走過這一段日子。

感謝我的父母親及親愛的家人，你們無怨無悔的支持與付出，讓我能全心全意關注我的研究，在生活上無後顧之憂。你們所給我精神上的支持與鼓勵，這一切都是使我能義無反顧向前邁進的動力。

感謝我的好朋友建勳。在遭遇困難時，我們彼此互相鼓勵，這是促使我繼續努力的動力。

當然我還要感謝其他許許多多的朋友。紙短情深，在此無法一一致謝，僅以本篇論文獻給你們，再次說聲謝謝！

徐錦祥　於交大

July, 2004

Contents

摘要.....i

Abstract.....ii
誌　　謝.....iv
Contents.....v
List of Figures.....vi
1. Introduction.....1
1.1. Motivation.....1
1.2. Object Tracking.....1
1.3. Thesis Organization.....3
2. Related Work.....4
2.1. Active Contour Models.....4
2.2. Shape Priors.....7
3. Overview of the Proposed Approach.....9
4. The Tracking Method.....13
4.1. Abdol-Reza Mansouri's Model.....13
4.2. Level Set Representation.....16
4.3. Problems of the Abdol-Reza Mansouri's Model.....17
4.4. Implementation Issues of the Level Set Method.....20
5. Shape Prior.....26
5.1. Geodesic Active Contour Model.....26
5.2. An Euclidean Invariant Shape Prior.....31
6. Experiments.....39
6.1. Synthetic Image Sequences.....39
6.2. Digital Camera Photos.....46
6.3. Real Image Sequences.....49
6.4. Parameter Settings.....56
Weight of Shape Prior.....56
The Curvature Term.....67
7. Conclusions and Future Work.....75
7.1. Conclusions.....75
7.2. Future Work.....75
References.....76

List of Figures

Fig 4.1 Idea of Abdol-Reza’s model 13

Fig 4.2 Discriminate analysis.. 15

Fig 4.3 Level set representation 17

Fig 4.4 The camouflage synthetic sequence 19

Fig 4.5 Problem of Abdol-Reza’s model.. 20

Fig 4.6 Approximation of $F\Delta t$ 21

Fig 4.7 Numerical Problem... 22

Fig 4.8 Bubble effect.. 23

Fig 5.1 Image energy and Shape energy 27

Fig 5.2 Gradient vector 28

Fig 5.3 Shape and its forces field.. 29

Fig 5.4 Shape prior illustration 1 33

Fig 5.5.Shape prior illustration 2 34

Fig 6.1 Abdol-Reza’s model 40

Fig 6.2 Abdol-Reza’s model + shape prior 40

Fig 6.3 Abdol-Reza’s model 41

Fig 6.4 Abdol-Reza’s model + shape prior 41

Fig 6.5 Abdol-Reza’s model 42

Fig 6.6 Abdol-Reza’s model + Shape prior... 42

Fig 6.7 Abdol-Reza’s model. 43

Fig 6.8 Abdol-Reza’s model + Shape prior. 43

Fig 6.9 Abdol-Reza’s model. 44

Fig 6.10 Abdol-Reza’s model + Shape prior. 44

Fig 6.11 Abdol-Reza’s model... 44

Fig 6.12 Abdol-Reza’s model + Shape prior. 44

Fig 6.13 Abdol-Reza’s model + Shape prior + Background removal..........45

Fig 6.14. Abdol-Reza’s model.on digital camera photos 46

Fig 6.15. Abdol-Reza’s model + shape prior on digital camera photos.......47

Fig 6.16 Mechanism of shape prior 48

Fig 6.17 Abdol-Reza’s model 49

Fig 6.18 Abdol-Reza’s model + shape prior 50

Fig 6.19 Two hands, with shape prior model.. 51

Fig 6.20 Two hands, Abdol-Reza’s model 52

Fig 6.21 A bus with complex appearance 53

Fig 6.22 A bus 54

Fig 6.23 Human 55
Fig 6.24 Initial contour 57
Fig 6.25 $\alpha = 500$ 58
Fig 6.26 $\alpha = 1000$ 59
Fig 6.27 $\alpha = 1500$ 61
Fig 6.28 $\alpha = 2000$ 62
Fig 6.29 $\alpha = 2500$ 63
Fig 6.30 $\alpha = 3000$ 64
Fig 6.31 $\alpha = 3500$ 65
Fig 6.32 $\alpha = 4000$ 66
Fig 6.33 $\lambda =0.01$ 68
Fig 6.34 $\lambda =0.03$ 69
Fig 6.35 $\lambda =0.05$ 70
Fig 6.36 $\lambda =0.07$ 71
Fig 6.37 $\lambda =0.09$ 72
Fig 6.38 $\lambda =0.11$ 73
Fig 6.39 $\lambda =0.13$ 74

1. Introduction

1.1. Motivation

High level vision tasks for video processing require tracking of the complete contour of the objects, such as in the applications of computer vision such as computer vision, video surveillance, traffic monitoring, etc.

1.2. Object Tracking

Numerous approaches for tracking objects in an image sequence are proposed and can be mainly classified in three categories:

1. Correspondence-based object tracking: Tracking is performed by establishing correspondence of the objects in consecutive frames. These approaches rely on the detection of temporal changes and employ a thresholding technique over the inter-frame difference. These methods can only be applied to images with static backgrounds and they provide coarse object silhouettes.

2. Motion-based object tracking: Tracking is performed by estimating the motion of objects in consecutive frames. Objects are represented by planar surfaces, such as rectangle and ellipse, or their centroids. These methods are relatively fast but have considerable difficulties in dealing with non-rigid movements and objects.

3. Model-based object tracking: Object representation includes rigid models and non-rigid models, or deformable templates. Such models usually have a number of parameters to control the shape and pose of the model. These methods suffer from high computational costs for complex models due to the need for coping

with scaling, translation, rotation and deformation.

In correspondence-based object tracking, background subtraction is the most popular detection method used in object trackers, where color observations of individual pixels in a reference frame are statistically modeled. Detection is performed by labeling the pixels that deviate from the static model.

In motion-based tracking, a statistical analysis is performed and is used to provide the motion-based estimation. Motion models used are translation, scaling and affine motion models. One of the most common motion based tracker is “template matching”, where translation of an object template is computed by searching the image for a similar template. Additionally, by assuming a smooth background, the input frame can be used directly to provide an accurate object tracking result. In [8], the authors proposed a kernel-based tracker in which color priors are computed using weighted kernel density estimation. Mean-shift vector is then computed iteratively by maximizing likelihood between the object color prior and the model generated from hypothesized object position.

There is a substantial use of flexible models or deformable templates in model-based tracking. There are three broad classes of these models.

(i) Articulated models: Articulated models are built up from a number of rigid components connected by sliding or rotating joints. This approach is only applicable to a restricted class of variable shape problems.

(ii) Statistical models of shape: The shape is represented by a set of boundary points connected by arcs with a statistical model of relationships between them, or a set of points with distributions related by a covariance matrix.

(iii) Active contour models: There are two general types of active contour models in the literature today: *parametric active contours (snakes)* [1] and *geometric active contours* [2][4]. Parametric active contours, or snakes, can be considered as parameterized models, or the parameters being spline control points. The idea of fitting is minimization of an energy functional to apply forces to the model. In recent years, it is popular to represent parametric active contour by geometric active contours such as one of the level sets in higher dimensional space.

This thesis presents an approach for contour tracking formulated as a calculus of variations problem. The proposed energy functional contains two energy terms, the image energy *Er* and the shape energy *Es*. Image energy, which is based on a Bayesian framework [17], performs discriminate analysis on pixels. The shape energy, motivated from the geodesic active contour model [2], is weighted by confidence of the decrease of the image energy and resolve discriminate uncertainties. Tracking is achieved by evolving the contour, which is represented using level sets, to a position in the gradient descent direction of the energy functional. Also, invariance to transformations of shape energy is achieved by minimize a pseudo distance [22] between the evolving contour and the shape model.

1.3. Thesis Organization

This thesis is organized as follows. The following chapter contains a review of related work. Chapter 3 describes the problem investigated in this thesis. In Chapter 4, we introduce Abdol-Reza's model for tracking and present its drawback. Level set implementation issues are also briefly discussed. Then the shape prior solution is described in chapter 5, together with extension of our method. Finally, the thesis ends with several experimental results in chapter 6 followed by conclusions in chapter 7.

2. Related Work

2.1. Active Contour Models

Since active contour was introduced to the vision community by Kass et al. (1988), extensive researches was done on “snakes” or parametric active contour models for boundary detection. The classical approach is based on deforming an initial contour towards the boundary of the object to be detected. The deformation is obtained by trying to minimize a functional designed such that its minimum is obtained at the boundary of the object. The energy functional is basically composed of two components, one controls the smoothness of the curve and another attracts the curve towards the boundary. However, there are three key difficulties with parametric active contour algorithms. First, the initial contour must be close to the true boundary or it will likely converge to the wrong result. Then, active contours have difficulties progressing into *boundary concavities.* Finally, energy model is not capable of handling changes in the topology of the evolving contour when direct implementations are performed. An approach insensitive to initialization and the ability to move into boundary concavities is proposed in [1]. The author present external forces originate from an edge map of the image to provide larger capture range. However, this parametric model cannot handles topology changes as well.

Recently, novel geometric models of active contours were proposed [2] [4]. These models are based on the theory of curve evolution and geometric flows, which has received a large amount of attention in recent years. It allows automatic changes in the topology when implemented using the level-sets based numerical algorithm

[6][7]. Thereby, several objects can be detected simultaneously without previous knowledge of their number in the scene and without using special tracking procedures.

However, because the flow may be slow to converge in practice, a constant term is added to keep the curve moving in the desired direction. Kaleem et al. [10] modify this term based on the gradient flow derived from a weighted area functional, with image dependent weighting factor. Since this flow requires the computation of only first order derivatives, it offers significant computational savings over the weighted length minimizing flow.

Active contour models that rely on the edge-detector or image gradient can detect only objects with edges defined by gradient. In practice, the discrete gradients are bounded and then the stopping function is never zero on the edges, and the curve may pass through the boundary. Chan and Vese detailed a level set implementation of the Mumford-Shah functional [9], which is based on the use of the Heaviside function as an indicator function for the separate phases. The idea is to partition the given image into two homogeneous regions, without a stopping edge-detector. The authors also extend this binary image oriented method to segment images with more than two regions by multiphase level sets [12].

To segment objects in textured background, Paragios and Deriche proposed a region-based energy, where statistical models were used for textured object and background regions [16]. They extended the region model to the mixture of Gaussians for magnitude of Gabor filter responses. The texture segmentation is obtained by unifying region and boundary-based information.

Tracking is another segmentation method by using the segmentation results in the image frames history. If the contour is initialized with its previous position, contour segmentation approaches become object trackers, and tracking is defined based on motion information to evolve an initial object contour[4][14][17][18][19].

In [4], the evolution equation for contour is obtained by image differences, this tracking algorithm can be applied only when an important degree of similarity among the images and displacements involved are small. G. Tsechpenakis et al. proposed a method handling the appearance of occlusions between different objects [19]. The use of the object motion history and statistical measurements provide information for the extraction of uncertainty regions. In [14], tracking is expressed as detection and tracking of moving objects in image sequences. In the proposed algorithm, a detection step forces a closed curve to converge towards moving areas of an image, while a tracking step evolves the curve to coincide with the exact boundary of the moving object. The tracking step is only an intensity boundary detection algorithm using active contours and implemented using level sets. Since the tracking step relies on the previous frame, the background is assumed to be stationary. The problem addressed in [17] is that object tracking can be treated as two-class discriminate analysis of pixels, where the classes correspond to the object and the background regions. Since his approach compute for each pixel by brute-force search in a circular neighborhood, there are two problems exist even when strong assumption on intensity boundaries. First, the contour cannot capture the parts of the object near where existing a background region with similar intensities to them. The second, the background around the contour will be classified as the object if there exist some pixels with similar intensities within the object. The classification criterion is extended in [18]. A window of specified fixed size is defined for each pixel around the contour. The contour will move in the direction that can equalize the numbers of pixels within the window that belong to two classes (object and background). Also, shape priors are used to recover the missing object regions during occlusion. Nevertheless, since the shape prior takes effect only when the occlusions are detected, this approach still suffers the problems as encountered by [17].

Our approach is the incorporation of knowledge about the shape with [2], to control the difficult conditions of the image. However, different from[18], we improve the tracking results of [17] instead of handling occlusions of the objects.

2.2. Shape Priors

In the substantial literature of deformable models, there are three main mechanisms can be found to constraint the shape of the curve during the evolution of the deformable model:

1. Free-form approaches: These methods do not encode a default shape, but the energy functional imposes smoothness and compactness of the boundary of the surface. They can be seen as general, weak and local shape constraints.
2. Analytical parametric templates: The analytical shape constraints are defined by the distribution of the admissible parameters. These methods are commonly used when some prior information about the geometrical shape is available, which can be encoded using a small number of parameters.
3. Prototype-based constraints: Shapes are represented by the mean shape of a collection of individuals and their statistical variations. These methods require either training or global shape modeling.

T. F. Cootes et al. propose a method that uses point distribution model (PDM) [13] as the prototype-based constraints. It describes the average and characteristic shape variations of a set of training samples, which are given in the form of a set of points on the learning boundaries. In [21], the authors investigate the use of discrete cosine transform (DCT) coefficients in describing object shape. The method starts with local shape parameterization, then, the shape is converted into an implicit representation using global shape parameters. As can be seen, incorporating prior shape information in a deformable model, requires either training or global shape modeling. Training involves manual interaction to accumulate information on the shape variability of the

same object class. Global modeling can be characterized using only a few parameters, and tend to be much more stable than local properties. The choice of a certain shape representation determines to a great extent the flexibility, processing speed, and amount of user interaction.

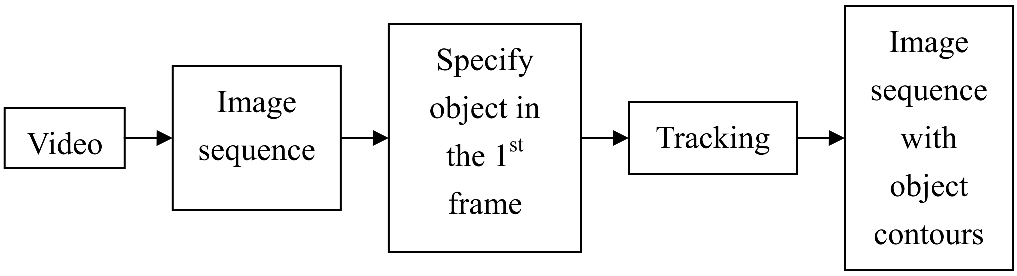
Leventon et al. [23] have incorporated statistical shape information into the evolution of geodesic active contours. They compute a prior on shape variation given a set of training instances. Each curve in the training dataset is embedded as the zero level-set of three-dimensional surface, which is a signed distance function. Daniel Cremers et al. [22] propose a closed-form, spline based solution for incorporating invariance with respect to similarity transformations in the variational framework. Dainiel Cremers and Stefano Soatto [20] integrate prior shape knowledge into level set based segmentation methods and proposed dissimilarity measures for shapes encoded by the signed distance function.

The proposed shape prior is a global shape model using the initial contour as the shape prior but not the training set from the image history, therefore the number of parameters involved can be reduced. It is motivated by modeling the flow field of the shape forces as geodesic active contours [23], incorporating invariant transformation by the pseudo distance measure [22] and alternatively computing the total energy and the shape energy during the evolution of the curve.

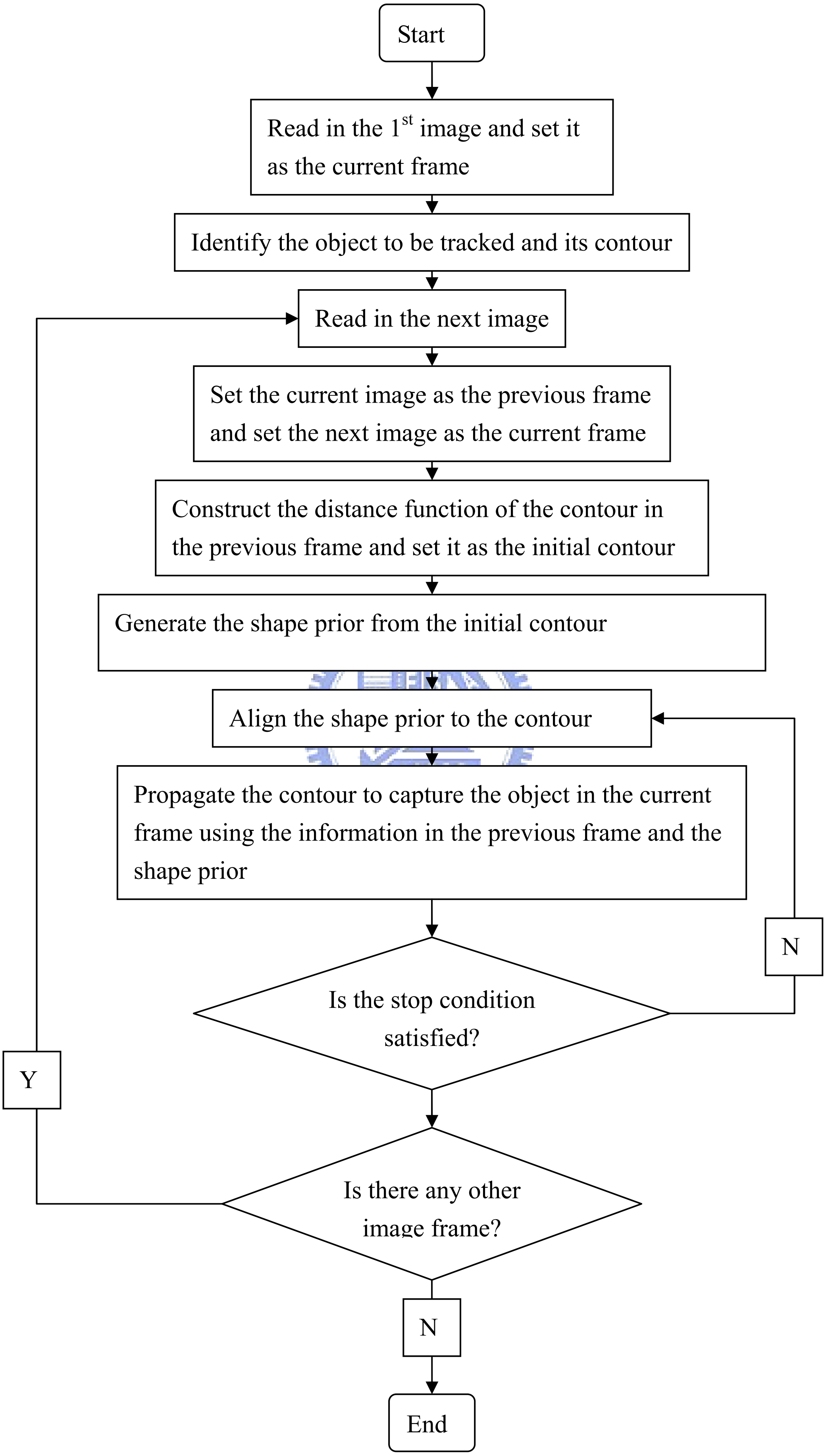
3. Overview of the Proposed Approach

Given an image sequence with a specified object and its boundary in the first frame, the thesis is to identify the boundary in all the image sequence. We use level set model to represent the curve that used to capture the object boundary. Two favorable features of level set method are its automatically handling of topology changes and easy implementation. The formula to propagate the curve is an Euler-Lagrange equation derived from an energy functional. One of two terms in the energy functional is related with the image information and the other is for shape preserving. To solve the Euler-Lagrange equation means to minimize the energy functional and to move the curve towards the object boundary. The idea is to classify each pixel in the image as the member of the object or that of the background. If the classification is uncertain, the shape preserving term will attempt to preserve the shape of the curve and will lead the decision.

The tracking system is presented in the following flow chart:


Video
Image sequence
Specify object in the 1st frame
Tracking
Image sequence with object contours

The detailed algorithm of the presented approach is showed in the following flow chart:


Start
Read in the 1st image and set it as the current frame
Identify the object to be tracked and its contour
Read in the next image
Set the current image as the previous frame and set the next image as the current frame
Construct the distance function of the contour in the previous frame and set it as the initial contour
Generate the shape prior from the initial contour
Align the shape prior to the contour
Propagate the contour to capture the object in the current frame using the information in the previous frame and the shape prior
Is the stop condition satisfied?
N
Is there any other image frame?
Y
N
End

Our approach combines three models:

1. Abdol-Reza’s model.
2. Geodesic active contour model.
3. Pseudo distance measure.

4. The Tracking Method

This chapter takes an overview of Abdol-Reza's model. Then we point out its drawbacks in section 4.2. Finally, the implementation issues of level set method are also presented.

4.1. Abdol-Reza Mansouri's Model

Object tracking can be treated as two-class discriminate analysis of pixels, where the classes correspond to the object and the background regions. This is also the idea of Abdol-Reza's model. For each pixel **x** in the current frame $I^{n+1}(i,j)$, search for the most similar pixel **y** in a disk with $I^{n}(i,j)$ as its center in the previous frame. If **y** is a member of the object in the previous frame, then **x** is classified as the object, otherwise, **x** is belonging to the background.

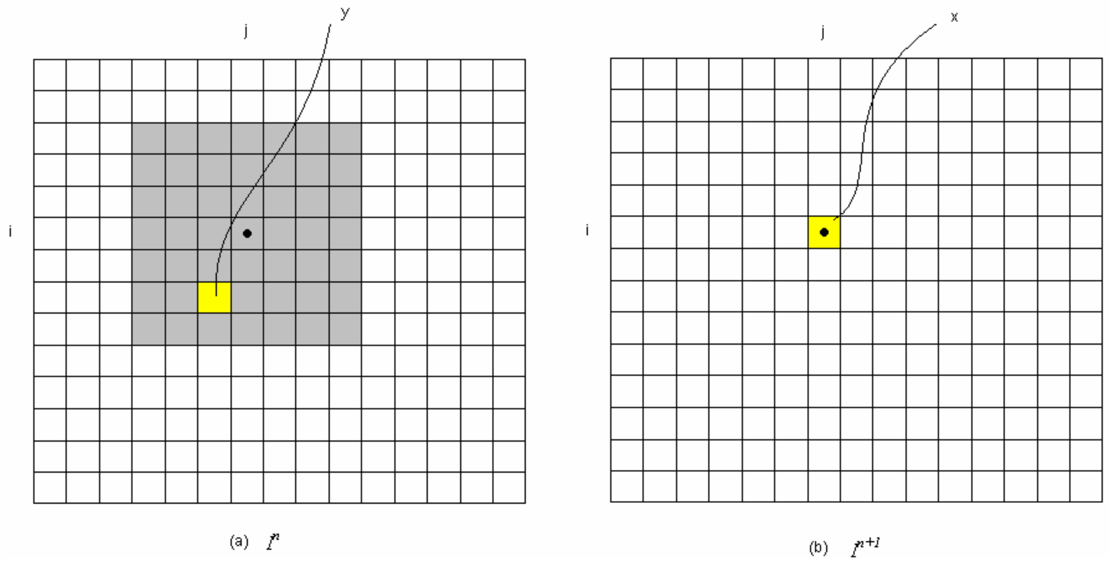


Fig 4.1 Idea of Abdol-Reza's model

The shadow part is the search window.

Why this classification method works properly in Abdol-Reza's model is because it uses level set representation for the contour model. Level set representation can divide pixels into two classes by discriminate them as either inside or outside the contour and processes only the pixels near the contour, neglecting those far away. Without this mechanism, the classification result might be a collection of discontinuous pixels, instead of an entire object.

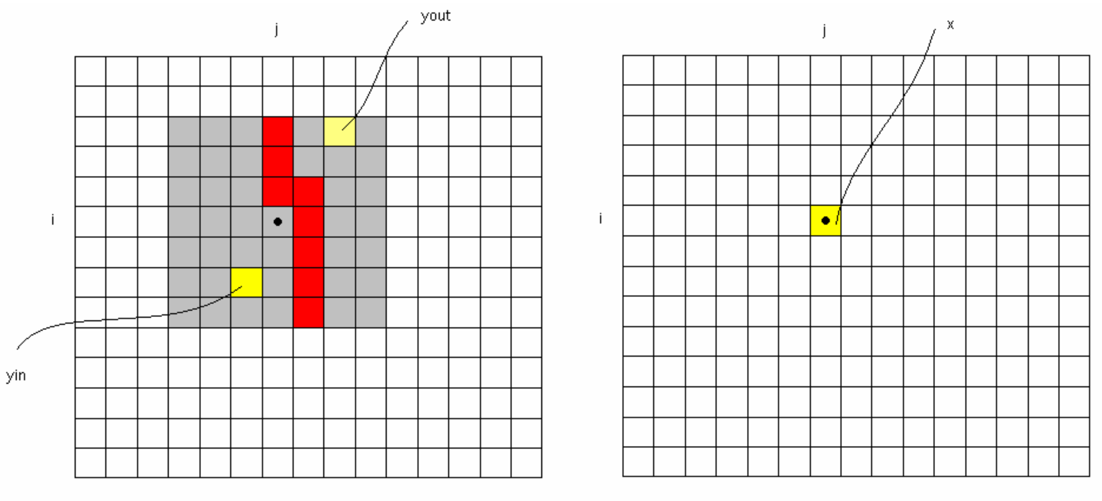
We now go through the approach of Abdol-Reza's model. Let I^k be a sequence of images with domain Ω (an open subset of R^2). Let $R_0 \subset \Omega$ be a region in the n-th image (I^n) and $R_1 \subset \Omega$ be the corresponding region in image I^{n+1} that we want to estimate. And let $\vec{\gamma} : [0,1] \to \Re^2, s \mapsto \vec{\gamma}(s)$ be a closed curve, oriented counterclockwise, that we estimate for the boundary ∂R_1 of R_1. Given I^n, I^{n+1} and R_0, the optimizing estimate R of R^1 is found by minimizing the energy functional

$$\begin{aligned} &E_r(\vec{\gamma} | I^n, I^{n+1}, R_0) = \\ &-\int_R \log P_{in,\mathbf{x}}(I^{n+1}(\mathbf{x}) | I^n, R_0) d\mathbf{x} - \int_{R^c} \log P_{out,\mathbf{x}}(I^{n+1}(\mathbf{x}) | I^n, R_0) d\mathbf{x} \qquad (4.1) \\ &+ \lambda \int_0^1 \left\| \frac{\partial \vec{\gamma}}{\partial s} \right\| ds \end{aligned}$$

The first two terms on the right hand side of the functional are the external forces introduced by the image information. The first term means, given I^n, R_0 and $\mathbf{x}$ belongs to R_1, the probability that $\mathbf{x}$ has the observing intensity I^{n+1} *(***x***)* . Since we have the prior of the object R_0 in the previous frame, the probability is high if I^{n+1} *(***x***)* is similar to the intensity of some *y* in R_0; and it is low if $\mathbf{x}$ is dissimilar to any pixel in R_0. In other hand, the second term means, given In, R_0 and $\mathbf{x}$ dose not belong to R_1, the probability that $\mathbf{x}$ has the observing intensity I^{n+1} *(***x***)*. The probability is high if I^{n+1} *(***x***)* is dissimilar to any pixel in R_0, otherwise, the probability is low. Using these two terms, we can determine the pixel around the contour should be included or excluded by the contour. According to subtraction of these two terms, the evolving speed could be either positive or negative which indicates shrinking or expanding of the contour at that point. The other two terms are the internal force associated with the contour and will cause the contour to be smoother. Finally, the definition of the probability functions will be detailed later.

In order to minimize (1) we search for the gradient descent direction of (4.1), which can be computed from its Euler-Lagrange equation:

$$\frac{d\vec{\gamma}(s)}{dt} = \left[\log P_{in,\vec{\gamma}}(I^{n+1}(\vec{\gamma}(s)) \,|\, I^n, R_0) - \log P_{out,\vec{\gamma}}(I^{n+1}(\vec{\gamma}(s)) \,|\, I^n, R_0)\right]\vec{n}(s) - \lambda\, \kappa_{\gamma}(s)\vec{n}(s) \qquad (4.2)$$



(a) I^n (b) I^{n+1}

Fig 4.2 Discriminate analysis

The red parts represent the contour. P_{in} is color difference between $\boldsymbol{x}$ and y_{in} and P_{out} is color difference between $\boldsymbol{x}$ and y_{out}. The subtraction between P_{in} and P_{out} can determine $\boldsymbol{x}$ as which class.

Then, equation (4.2) can be solved numerically by discretizing the interval on which $\vec{\gamma}$ is defined, leading to an explicit representation of $\vec{\gamma}$. A better alternative is to represent the curve $\vec{\gamma}$ implicitly by the zero-level set of a function $u : \Re^2 \rightarrow \Re$.

Two favorable properties of the implicit level set representation over explicit contour representations are its independence of a particular parameterization, and the fact that the topology of the boundary is not constrained, such that merging and splitting of the contour during evolution is facilitated.

Since $\vec{\gamma}$ obeys an evolution equation and the zero-level set of *u* is assumed to coincide with $\vec{\gamma}$, *u* must evolve according to a certain evolution equation related to that of $\vec{\gamma}$. We can thus embed *u* in a one-parameter family and construct the evolution equation that the zero-level set of *u* satisfy the evolution equation of $\vec{\gamma}$.

4.2. Level Set Representation

If the evolution of $\vec{\gamma}$ is described by the equation

$$\frac{d\vec{\gamma}(s,t)}{dt} = F(\vec{\gamma}(s,t))\vec{n}(s,t)$$

where *F* is a function defined on $\Re^2$, the corresponding evolution of *u* is given by:

$$\frac{\partial u(\mathbf{x})}{\partial t} = \frac{\partial u}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial u}{\partial y}\frac{\partial y}{\partial t} = \begin{bmatrix} \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} \end{bmatrix}^T \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \end{bmatrix} = F(\mathbf{x})\vec{n}(\mathbf{x}) \cdot \nabla u(\mathbf{x}) = F(\mathbf{x})\left\|\nabla u(\mathbf{x},t)\right\|$$

Since $\vec{\gamma}(s,t)$ is a vector in $\Re^2$, it can be represented by a point $\boldsymbol{x}$ in the domain of *u*, and $F(\vec{\gamma}(s,t))$ thus can be replaced by *F(**x**)*, leading the evolution of $\vec{\gamma}$ to the above equation. The condition of this correspondence is that all the points of the curve $\vec{\gamma}(s,t)$ must be on the same level set which is the zero-level set (the set with *u = 0*) in most cases.

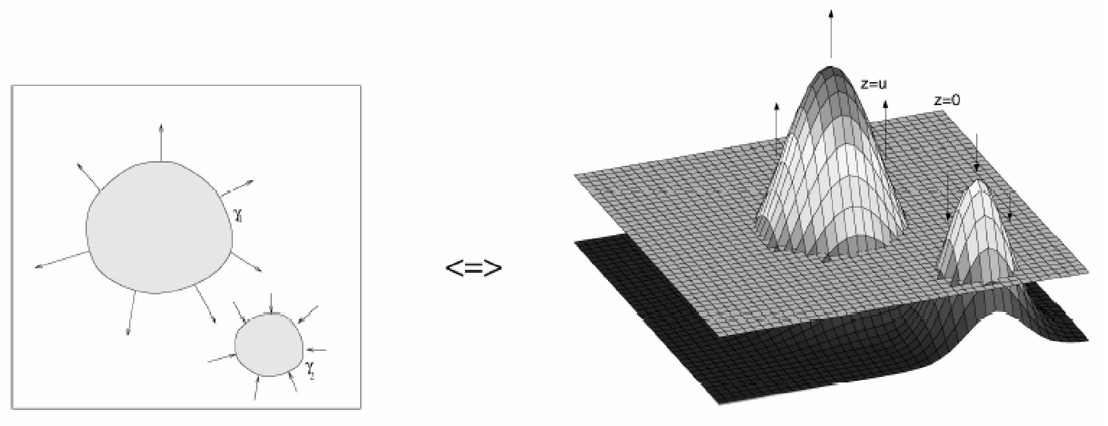


Fig 4.3 Level set representation

This figure is cut from [17]. The figure shows the equivalence between the evolution of curves $\vec{\gamma}_1, \vec{\gamma}_2$, and the evolution of function u.

Then, the level set evolution equation corresponding to the curve evolution (4.2) for tracking is given by:

$$\begin{aligned}\frac{\partial u(\boldsymbol{x})}{\partial t} &= \left[\log P_{in,\boldsymbol{x}}(I^{n+1}(\boldsymbol{x})\big|I^n, R_0) - \log P_{out,\boldsymbol{x}}(I^{n+1}(\boldsymbol{x})\big|I^n, R_0)\right]\|\nabla u\| \\ &- \lambda\kappa_u(s)\|\nabla u\|\end{aligned} \qquad (4.3)$$

where

$$-\log P_{in,\boldsymbol{x}}(I^{n+1}(\boldsymbol{x})\big|I^n, R_0) \approx \inf_{\{z:|z|\le\eta, \boldsymbol{x}+z\in R_0\}} (I^{n+1}(\boldsymbol{x}) - I^n(\boldsymbol{x}+z))^2$$

$$-\log P_{out,\boldsymbol{x}}(I^{n+1}(\boldsymbol{x})\big|I^n, R_0) \approx \inf_{\{z:|z|\le\eta, \boldsymbol{x}+z\in R_0^c\}} (I^{n+1}(\boldsymbol{x}) - I^n(\boldsymbol{x}+z))^2$$

$$\kappa_u = \nabla\cdot\frac{\nabla u}{\|\nabla u\|} = \frac{u_{xx}u_y^2 - 2u_xu_yu_{xy} + u_{yy}u_x^2}{(u_x^2 + u_y^2)^{3/2}}$$

4.3. Problems of the Abdol-Reza Mansouri's Model

There are two problems in the model of Abdol-Reza. First, the contour cannot capture the parts of the object near which existing a background region with similar intensities to them. The second, the background around the contour will be classified as the object if there exist some pixels with similar intensities to them within the object. These two problems exist even in cases that objects in the images have strong intensity boundaries.

Fig 4.4 is derived from Abdol-Reza's work [17] and shows the limits of his approach. The image sequence is constructed by cutting out a disc-like shape from the center of the image and pasting it so as to create apparent motion from the lower left corner to the upper right corner of the image. The tracking is correct until in frame 4 of the sequence, the right side of the tracked region is strongly deformed. This is due to the fact that the region and background textures are so similar there that the probability estimates *Pin* and *Pout* are almost identical for most of those pixels.

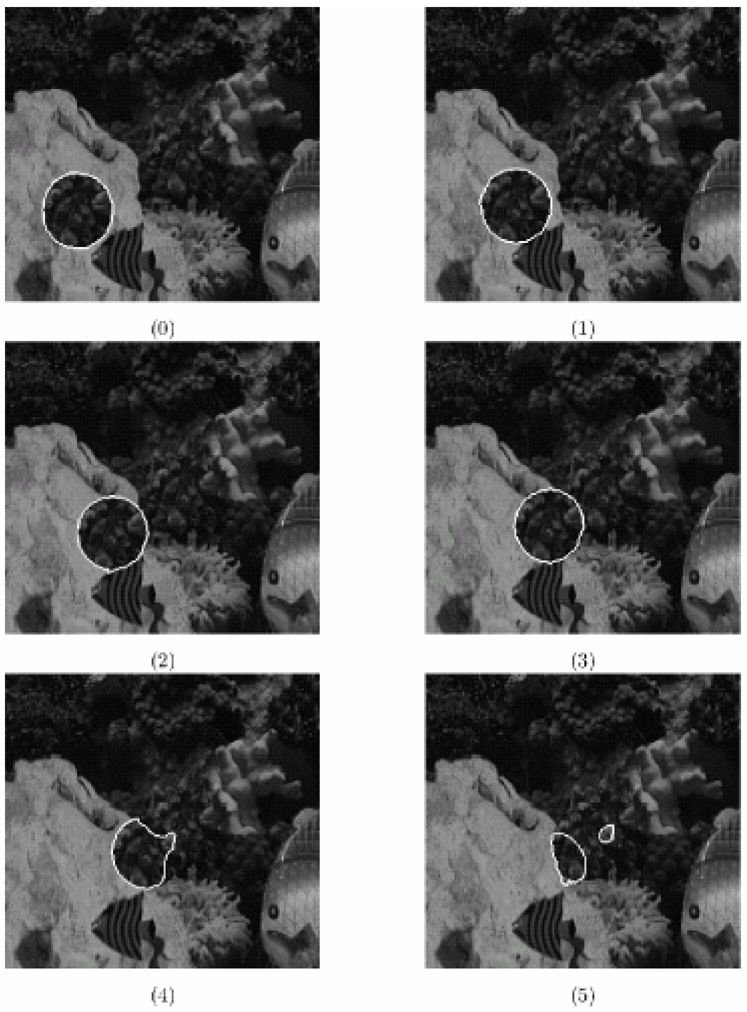


Fig 4.4 The camouflage synthetic sequence

This figure is derived from [17]. The tracked object is cut from the texture in the center of the image and pasted to create animation from the lower left corner to the upper right corner of the image.

In Fig 4.5, the lower right corner of the object in frame (a) is pasted by a square shape cut from the background around the object. Then frame (a) is duplicated to create frame (b) and is applied by some noise to its background such that the background in frame (b) is more similar to the square shape than the background in frame (a). The contour is initialized in frame (a) to properly capture the object. After evolution, the lower right corner of the contour sticks out and attempts to include the background, as shown in frame (b). This is because that the background is so similar to the square shape of the object that the probability estimates are almost identical for those pixels.

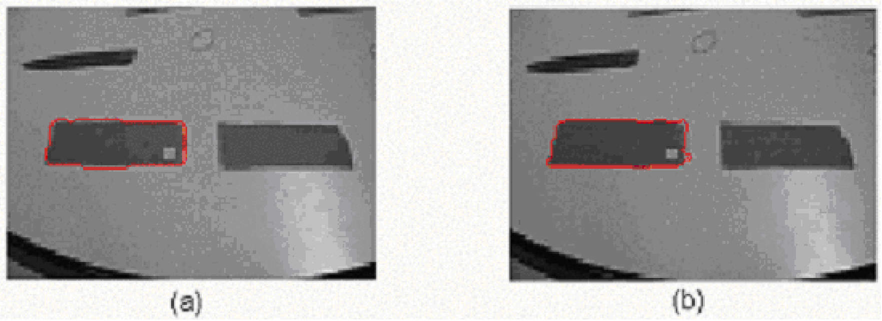


Fig 4.5 Problem of Abdol-Reza's model

The lower right part of the contour sticks out and tries to include the background (b), due to the similarity between the background and the square shape in the object.

4.4. Implementation Issues of the Level Set Method

The conversion from parametric representation of a curve to level set representation is exactly correct when the zero level set that representing the curve moves with a specified speed after updates of u. However, this is usually not true since the update of u is just an estimation derived from the level set equation, making some errors to the update. For example, consider an 1D curve represented by 2D level sets u, as shown in Fig 4.6(a). A perfect expression of the curve to move right with a speed F is to imaginarily move the level sets u to right with a distance $F\Delta t$, that is, to update each grid of u with the value below it after the imaginary movement, then each level set must move right with a speed F, including the level set which indicates the curve. However, this naïve kind of update is almost impossible in the real world. The update is actually an approximation derived by the level set equation, which is an increasing or decreasing amount $F\Delta t\|\nabla u\|$, as shown in Fig 4.6(b). We can see that this estimation of update of *u* is not correct since the level set on the new *u* cannot reflect the move with specified speed. And, if the function *u(**x**)* is a straight but not horizontal line on the plot, the estimation is correct and will match the imaginary movement. Also, this assumption is not useful in practice, except in very simple cases.

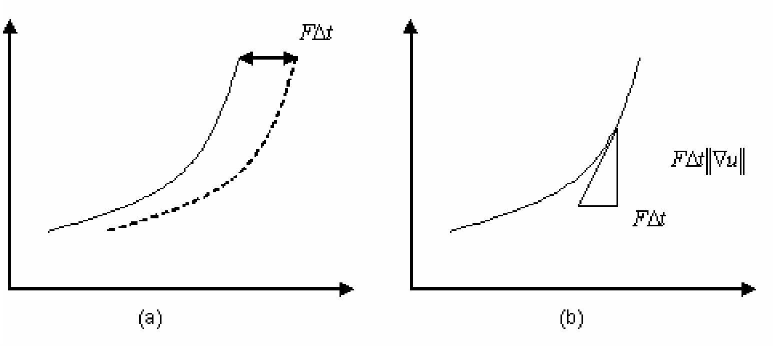


Fig 4.6 Approximation of $F\Delta t$

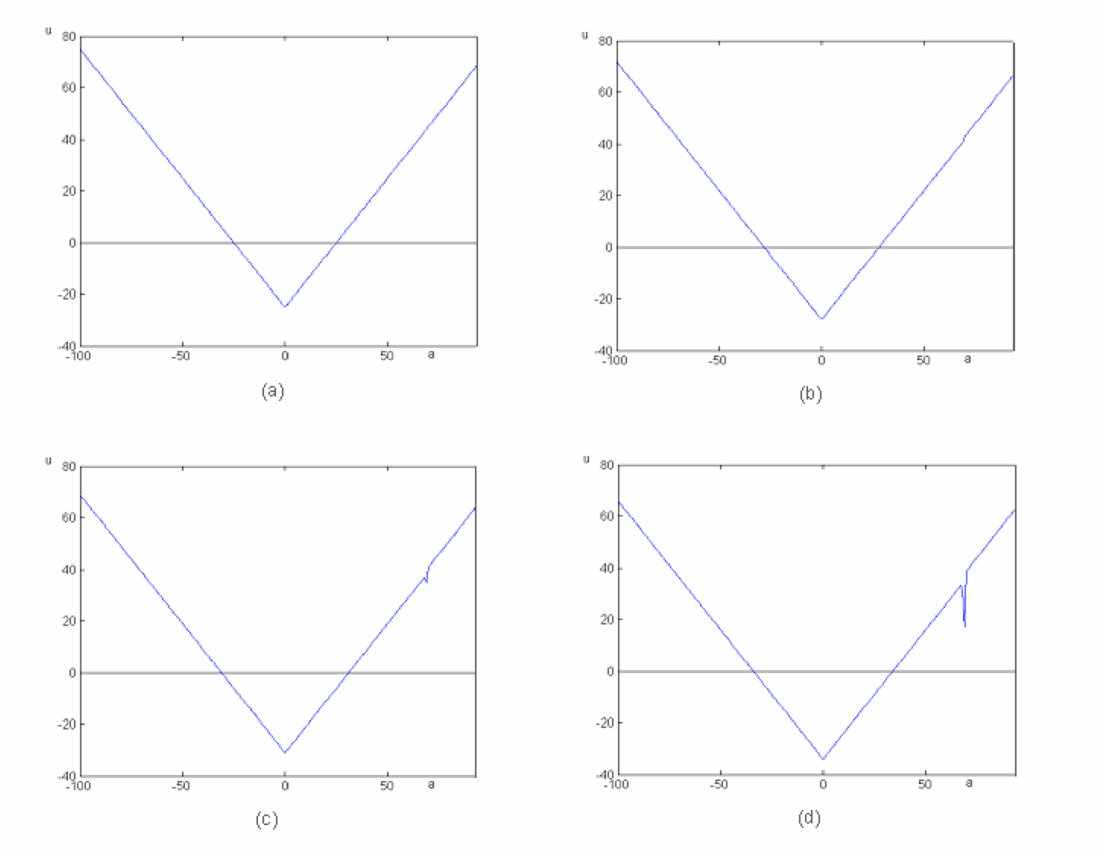
To explain it more clearly, let us take an extreme case as the example, the numerical problem when derivatives are not defined on the points that are discontinuous. Let *u* be a distance function with distance from the original, that is , $\|\nabla u\| = 1$ everywhere except the original, as shown in Fig 4.7(a). Recall the level set equation:

$$\frac{\partial u(x)}{\partial t} + F(x)\|\nabla u(x,t)\| = 0$$

Let

$$F(x) = \begin{cases} M & x \le a \\ N & x > a \end{cases} \quad ,M > N > 0$$

Then, at iteration 1, *u* becomes Fig 4.7(b). Since for all *x*, *F(**x**)* > 0, each grid of *u* moves downward. Also, because $M > N$, the grids $u(x \le a)$ moves more amount than the grids $u(x > a)$, and there is a discontinuity $\delta = u(a^{+}) - u(a^{-})$ occurs at *u(a)*.



(a)

(b)

(c)

(d)

Fig 4.7 Numerical Problem

This problem is simulated by a Matlab program. Note the difference between (a) and (b). In (b), *u* has moved downward and the point *u(a)* is a discontinuity. Then, (c) and (d) are iteration 2 and iteration 3 respectively.

Assume we use forward differencing

$$\nabla u = \frac{u_{i+1} - u_i}{\Delta x}$$

where the lower script denotes the grid index. Then,

$$\begin{aligned}
&\nabla u(a-\Delta x) = \frac{u(a) - u(a-\Delta x)}{\Delta x} = 1 \\
&\nabla u(a) = \frac{u(a+\Delta x) - u(a)}{\Delta x} = \frac{u(a) + \Delta x + \delta - u(a)}{\Delta x} = 1 + \frac{\delta}{\Delta x} \\
&\therefore \|\nabla u(a)\| > \|\nabla u(a-\Delta x)\| > 1
\end{aligned}$$

That is, from the level set equation, $u(a)$ will move downwards more than $u(a-\Delta x)$ at iteration 2, as shown in Fig 4.7(c). If we compute iteratively in this manner, $u(a)$ will touch the x-axis to become the member of the zero level set that represents the curve and produce a “bubble effect”, as shown in Fig 4.8.

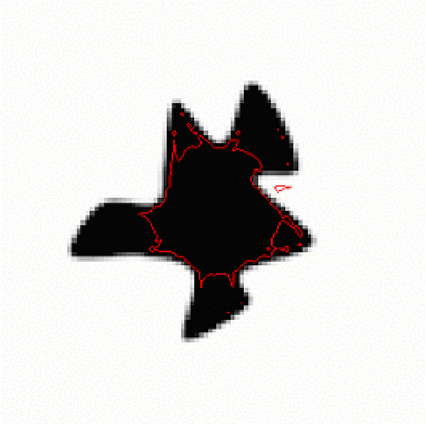


Fig 4.8 Bubble effect

This is the experimental result of level set method using forward differential scheme. The bubble effect refers to the occurrence of unexpected bubbles during the propagation of the curve.

To overcome this phenomenon, the upwind differencing scheme is proposed by [7].

$$u_{i,j}^{n+1} = u_{i,j}^{n} + \Delta t(-[\max(F_{i,j},0)\nabla^{+} + \min(F_{i,j},0)\nabla^{-}])$$

where we lower script u and F by their grid indices, $u_{i,j} \equiv u(i,j)$ and $F_{i,j} \equiv F(i,j)$, and the upper script of u denotes the number of iteration and

$$\begin{cases} \nabla^{+} = \left[\max(D_{ij}^{-x},0)^2 + \min(D_{ij}^{+x},0)^2 + \max(D_{ij}^{-y},0)^2 + \min(D_{ij}^{+y},0)^2\right]^{1/2}, \\ \nabla^{-} = \left[\max(D_{ij}^{+x},0)^2 + \min(D_{ij}^{-x},0)^2 + \max(D_{ij}^{+y},0)^2 + \min(D_{ij}^{-y},0)^2\right]^{1/2}, \\ D_{ij}^{-x} = u_{i,j}^{n} - u_{i-1,j}^{n}, \\ D_{ij}^{+x} = u_{i+1,j}^{n} - u_{i,j}^{n}, \\ D_{ij}^{-y} = u_{i,j}^{n} - u_{i,j-1}^{n}, \\ D_{ij}^{+y} = u_{i,j+1}^{n} - u_{i,j}^{n} \end{cases}$$

In 1D case it is

$$u_i^{n+1} = u_i^n + \Delta t(-[\max(F_i,0)\nabla^+ + \min(F_i,0)\nabla^-])$$

where

$$\begin{cases} \nabla^+ = \left[\max(D_i^{-x},0)^2 + \min(D_i^{+x},0)^2\right]^{1/2}, \\ \nabla^- = \left[\max(D_i^{+x},0)^2 + \min(D_i^{-x},0)^2\right]^{1/2}, \\ D_i^{-x} = u_i^n - u_{i-1}^n, \\ D_i^{+x} = u_{i+1}^n - u_i^n \end{cases}$$

The value on each grid of the speed function is greater than zero, those u's located at troughs will not be updated. Therefore, the grid $u(a)$ will not be updated in iteration 3 and remains its value until it is equal to $u(a-\Delta x)$ such that reduces the trend to produce large discontinuities.

Notice that in iteration 2 $u(a+\Delta x)$ is not a trough and, not like $u(a)$, will be updated in iteration 3 by the upwind differencing scheme. This leads to the fact that $u(a+\Delta x)$ might be updated with a lower value than $u(a)$ and $u(a)$ will become non-trough at iteration 3 such that is need to be updated in iteration 4. So, it seems that upwind differencing scheme just lower down $u(a)$ to touch the x-axis but $u(a)$ will eventually becomes a member of the zero level set. However, we will show that this worry is unnecessary under the satisfaction of Courant-Friedriches-Lewy (CFL) condition [7]. That is, $u(a)$ is still a trough ($u(a) < u(a+\Delta x)$) at iteration 3 such that it will not be updated at the next iteration if CFL condition is satisfied. Let the discontinuity at iteration 2 be $\rho = u(a+\Delta x) - u(a)$. Note that $\rho > \delta$ >0. The CFL condition is, for each grid point

$$F\Delta t < \Delta x$$

And without lost of generality, we can let $\Delta x = 1$, therefore $F\Delta t < 1$.

Now we will prove $u(a) < u(a+\Delta x)$ at iteration 3.

$$
\begin{aligned}
&\because F(a+\Delta x) > 0 \\
&u_{a+\Delta x}^{3} = u_{a+\Delta x}^{2} + \Delta t(-F_{a+\Delta x}\nabla^{+}) \\
&= u_{a+\Delta x}^{2} - \Delta t F_{a+\Delta x}\left[\max(D_{a+\Delta x}^{-x},0)^{2} + \min(D_{a+\Delta x}^{+x},0)^{2}\right]^{1/2} \\
&D_{a+\Delta x}^{-x} = u_{a+\Delta x}^{2} - u_{a}^{2} = \rho > \delta > 0 \\
&D_{a+\Delta x}^{+x} = u_{a+2\Delta x}^{2} - u_{a+\Delta x}^{2} = 1 \\
&= u_{a+\Delta x}^{2} - \Delta t F_{a+\Delta x}\left[\max(\rho,0)^{2} + \min(1,0)^{2}\right]^{1/2} \\
&= u_{a+\Delta x}^{2} - \Delta t F_{a+\Delta x}\rho > u_{a+\Delta x}^{2} - \rho = u_{a}^{2}
\end{aligned}
$$

It is easy to derive a general form $u_{a+\Delta x}^{n+1} > u_{a}^{n}$ by mathematical induction.

5. Shape Prior

In this chapter, we present how we combine the shape prior with Abdol-Reza's model. The shape prior refers to knowledge of shape models which we have known that can be used to guide the propagation of the curve. The sections contain geodesic active contour model that is used to represent the shape prior and a distance measure for measuring the dissimilarity between the propagating curve and the shape prior.

5.1. Geodesic Active Contour Model

Let $J_{R_{0,\theta,\delta}}^{n+1}$ be a binary image with ones on the region R_0, after rotated an angle θ and translated a distance δ, and zeros on the remaining domain. R_0 is indeed the region of the object captured by the curve in the previous frame image I^n, and the binary image $J_{R_{0,\theta,\delta}}^{n+1}$ containing R_0 after the transformation is used to construct the shape prior for the current image I^{n+1}. During the curve γ is propagating to capture the object in I^{n+1}, the binary image $J_{R_{0,\theta,\delta}}^{n+1}$ for the shape prior is reconstructed each time before computing equation (5.1) in each iteration. The region R_0 in the binary image only subjects to a rotation θ and a translation δ to align with the propagating curve γ, and never deforms its shape all the time (in processing I^{n+1}). Although the transformation is limited to rotation and translation in this thesis, it can be extended to combinations of several affine transforms such as scaling and shearing. Now we will define the energy functional

$$E = E_r(\vec{\gamma} \,|\, I^n, I^{n+1}, R_0) + E_s(u, \theta, \delta) \qquad (5.1)$$

where the image energy *Er* is equation (4.1) used to perform discriminate analysis on each pixel and the shape energy *Es* is

$$E_s = \int_0^1 w(E_\gamma)\cdot \left\| g(J^{n+1}_{R_0,\theta,\delta}(\vec{\gamma}))\right\| \cdot \left\|\frac{d\vec{\gamma}}{ds}\right\| ds \quad (5.2)$$

where *w(E$_r$)* is a weighting function ,which means the shape prior is image energy dependent. For pixels that can clearly classified by *Er*, the weight is low. If the pixel lies in an uncertainty state, the weight is high and the shape prior will leads the classification. This novelty, different from most existing methods that the weights used to combine the terms of the energy functional are usually constant thresholds, offers more flexibility in the propagation of the contour. In particular, the shape prior will not constrain the deformability of the shape of the object in many cases.

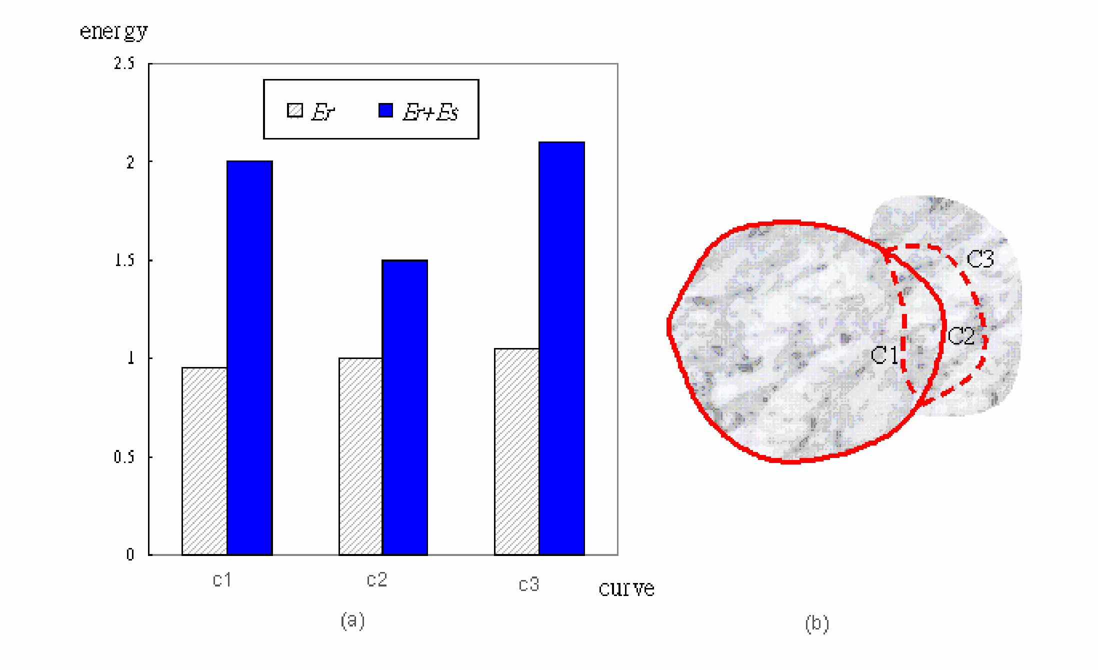


Fig 5.1 Image energy and Shape energy

The energy for the curve c1, c2 and c3 is almost identical, but the energy for c2 is lowered than the others after plus the shape energy.

Fig 5.1illustrates our basic idea. In Fig 5.1(b), the object is partially involved in the background that is textured with similar appearance to it. Either the curve expands (c3) or shrinks (c1), the energy changes are difficult to distinguish. However, after plus the shape energy, we can eliminate this uncertainty and make a decision.

Despite the weighting function, the term *Es* comes from the geodesic active contour model. $g(I)$ stands for $\frac{1}{1+\left\|\nabla\hat{I}\right\|^{p}}$, where $\hat{I}$ is a smoothed version of I, computed using Gaussian filtering, and p = 1 or 2. means that to convert the binary image to an image with continuous numbers from zero to one. Also, if we view the image in 3-dimensional space, the valley will locate at the boundary of the object in the binary image $J^{n+1}_{R_{0,}\theta,\delta}$. Fig 5.2 is the observing intensity of a binary image crossed by a scan line. In Fig 5.2(a) and (b), I and its smoothed version $\hat{I}$ are presented. Fig 5.2(c) shows $g(\cdot)$ is valley on the boundary of I, and its gradient vectors direct to the middle of the boundary.

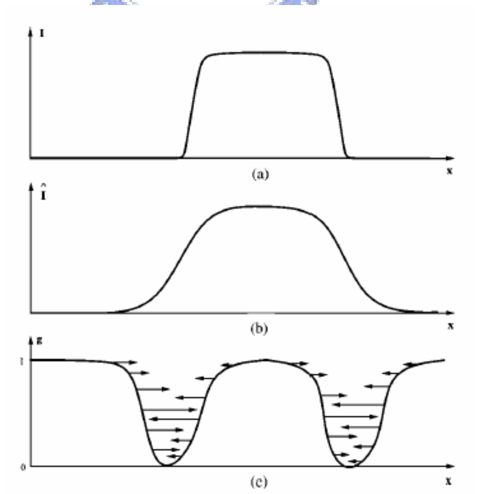


Fig 5.2 Gradient vector

(a) The intensity signal on a scan line over the binary image. (b) The smoothed version of (a). (c) The derived function g. The evolving contour is attracted to the middle of the boundary by the forces created by $\nabla g \cdot \nabla u$.

The integration (5.2) means the sum of $\left\| g(J^{n+1}_{R_0,\theta,\delta}(\vec{\gamma})) \right\|$ over the locations occupied by the curve γ multiplied by its piece of length. Since the derived function $g(\cdot)$ takes small values on positions corresponding to the boundary of the object in the binary image, we can expect that the minimum is obtained when the curve is aligned to the boundary of the object in the binary image. To minimize the integration, we search for its gradient descent direction, which is computed from the Euler-Lagrange equation

$$\frac{\partial \vec{\gamma}(t)}{\partial t} = w(E_\gamma)\left[g(J^{n+1}_{R_0,\theta,\delta}(\vec{\gamma}))\kappa_{\vec{\gamma}} - \nabla g(J^{n+1}_{R_0,\theta,\delta}(\vec{\gamma})) \cdot \vec{n} \right]\vec{n} \qquad (5.3)$$

The term $\nabla g(\cdot)$ attracts the movement of the curve towards the boundary of shape prior. An example is given in Fig 5.3. Let $w(E_\gamma) = 1$ to clarify the effect of the vector field.

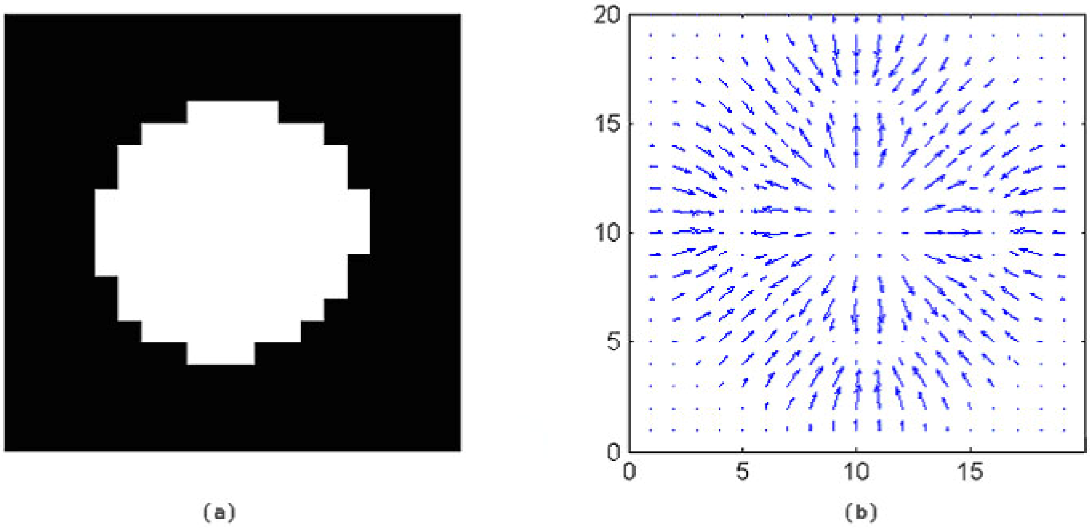


Fig 5.3 Shape and its forces field

(a) A binary image J of size 20x20 containing a circle object. (b) The corresponding gradient vectors $\nabla g \cdot \nabla u$ represent the forces that attract the curve to the pointed direction.

Obviously, this is an explicit representation of curve $\vec{\gamma}$, and we can embed it to the level set representation as described in the previous chapter.

$$\frac{\partial u}{\partial t} = w(E_\gamma)\left[g(J^{n+1}_{R_0,\theta,\delta})\kappa_u - \nabla g(J^{n+1}_{R_0,\theta,\delta}) \cdot \frac{\nabla u}{\|\nabla u\|} \right]\|\nabla u\| \qquad (5.4)$$

Combining (5.4) with (4.4) we can obtain

$$\begin{aligned} \frac{\partial u(x)}{\partial t} &= \left[\log P_{in,x}(I^{n+1}(x)|I^n, R_0) - \log P_{out,x}(I^{n+1}(x)|I^n, R_0)\right]\|\nabla u\| \\ &- \lambda\kappa_u(x)\|\nabla u\| + w(E_\gamma)\left[g(J^{n+1}_{R_0,\theta,\delta}(x))\kappa_u - \nabla g(J^{n+1}_{R_0,\theta,\delta}(x)) \cdot \frac{\nabla u}{\|\nabla u\|} \right]\|\nabla u\| \end{aligned} \qquad (5.5)$$

where

$$w(E_\gamma) = \frac{\alpha}{\left|\log P_{in,x}(I^{n+1}(x)|I^n, R_0) - \log P_{out,x}(I^{n+1}(x)|I^n, R_0)\right| + \beta} \qquad (5.6)$$

The weighting function is related to the reciprocal of the image energy, where α (> 0) and β (> 1) are two user specified constants. When the uncertainty occurs, the weight will take a large value, and, on the other hand, the weight is low when the classification can be confirmed. The purpose of the shape prior is to lead the propagation when the classification is uncertain when the term E_γ is almost equal either the contour shrinks or expands.

Before computing equation (5.5), however, we need to reduce the involved parameters, that is, to compute $\theta*$ and $\delta*$ which are the optimizing transform of the shape prior to align with the contour. The final level set equation is (5.7).

$$\begin{aligned} \frac{\partial u(x)}{\partial t} &= \underbrace{\left[\log P_{in,\bar{\gamma}}(I^{n+1}(x)|I^n, R_0) - \log P_{out,\bar{\gamma}}(I^{n+1}(x)|I^n, R_0)\right]\|\nabla u\| - \lambda\,\kappa_u(s)\|\nabla u\| +}_{E_r} \\ &\underbrace{w(E_\gamma)\left[g(J^{n+1}_{R_0,\theta*,\delta*}(u))\kappa_u - \nabla g(J^{n+1}_{R_0,\theta*,\delta*}(u)) \cdot \frac{\nabla u}{\|\nabla u\|} \right]\|\nabla u\|}_{Es} \end{aligned} \qquad (5.7)$$

This is done by using a dissimilarity measurement between distance functions of the curve and the shape prior. We will discuss this subject in the next section.

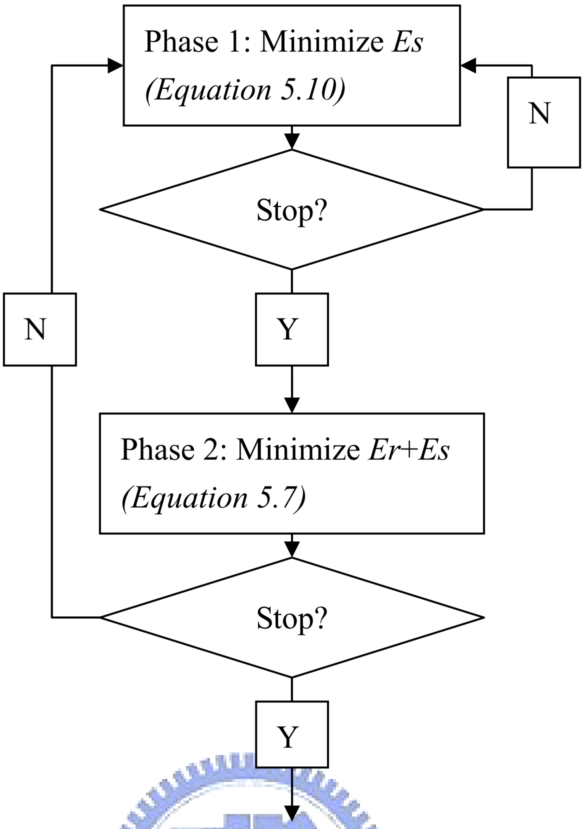
5.2. An Euclidean Invariant Shape Prior

Computing θ* and δ* separately eliminates translation and rotation from the shape energy *Es* and no additional parameters entering the minimization. A commonly alternative approach is to explicitly model a translation and an angle and minimize with respect to these quantities (by gradient descent or other). In Daniel Cremers et al.'s opinion [20], the explicitly model has several drawbacks:

1. The introduction of explicit pose parameters makes numerical minimization more complicated— corresponding parameters to balance the gradient descent must be chosen. In practice, this is not only tedious, but it may cause numerical instabilities if the corresponding step sizes are chosen too large.
2. The joint minimization of pose and shape parameters mixes the degrees of freedom corresponding to translation and rotation with those corresponding to shape deformation.
3. Potential local minima may be introduced by the additional pose parameters. In a given application, this may prevent the convergence of the contour towards the desired segmentation.

Therefore, to reduce the parameters in minimizing *Es*, we compute θ* and δ* and reconstruct the binary image before computing (5.7). The idea of evolution process thus consists of two phases:

1. Aligning the shape prior with the curve (compute θ* and δ*).
2. Evolving the curve, including propagating the certain parts toward the object and attracting the uncertain parts to the shape prior (computing equation (5.7)).



Each iteration of the algorithm consists of the following steps:

Step 0. Manually draw the initial contour of the object being tracked in the first frame such that the contour points contain the desired object boundary. You may exclude the shadow. Set the current frame number to 2.

Step 1. Use the contour obtained in the previous frame as the initial contour and the prior shape of the current frame.

Step 2. Compute the rotation and translation parameters between the current contour and the prior shape (see equation 5.10)

Step 3. Do for all contour points one at a time:

use equation 5.7 to compute E_r, E_s, and $W(E_r)$. Then update the level set value at the contour point.

Step 4. After all the contour points are updated, check if the contour does not converge or the number of iteration does not exceeds a specified upper bound; if not, go to Step 2; otherwise, terminate with the contour as the final contour of the current frame.

Step 5. Is the frame the final frame? If yes, stop; otherwise, increase the frame number by 1 and go to Step 1.

The evolution process is performed iteratively to minimize the energy functional (6). The idea is illustrated in Fig 5.4 and Fig 5.5.

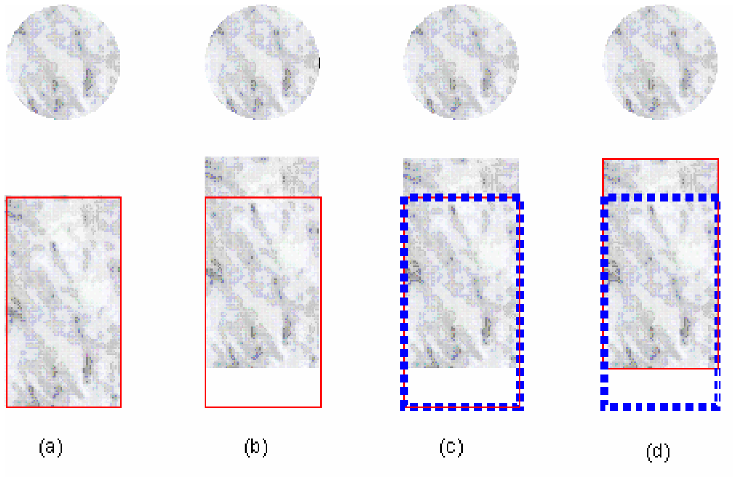


Fig 5.4 Shape prior illustration 1

Fig 5.4 presents that the contour (the red curve) will neglect the shape prior (the blue equal signs curve) to capture the object (the rectangle) when it can distinguish the object from the background (the circle). In Fig 5.4(a), a rectangular object staying at the bottom is textured with marble appearance, and at the top locates a part of the background that is also textured with marbles. A red curve is initialized to properly capture the object. Then, the object moves upwards with a small distance in Fig 5.4 (b). The evolution process of the curve is shown in Fig 5.4(c) and Fig 5.4(d). Fig 5.4(c) is the early phase of the computation, that is, to align the shape prior with the curve. Since the curve is staying at its original location and suffered no deformation so far, the shape can completely align with the curve. In Fig 5.4(d), the computation enters the 2nd phase and the curve starts to move according to the image information and the attraction of the shape prior. However, because the curve can distinguish the object from the background, which is white at this time, the curve just moves to capture the object directly, ignoring the shape prior.

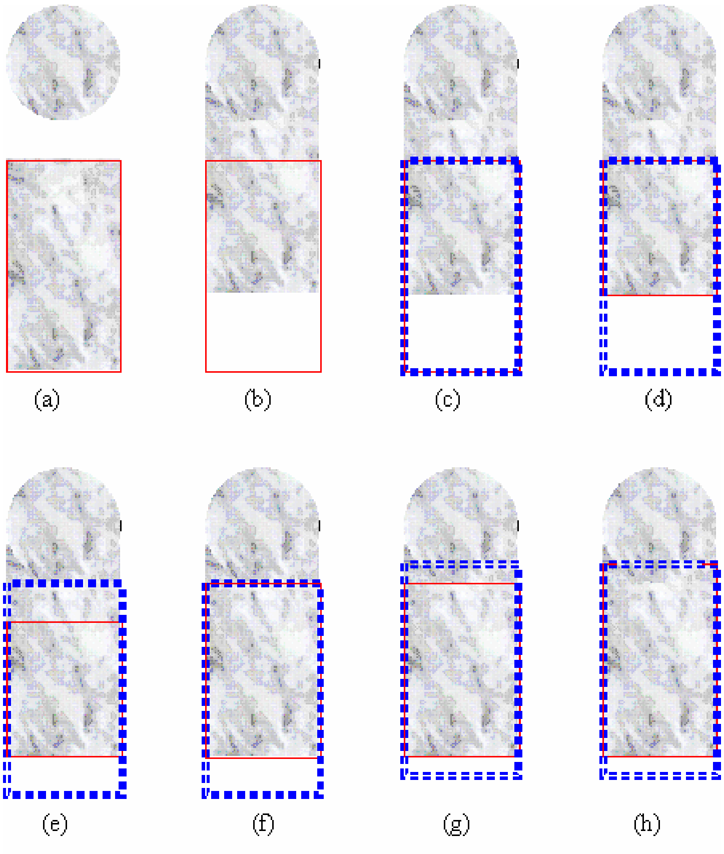


Fig 5.5.Shape prior illustration 2

The object continuously moves upwards and, at this time, overlaps with the part of the background that is similar to the object.

Fig 5.5 presents the idea of the shape prior to capture the object involved in the background in this figure. Fig 5.5(a) is the capturing result of Fig 5.4 and also the initialization of Fig 5.5, with the curve properly captured the object. Then the object moves with a large step into the background in Fig 5.5(b). Now, the remaining figures are trying to capture the object. Fig 5.5(c) is to align the shape prior with the curve. As in Fig 5.4(c), it is completely matched with the contour. However, in Fig 5.5(d), the attraction of the shape prior takes effect. The bottom part of the curve, again, moves freely to capture the bottom of the object since the curve knows where the object is, but the top part of the curve is locked by the shape prior because it dose not known the marbles above belongs to the object or belongs to the background. This is because the marble at the bottom can only belongs to the object since no similar background is around it, and the marble at the top has alternative choices. Continuously, the shape prior is aligned with the curve in Fig 5.5(e). However, the size of the curve has changed. To make the alignment possible, the best solution is to place the curve in the center of the shape prior, as shown in the figure. After this arrangement, the curve can continuously evolve in Fig 5.5(f). But opposed to Fig 5.5(d), the bottom part of the curve stays at the same location but the top part can take liberty to move upwards by the attraction of the shape prior. Note that the top part of the curve is moving towards the actual boundary of the object by the leading of the shape prior after its alignment, and the certain parts of the curve can stay at its positions. In Fig 5.5(g) and Fig 5.4(h), the evolving process repeats continuously in this way. The shape prior goes upwards with a little step in Fig 5.4(g), and then the top part of the curve is attracted upwards. After several iterations, the curve can capture the object with an allowable error.

Now we formulate the computation of θ^* and δ^* now. A dissimilarity measure is calculated for two distance functions. Let $R_0(\theta,\delta)$ denotes the region after rotated an angle θ and translated a distance δ. Let ϕ_0 be the distance function of $\partial R_0(\theta,\delta)$, the boundary of $R_0(\theta,\delta)$, over the domain Ω. Let ϕ_1 be the distance function of the propagating curve (the level set $u = 0$) over Ω. The dissimilarity measure between ϕ_0 and ϕ_1 is

$$d^2(\phi_0(x,\theta,\delta),\phi_1(x)) = \int_\Omega (\phi_0(x,\theta,\delta) - \phi_1(x))^2 \frac{h(\phi_0)+h(\phi_1)}{2} dx \qquad (5.8)$$

$$= f(\theta,\delta) = f(\rho)$$

where

$$H(\phi) = \begin{cases} 1, & \phi \geq \mathit{0} \\ 0, & else \end{cases} \qquad h(\phi) = \frac{H(\phi)}{\int_\Omega H(\phi) dx}$$

This distance measure is symmetric but merely a pseudo-distance [22]. The minima of (5.8) occurs at $\frac{\partial d^2(\phi_0,\phi_1)}{\partial \rho} = 0$. Let $\rho = \{\theta,\delta\}$ be the pose parameters, the optimization $\rho^* = \{\theta^*,\delta^*\}$ can be derived by calculating the gradient descent direction $-\frac{\partial d^2(\cdot)}{\partial \rho}$ and is given by

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= -\int_\Omega (\phi_1 - \phi_0)(h(\phi_1) + h(\phi_0)) \frac{\partial \phi_0}{\partial \rho} dx \\ &- \frac{1}{2\int_\Omega H(\phi)} \int_\Omega \left[(\phi_1 - \phi_0)^2 - \overline{(\phi_1 - \phi_0)^2} \right] \delta(\phi_0) \frac{\partial \phi_0}{\partial \rho} dx \end{aligned} \qquad (5.9)$$

where $\delta_\varepsilon(s) = H'_\varepsilon(s)$ is delta function, which is chosen to have and infinite support:

$$\delta_\varepsilon(s) = \frac{1}{\pi} \frac{\varepsilon}{\varepsilon^2 + s^2}$$

and

$$\overline{(\phi_1 - \phi_0)^2} = \int (\phi_1 - \phi_0)^2 h(\phi_0) dx$$

Notice that the distance measure (5.8) is not an energy functional and (5.9) is derived by computing its gradient descent direction but not derived from computing

Euler-Lagrange equation. Then, equation (5.9) can be used to solve the optimal pose parameters θ^* and δ^* iteratively.

Proof of (5.9):

$$
\begin{aligned}
&\frac{\partial\rho}{\partial t}=-\frac{\partial d^{2}}{\partial\rho}\\
&=-\int\frac{\partial}{\partial\rho}\left[(\phi_1-\phi_0)^2\frac{h(\phi_1)+h(\phi_0)}{2}\right]dx\\
&=-\int_{\Omega}(\phi_1-\phi_0)(h(\phi_1)+h(\phi_0))\frac{\partial\phi_0}{\partial\rho}dx+\frac{(\phi_1-\phi_0)^2}{2}\frac{\partial h(\phi_0)}{\partial\rho}\\
&=-\int_{\Omega}(\phi_1-\phi_0)(h(\phi_1)+h(\phi_0))\frac{\partial\phi_0}{\partial\rho}dx-\frac{1}{2}\int(\phi_1-\phi_0)^2\frac{\partial h(\phi_0)}{\partial\rho}dx\\
&=-\int_{\Omega}(\phi_1-\phi_0)(h(\phi_1)+h(\phi_0))\frac{\partial\phi_0}{\partial\rho}dx-\frac{1}{2}\int(\phi_1-\phi_0)^2\frac{\partial}{\partial\rho}\frac{H(\phi_0)}{\int H(\phi_0)dx}dx\\
&=-\int_{\Omega}(\phi_1-\phi_0)(h(\phi_1)+h(\phi_0))\frac{\partial\phi_0}{\partial\rho}dx-\frac{1}{2}\int(\phi_1-\phi_0)^2\left[\frac{\delta(\phi_0)\frac{\partial\phi_0}{\partial\rho}}{\int H(\phi_0)dx}-\frac{H(\phi_0)\int\delta(\phi_0)\frac{\partial\phi_0}{\partial\rho}dx}{(\int H(\phi_0)dx)^2}\right]dx\\
&=-\int_{\Omega}(\phi_1-\phi_0)(h(\phi_1)+h(\phi_0))\frac{\partial\phi_0}{\partial\rho}dx-\frac{1}{2}\int(\phi_1-\phi_0)^2\left[\frac{\delta(\phi_0)\frac{\partial\phi_0}{\partial\rho}}{\int H(\phi_0)dx}-\frac{h(\phi_0)\int\delta(\phi_0)\frac{\partial\phi_0}{\partial\rho}dx}{\int H(\phi_0)dx}\right]dx\\
&=-\int_{\Omega}(\phi_1-\phi_0)(h(\phi_1)+h(\phi_0))\frac{\partial\phi_0}{\partial\rho}dx-\frac{1}{2\int H(\phi_0)}\int(\phi_1-\phi_0)^2\left[\delta(\phi_0)\frac{\partial\phi_0}{\partial\rho}-h(\phi_0)\int\delta(\phi_0)\frac{\partial\phi_0}{\partial\rho}dx\right]dx\\
&=-\int_{\Omega}(\phi_1-\phi_0)(h(\phi_1)+h(\phi_0))\frac{\partial\phi_0}{\partial\rho}dx-\frac{1}{2\int H(\phi_0)}\int\left[(\phi_1-\phi_0)^2\delta(\phi_0)\frac{\partial\phi_0}{\partial\rho}-(\phi_1-\phi_0)^2h(\phi_0)\int\delta(\phi_0)\frac{\partial\phi_0}{\partial\rho}dx\right]dx\\
&=-\int_{\Omega}(\phi_1-\phi_0)(h(\phi_1)+h(\phi_0))\frac{\partial\phi_0}{\partial\rho}dx-\frac{1}{2\int H(\phi_0)}\int\left[(\phi_1-\phi_0)^2\delta(\phi_0)\frac{\partial\phi_0}{\partial\rho}-\delta(\phi_0)\frac{\partial\phi_0}{\partial\rho}\int(\phi_1-\phi_0)^2h(\phi_0)dx\right]dx\\
&=-\int_{\Omega}(\phi_1-\phi_0)(h(\phi_1)+h(\phi_0))\frac{\partial\phi_0}{\partial\rho}dx-\frac{1}{2\int H(\phi_0)}\int\left[(\phi_1-\phi_0)^2-\int(\phi_1-\phi_0)^2h(\phi_0)dx\right]\delta(\phi_0)\frac{\partial\phi_0}{\partial\rho}dx\\
&=-\int_{\Omega}(\phi_1-\phi_0)(h(\phi_1)+h(\phi_0))\frac{\partial\phi_0}{\partial\rho}dx-\frac{1}{2\int_{\Omega}H(\phi)}\int_{\Omega}\left[(\phi_1-\phi_0)^2-\overline{(\phi_1-\phi_0)^2}\right]\delta(\phi_0)\frac{\partial\phi_0}{\partial\rho}dx
\end{aligned}
$$

Let $\frac{\partial\rho}{\partial t}\to 0$ *or a very small value.*

$$\rho^{k+1}=\rho^{k}+h\cdot\frac{-\partial d^{2}}{\partial\rho}\Big|_{\rho=\rho^{k}};h=0.1;$$

$$\textit{repeat until}\left|\frac{-\partial d^{2}}{\partial\rho}\Big|_{\rho=\rho^{k}}\right|<\varepsilon$$

6. Experiments

There are three experiments. Experiment 1 explores the characteristics of the tracking method. Experiment 2 presents the mechanism of the shape prior. Finally we use real image sequences in experiment 3.

6.1. Synthetic Image Sequences

This experiment is to explore characteristics of the tracking method.

(a) Small Displacement

There are three cases in this experiment. Case 1 is an object with homogeneous color moving from left to right and approaching a background texture that is similar to the object. In case 2, an object with a small part of it is pasted by the background texture. The object is moving from left to right in a homogeneous background. Finally in case 3 is a combination of case 1 and case 2. An object with a small part similar to the background is moving approaching a background texture that is similar to that object. The image sequence is of size 80x100 each. Since the move takes 6 pixels, we set the size of the searching window $\eta = 7$. Number of iterations = 40.

Both two methods can capture the object correctly. In figure 3 and figure 4, the nearest distance between the background-like texture within the object and its boundary is 14 pixels. In figure 5 and figure 6, the least distance between the object and the object-like background is 8 pixels. The size of searching window is 7 pixels. Therefore, each pixel is under certainty and we can get good results.

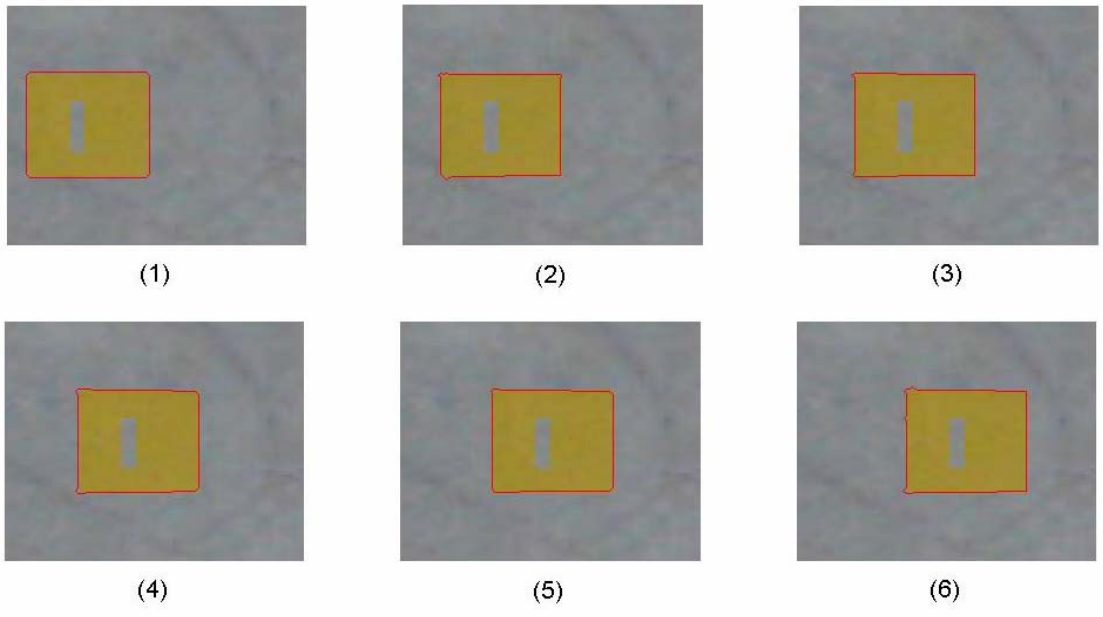


Fig 6.1 Abdol-Reza’s model

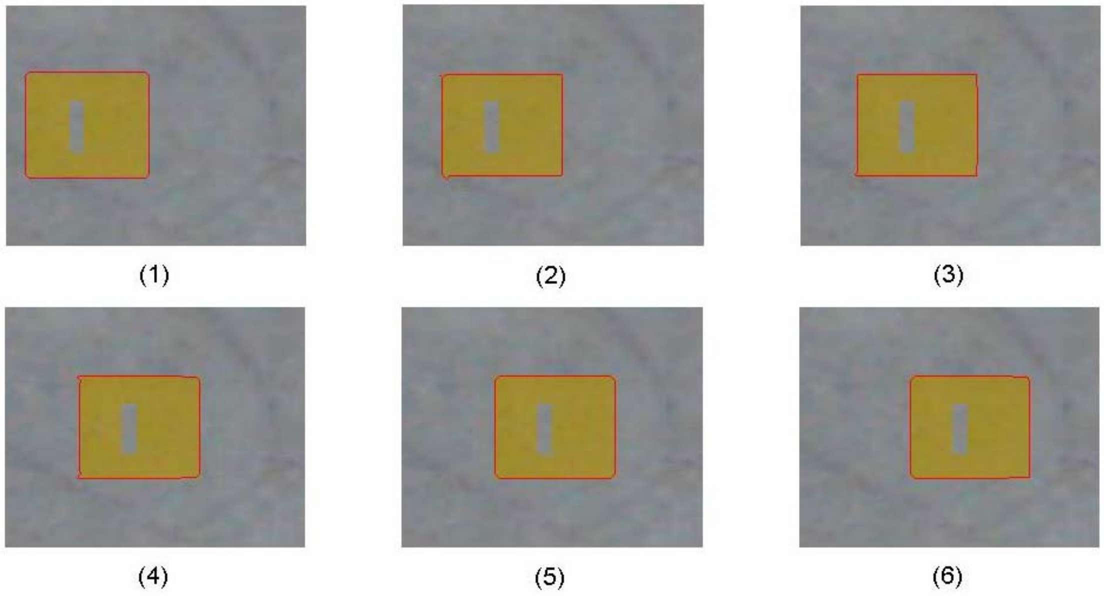


Fig 6.2 Abdol-Reza’s model + shape prior

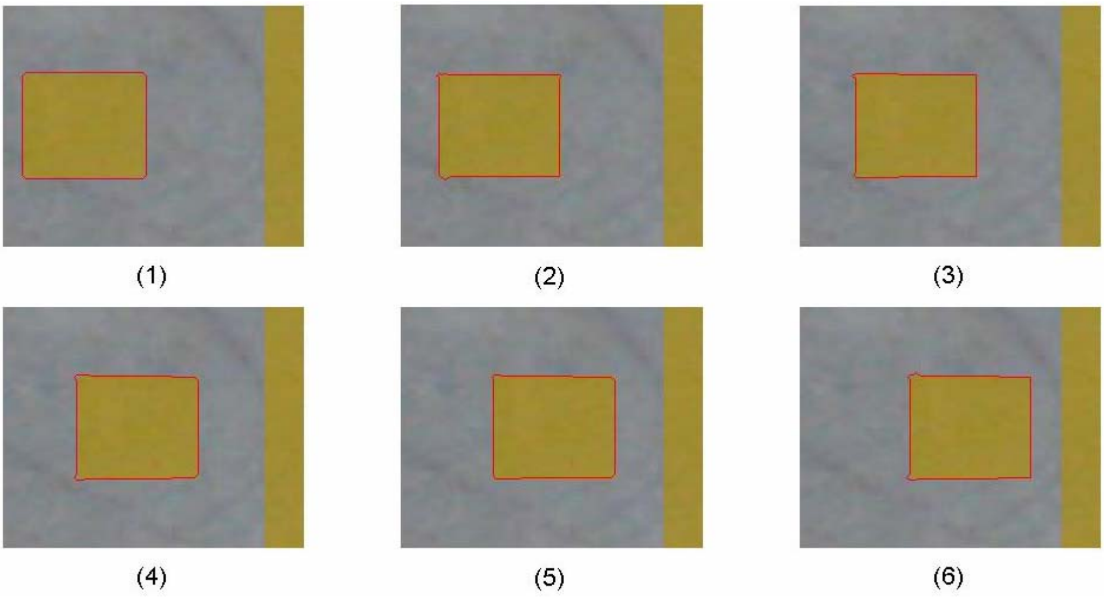

(1) (2) (3) (4) (5) (6)

Fig 6.3 Abdol-Reza’s model

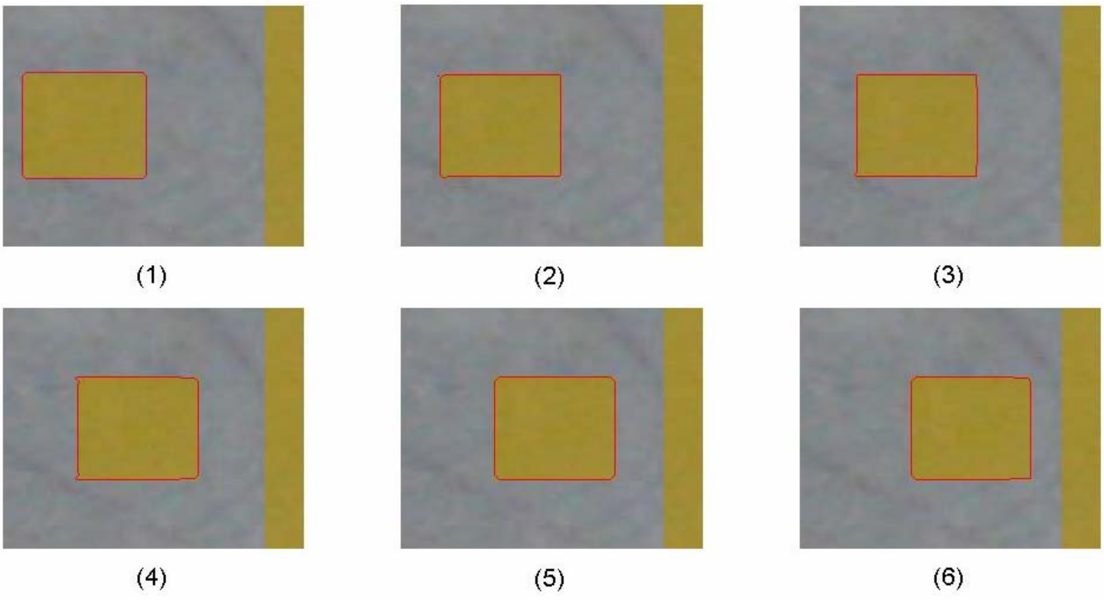
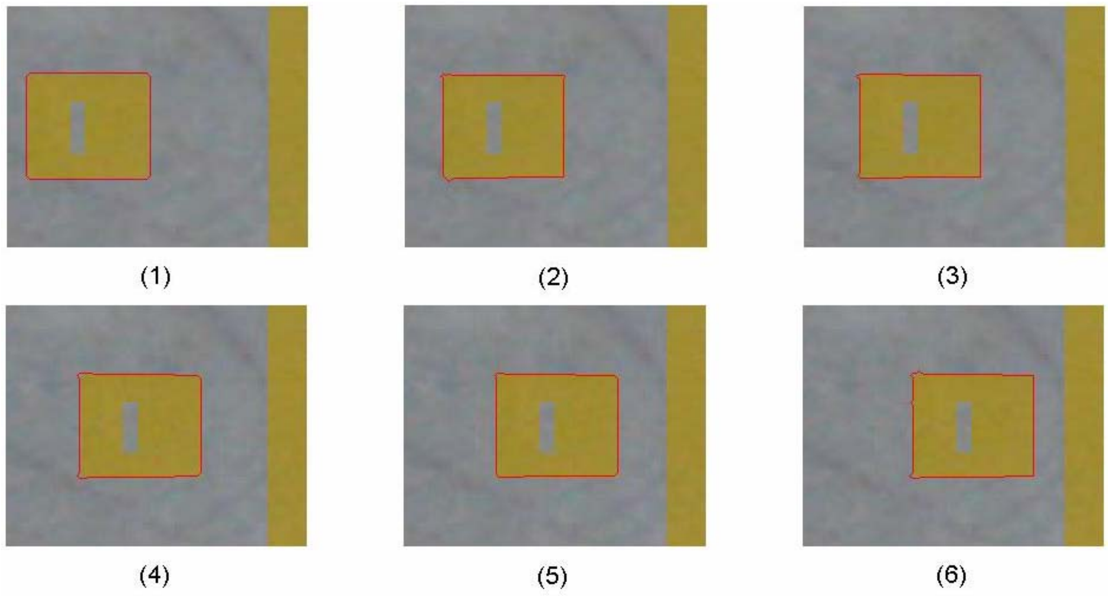

(1) (2) (3) (4) (5) (6)

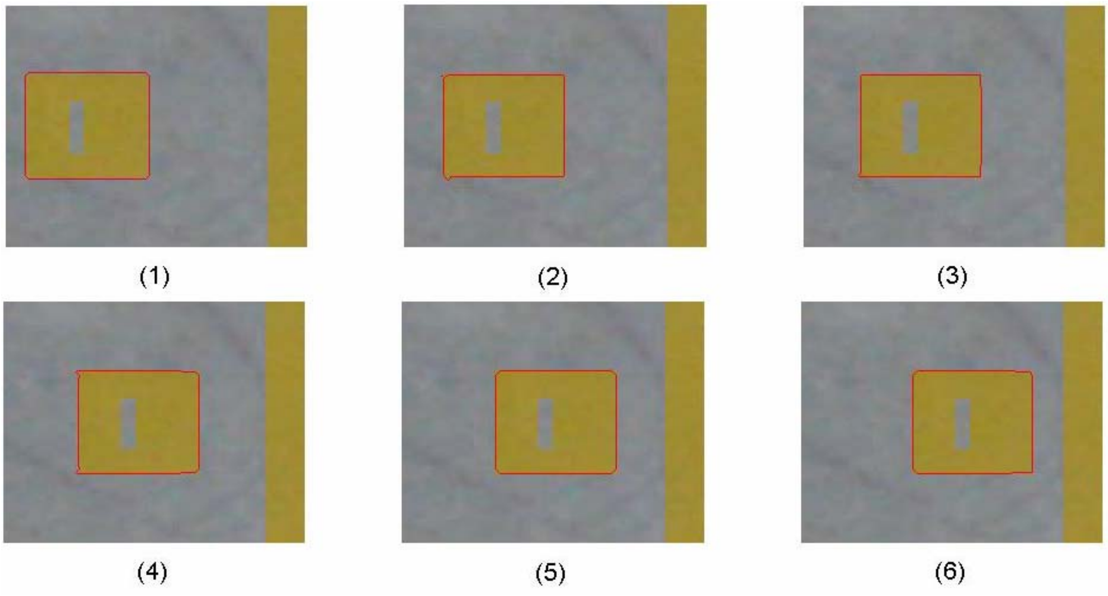
Fig 6.4 Abdol-Reza’s model + shape prior



(1) (2) (3)

(4) (5) (6)

Fig 6.5 Abdol-Reza’s model



(1) (2) (3)

(4) (5) (6)

Fig 6.6 Abdol-Reza’s model + Shape prior

(b) Large Displacement

This experiment is similar to experiment 1, but with a large move distance. We take frame 1, 3, 6 from each case in experiment 1 and compare the two approaches. The move distance is 18 pixels. We set the size of searching window η = 20 and the number of iterations is 80.

Abdol-Reza's model's model failed. In frame 2 of figure 1, the distance between the pixels around the corrupted contour and the background-like part in frame 1 is at most 14 pixels. However, since the search range is 20 pixels, we cannot determine the pixels left to the object as the object or background.

In figure, since the right part of the contour can correctly capture the object, it drags the shape to right and make the alignment. Then the left part of the contour is compensated by the shape prior and we can get the good result.

Similarly, in figure 4 the right part of the contour is determined by shape prior which is pushed by the left of the contour.

In figure 5 and figure 6, both methods failed. The right and left part of the contour cannot certainly to align the shape, thus the shape cannot be attracted to right position. This case can be handled by background registration and removal before contour propagation. The result is in figure 7.

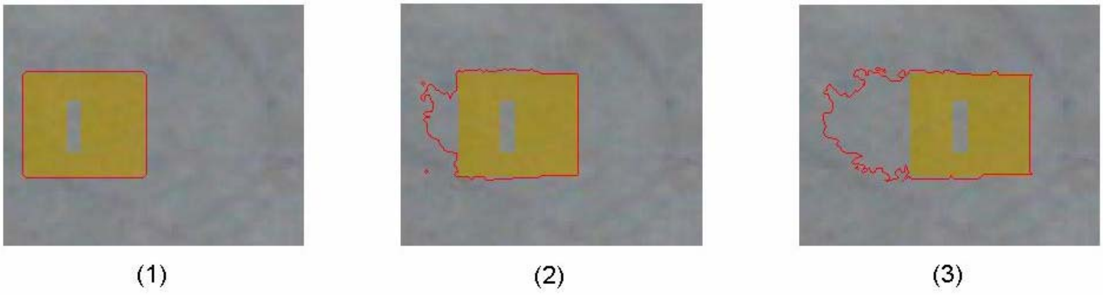


Fig 6.7 Abdol-Reza's model.

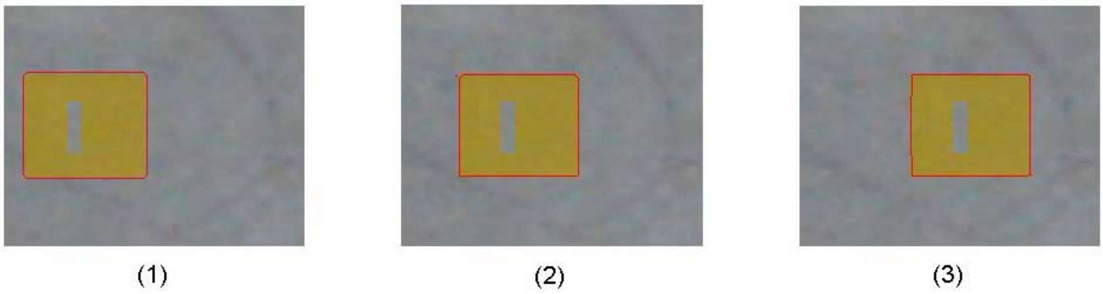


Fig 6.8 Abdol-Reza's model + Shape prior.

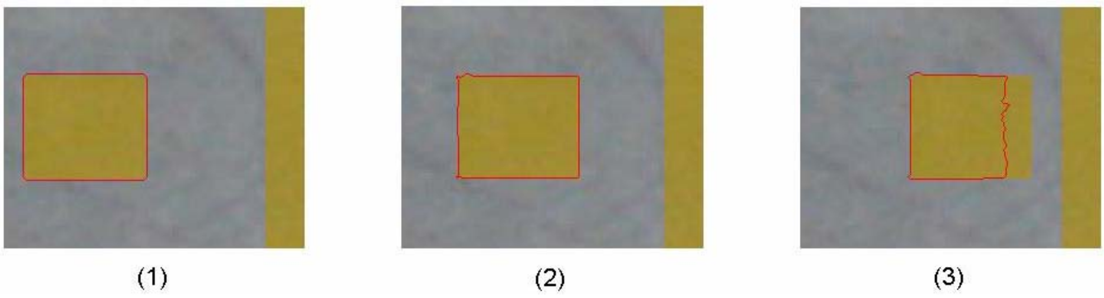


Fig 6.9 Abdol-Reza's model.

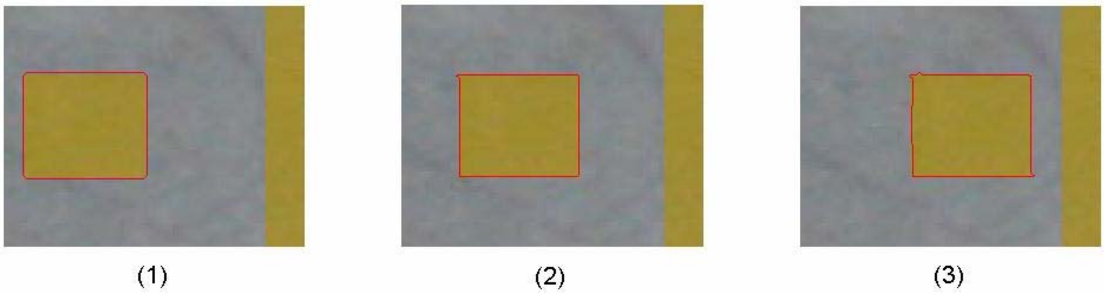


Fig 6.10 Abdol-Reza's model + Shape prior.

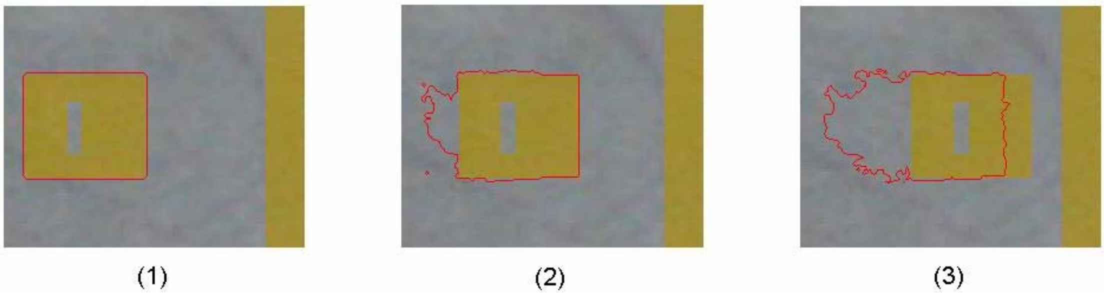


Fig 6.11 Abdol-Reza's model.

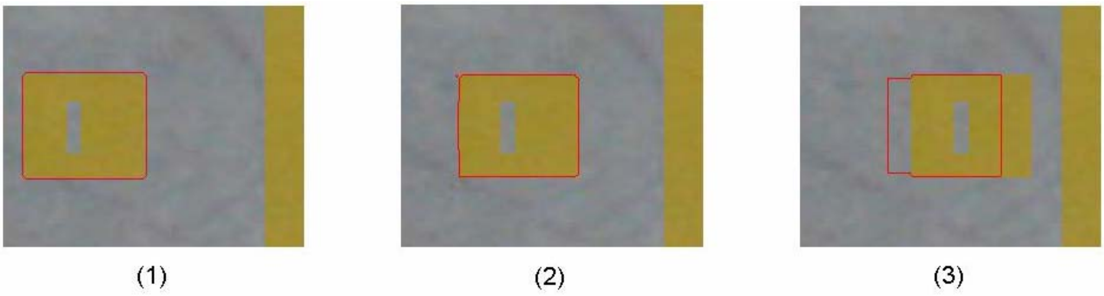


Fig 6.12 Abdol-Reza's model + Shape prior.

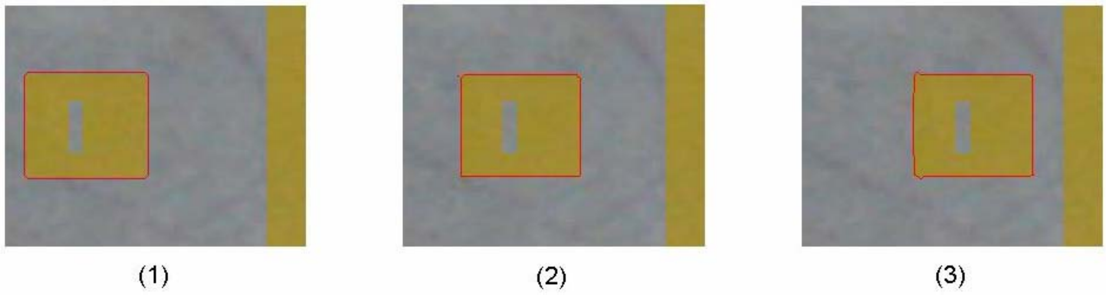


Fig 6.13 Abdol-Reza’s model + Shape prior + Background removal.

6.2. Digital Camera Photos

This experiment presents an image sequence that is composed of digital photos that are captured by a digital camera. The object is manually outlined in frame (a) then transforms gradually in later frames. We will present how the shape prior works when the object is partially involved in the background. We have set η = 10 pixels, and the number of iteration is 80.

Fig 6.14 presents Abdol-Reza's model. In frame (b), when the object rotates a small angle, its bottom-right corner is near a similar background and the contour shrinks there. The shrinking continues in frame (c) and frame (d). Finally in frame (e), after the object moves away from the similar background, the contour cannot recover since some part of the object is regarded as background in frame (d) and that part will result in misclassification in the next frame.

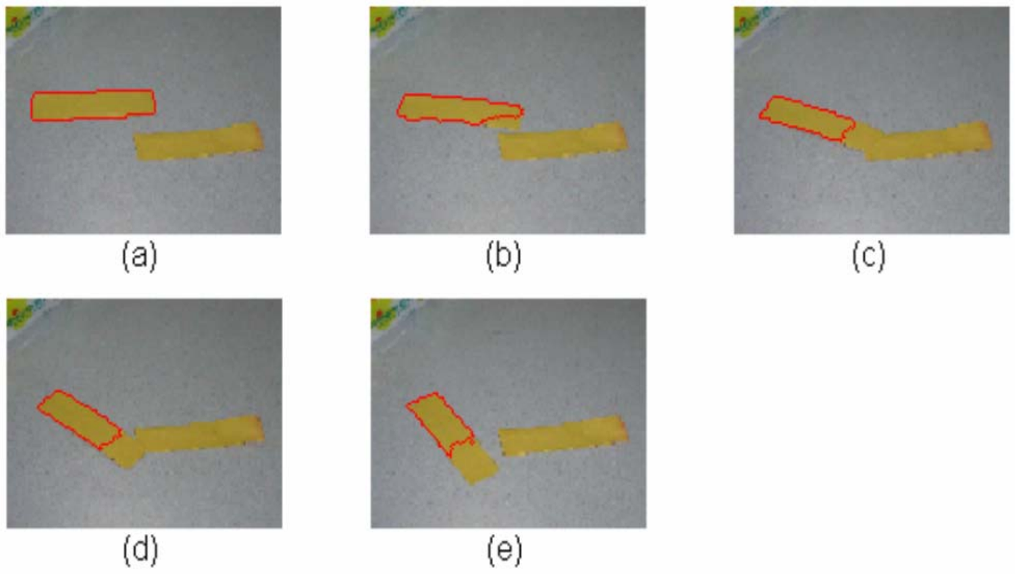


Fig 6.14. Abdol-Reza's model.on digital camera photos

Fig 6.15 illustrates our approach. In frame (b) the bottom-right corner of the contour is prevented from shrinking by the shape prior. In frame (c), the left part of the object is certain and can used to align the shape correctly. After the shape alignment, the involved part of the object can be preserved by the shape. As the same reason, the object can be captured correctly in the later frames. We will present how the shape works in the next figure.

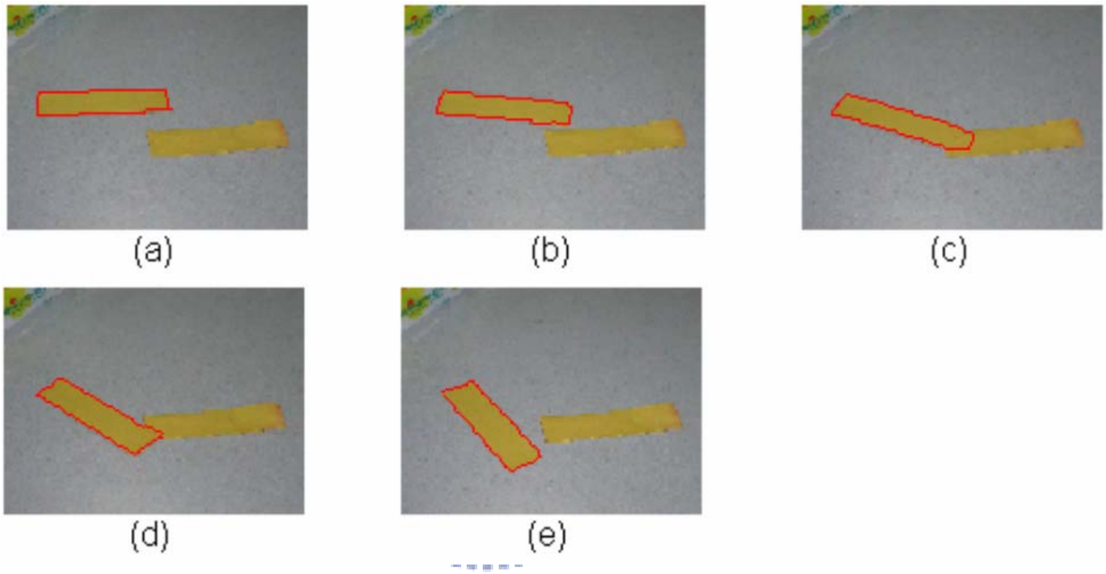


Fig 6.15. Abdol-Reza's model + shape prior on digital camera photos

Fig 6.16 shows how the shape prior works. Figure (a) is previous frame and figure (b) from figure (i) are current frames (To clarify that Fig 6.16 is not an image sequence but the process during contour propagation, we use the term figure (.) or simply (.) instead of frame (.).) The red curve is the propagating contour and the blue curve is the shape prior. The contour and the shape prior are initialized as the result of the previous frame (which is manually outlined in (a) in this illustration). Then the object rotates clockwise with an angle. In (b)(c), the left part of the contour is far from the similar background, which will be excluded by the searching window, is under certainty and goes along with the boundary of the object. And the right part of it moves slowly down. The shape then rotates with a small angle clockwise to accompany with the contour. In (d), the top-right part of the contour has completely shrunk to the object. Also, the direction of the shape prior goes along with that of the object. Translation starts. The right part of the contour is uncertain and is attracted by the forces generated by the shape prior. After the evolution of the contour, alignment is updated in (e). The shape prior moves a little along with its direction and the contour gradually moves toward the shape prior. Continuously, from (f) to (i), the contour gradually matches with the shape prior, which is almost equal to the boundary of the object.

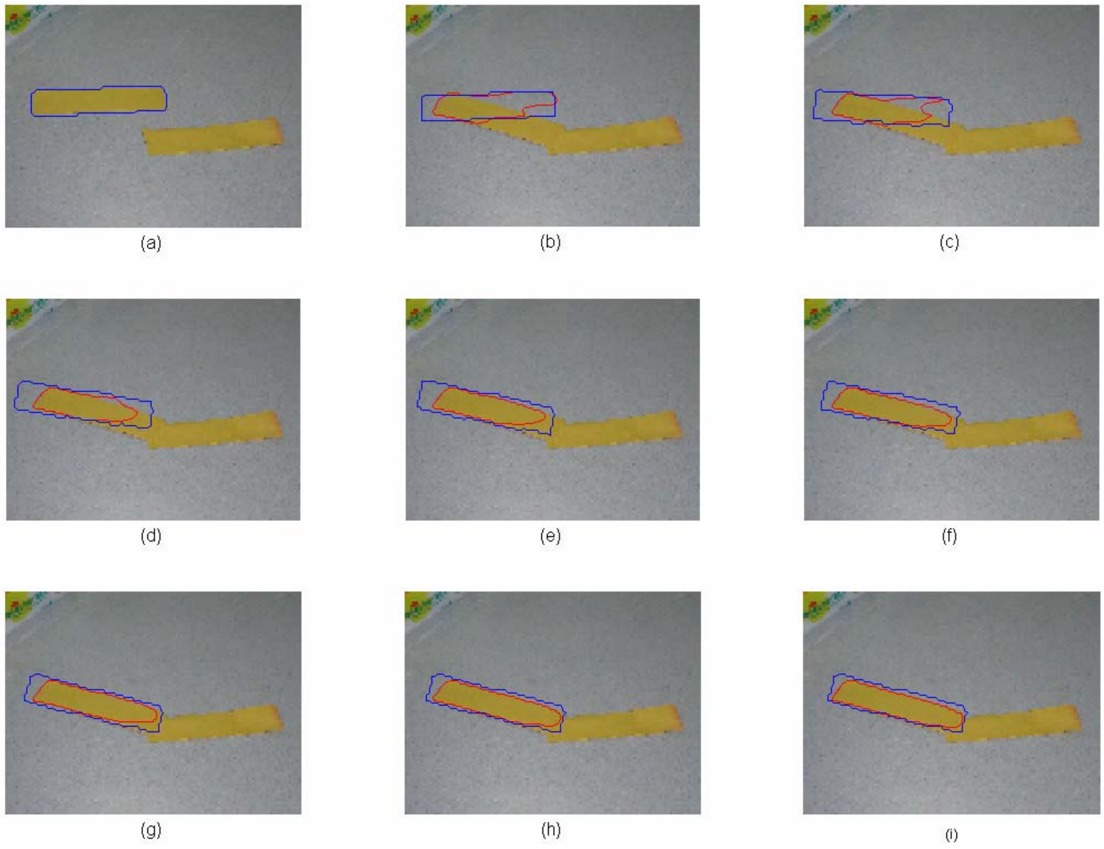
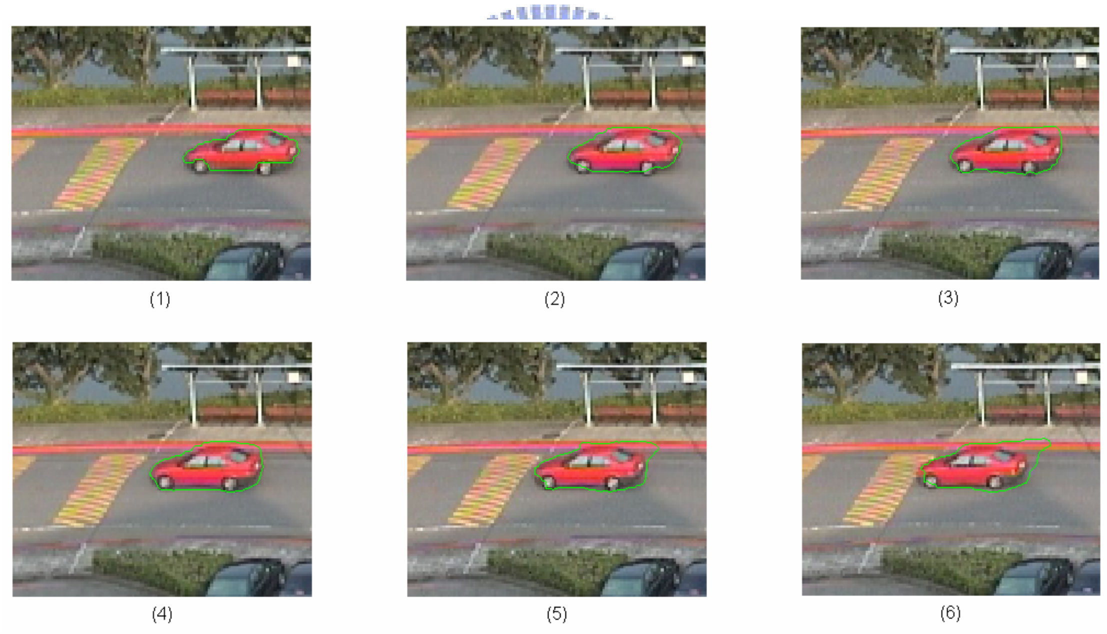


Fig 6.16 Mechanism of shape prior

In the following experiments, the mechanism of the shape prior will work similar to what illustrated in this experiment.

6.3. Real Image Sequences

This challenging figure is processed by Abdol-Reza's model. The car is neighboring the curb which is similar to its color. In frame 2, the contour becomes like ellipse due to the curvature smooth term. Because the gray road is something like the glasses of the car which might be gray somewhere, the uncertainty part is guided by the curvature term. In frame 3 depending on their similarity the background is regarded as the object. Even though the contour is attracted to the car again in frame 4, the error is larger and larger in following frames.


(1) (2) (3)

(4) (5) (6)

Fig 6.17 Abdol-Reza's model

In our method, the top of the car is uncertain and the shape guided the classification. The contour is determined by the head and the tail of the car. Then the shape prior is aligned by these certain parts. Hence we can use it to get a correct capture. The right wheel of the car is included gradually from frame 6 to end. This might because there are some parts of the car like the wheel texture. When the car moves, the wheel is initially included in the next frame but not correctly excluded later.

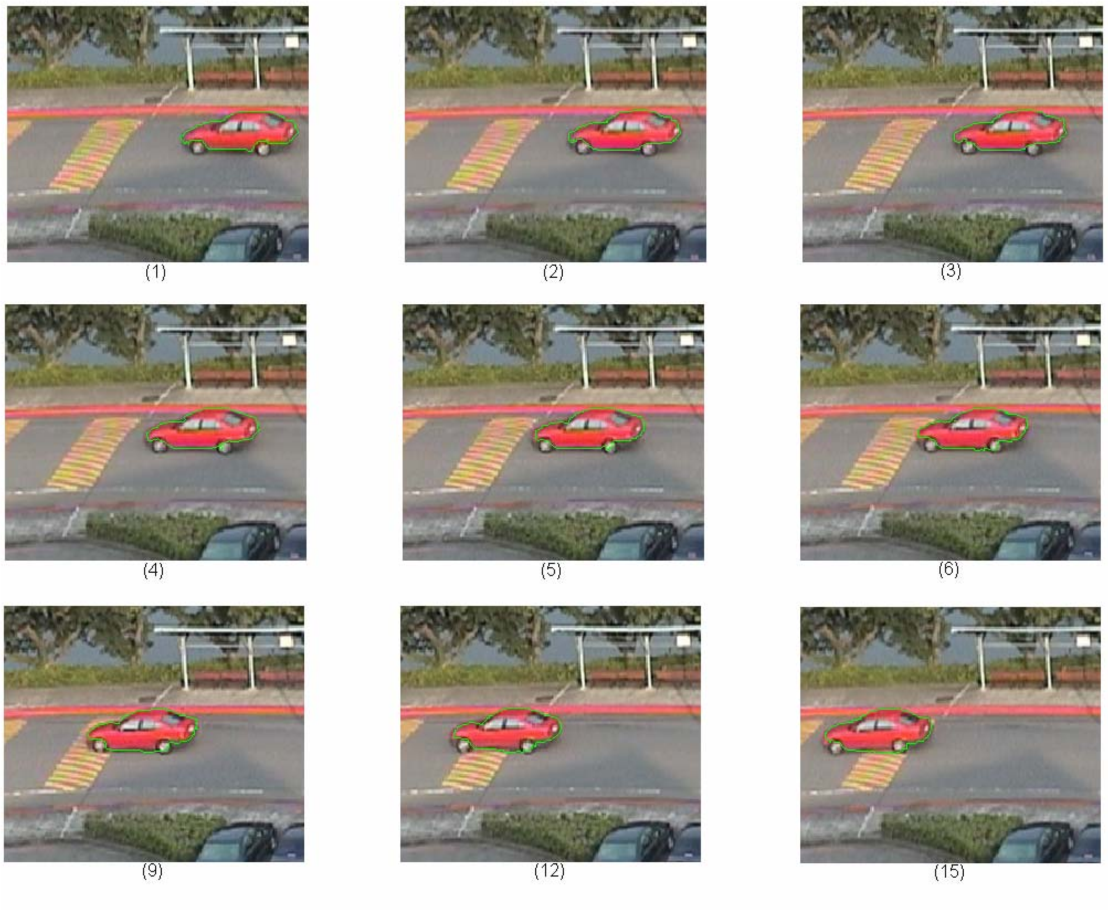
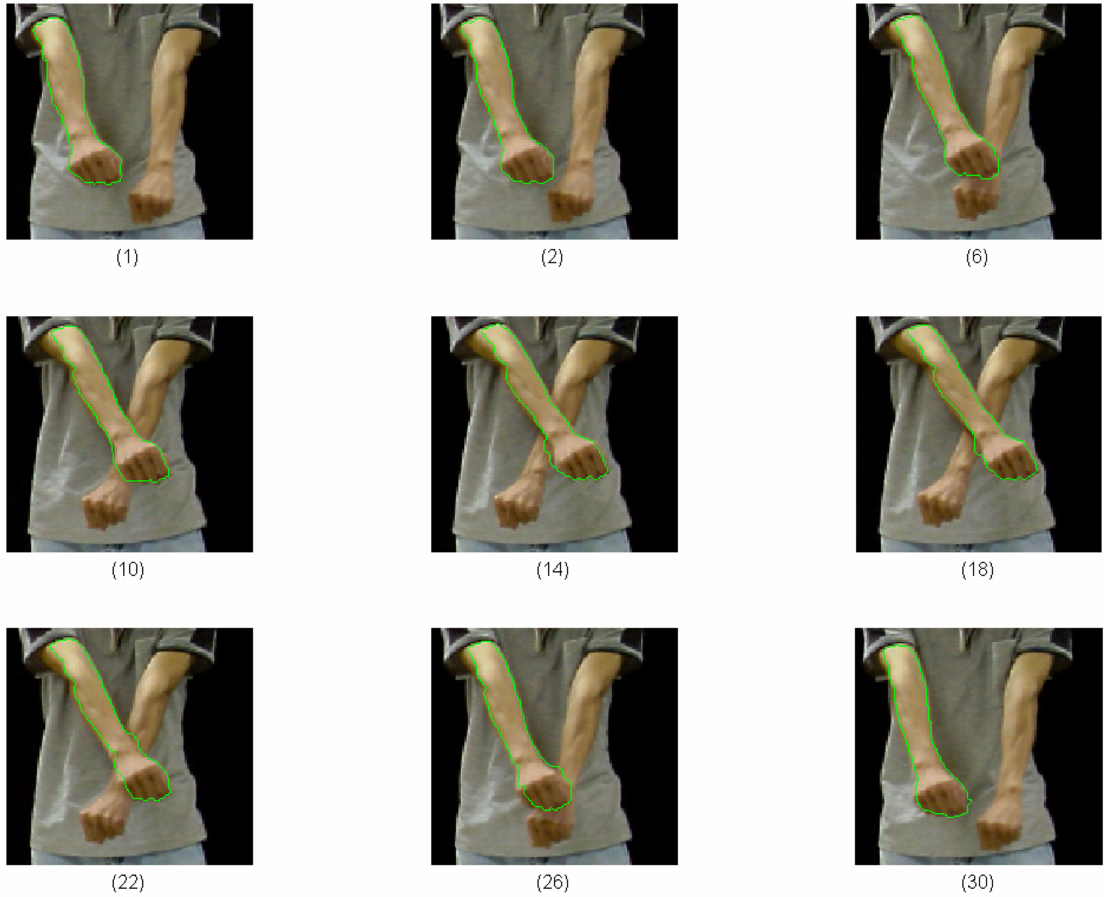

(1) (2) (3)
(4) (5) (6)
(9) (12) (15)

Fig 6.18 Abdol-Reza's model + shape prior

Fig 6.19 presents the result of applying shape prior model on one of the crossing hands. The two hands move toward each other to cross together then leave away. In frame (6), the top-left part of the contour does not capture the arm. This is because the top part of the arm becomes darker due to the light. However, the shape fail to compensate this uncertain part since the top part of the arm also becomes thicker and thicker. In Fig 6.19 (14), the left part to the fist is in fact under uncertainty, which is shown in Fig 6.20 (14), and will let the contour to stay its position. Therefore, the weight of the shape is high on this part and will dominate the tracking behavior. Comparing to Abdol-Reza's model, the shape prevents the contour from sticking out and produces a better result. Similar phenomenon occurs in frame (26). The right part of the fist is not perfectly captured. Again, the shape solves the bad condition. Finally in frame (30), we can see the captured contour is almost identical to the appearance of the arm.


(1) (2) (6)

(10) (14) (18)

(22) (26) (30)

Fig 6.19 Two hands, with shape prior model

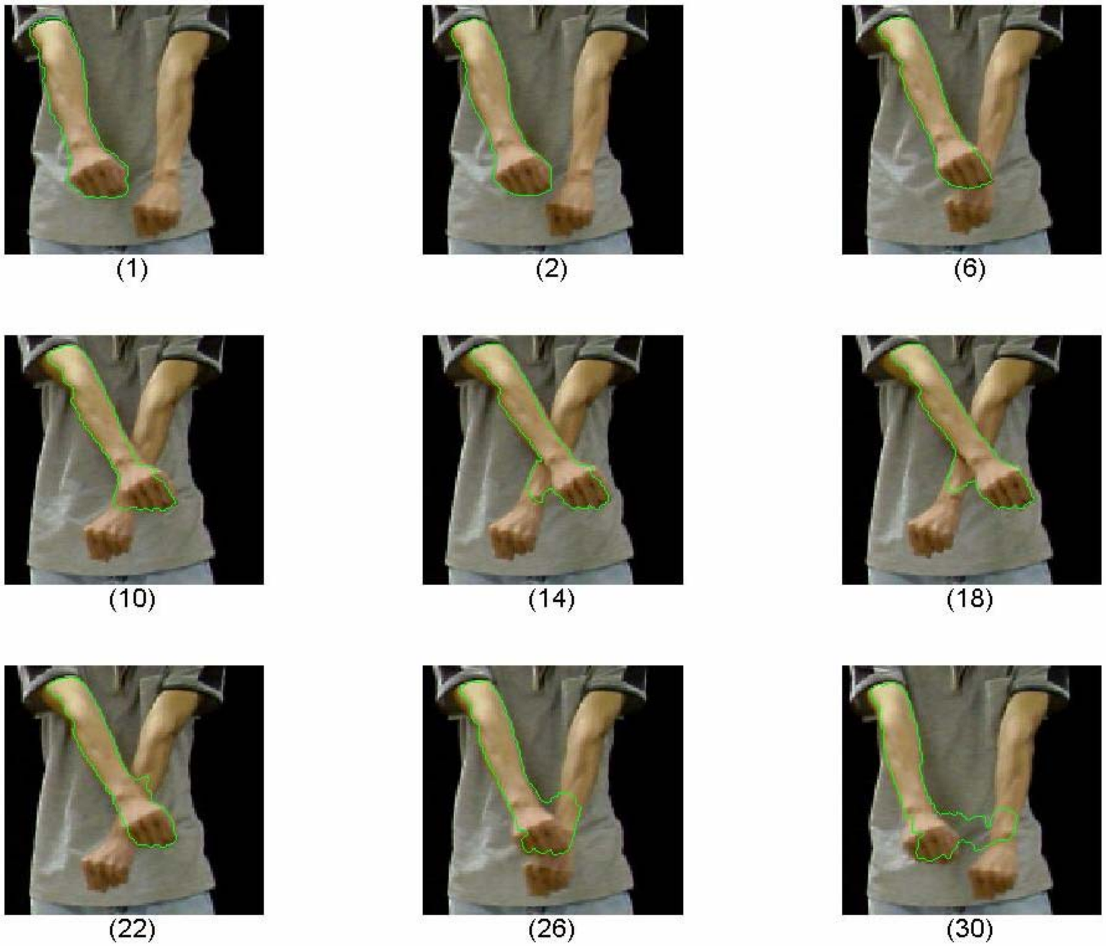
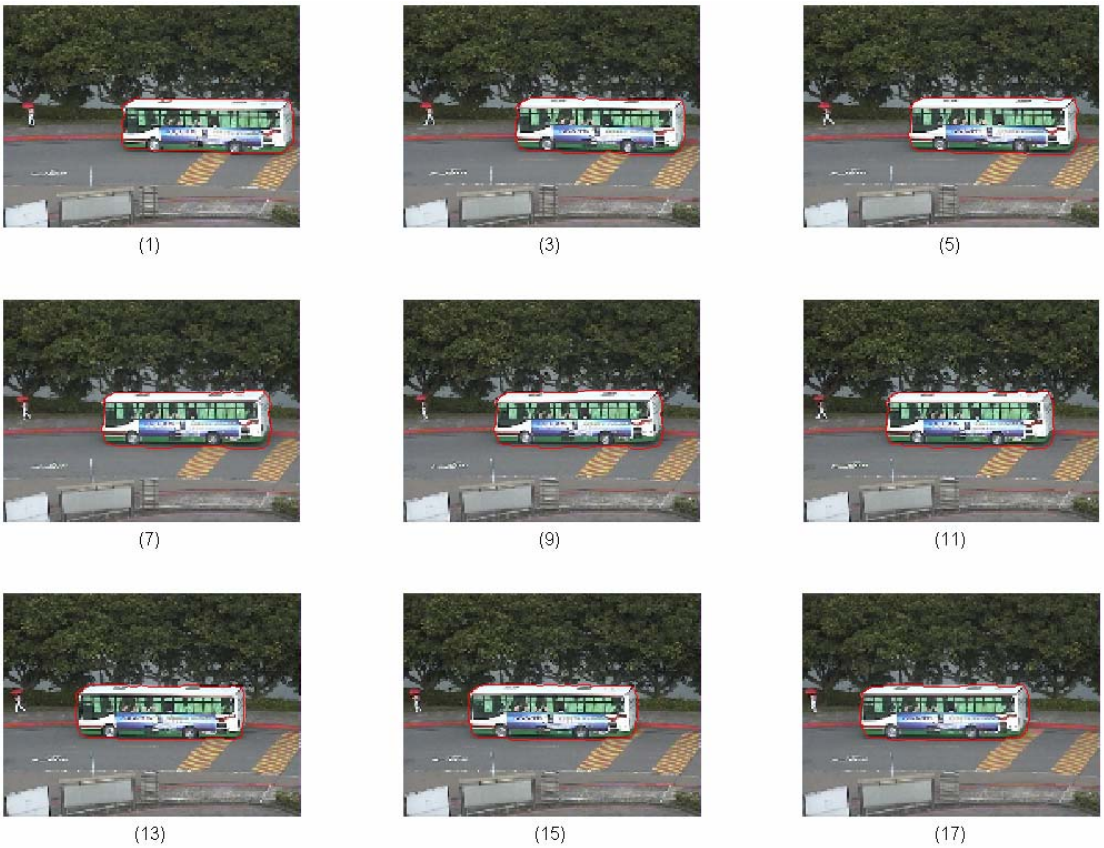


Fig 6.20 Two hands, Abdol-Reza's model

The number of iteration is set to 100 steps in all experiments. In figure 1, the interframe displacement is about 8 pixels. We have therefore set η = 12. Figure 3 is a bus moving backwards with rotation. The yellow part is the key gaudiness for shape alignment. Finally in figure 4, an image sequence on human body is illustrated. Even with large deformation of the object, the tracking can correctly capture the object. In this case, the shape prior takes no effect because the object is obvious apart from the background. Notice the auto changes of the topology handled by level set method.

Fig 6.21 presents the image sequence in which the object appearance is complex. As long as the colors are different from the background, we can easily handle this kind of case.

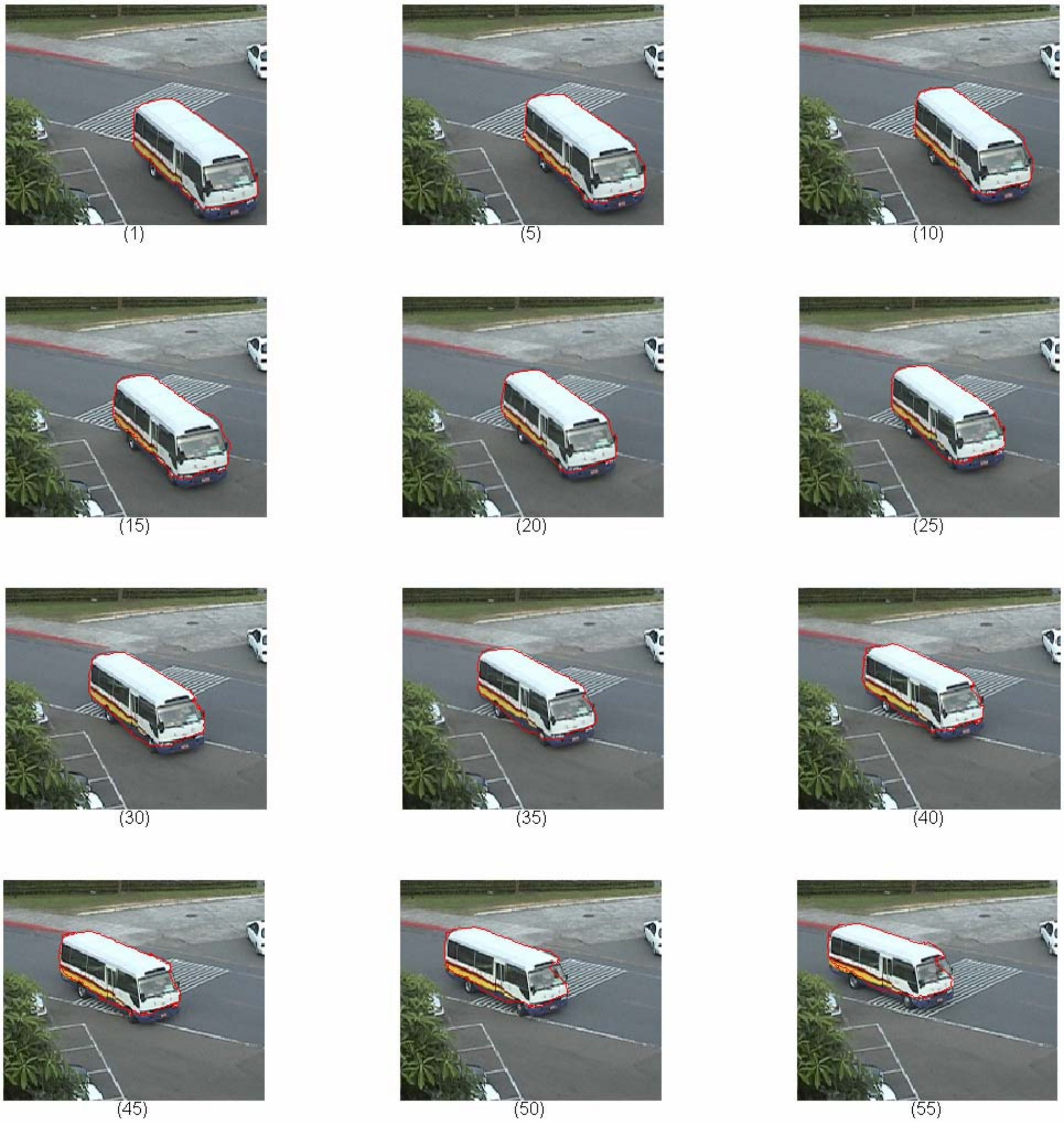

(1) (3) (5)

(7) (9) (11)

(13) (15) (17)

Fig 6.21 A bus with complex appearance

In Fig 6.22, the bus is moving backwards. In frame (50) to (55), the top right corner of the bus is concave. This is because the transparent glass of the bus looks very close to the zebra crossing and the weight of the shape is not strong enough. If we increase the weight of the shape prior, some part of the contour might be over attracted by the shape prior due to their uncertainty.



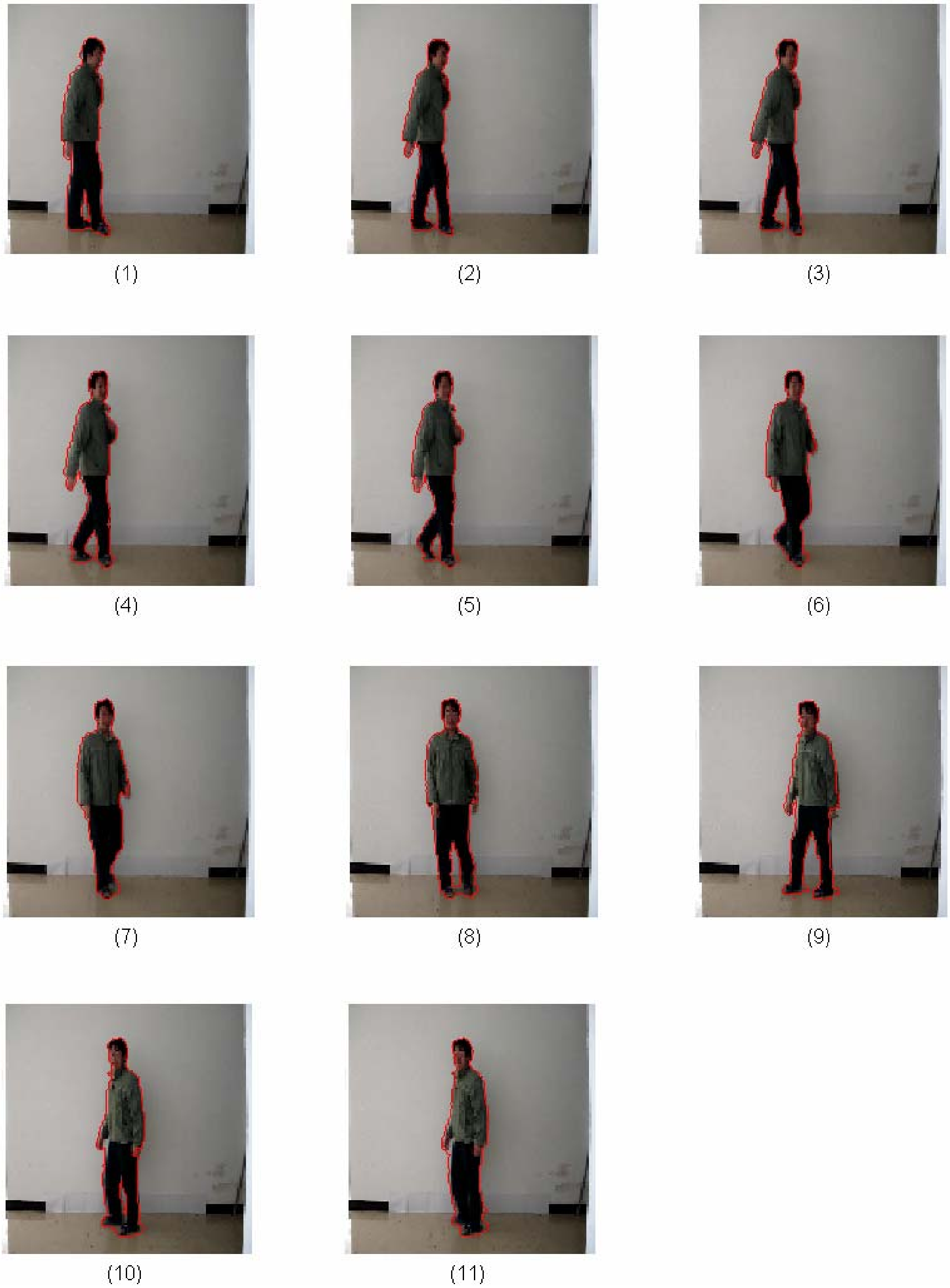
(1) (5) (10)

(15) (20) (25)

(30) (35) (40)

(45) (50) (55)

Fig 6.22 A bus



(1) (2) (3)

(4) (5) (6)

(7) (8) (9)

(10) (11)

Fig 6.23 Human tracking

6.4. Parameter Settings

Weight of Shape Prior

This experiment presents the influence of the shape prior under different weighting function. Red lines are propagating contours and blue lines are shape priors. The number within the parenthesis below each figure is the number of iterations. We have set $\lambda = 0$ and $\beta = 1$for all figures in this section. Notice that the shape is transformed to align with the contour during the propagation.

Although the weight $w(\gamma(s))$ of the shape prior is automatically calculated from pixel to pixel, it is indeed embedded a threshold α. α can be considered as the average weighting of the shape prior and $w(\gamma(s))$ can be considered as α multiplied by a scalar which is corresponding to the discriminate analysis of Abdol-Reza's model. When α is larger and larger, the average weighting of the shape prior becomes higher and higher and more and more parts of the contour will be attracted by the shape prior. To insure that deterministic parts of the contour to dominate the tracking, a reasonable, or not too large, α is needed. In this case, $\alpha = 0 \sim 1000$ is a good choice.

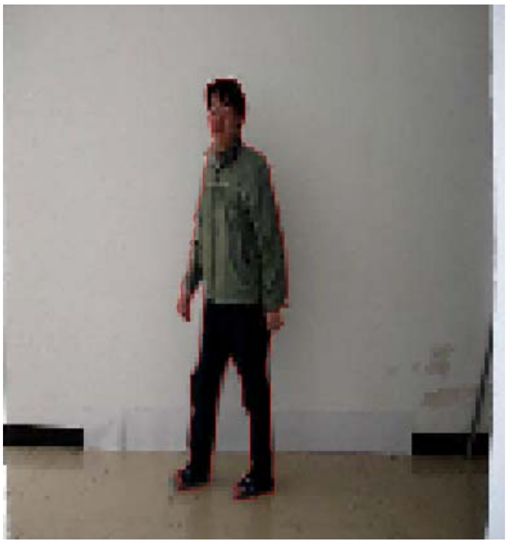
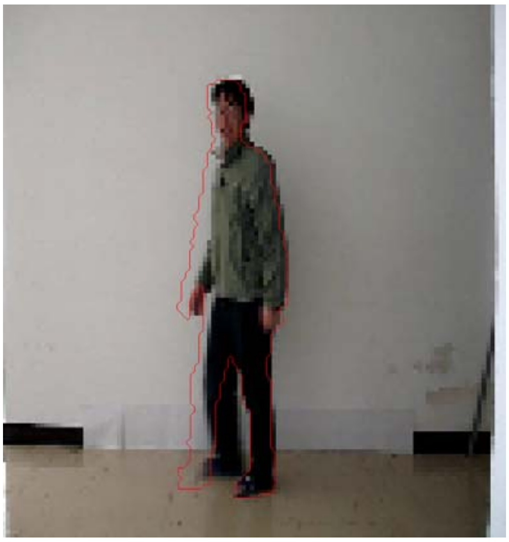
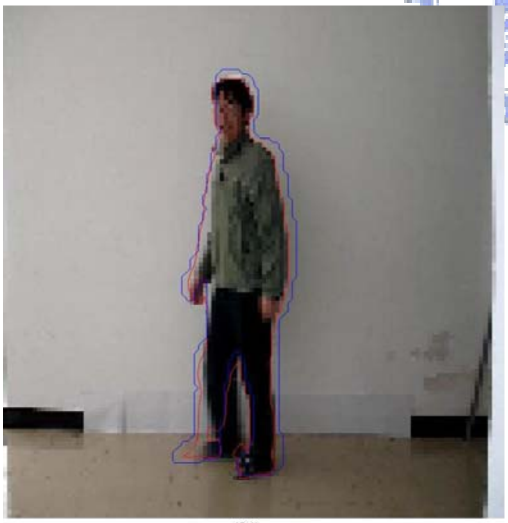
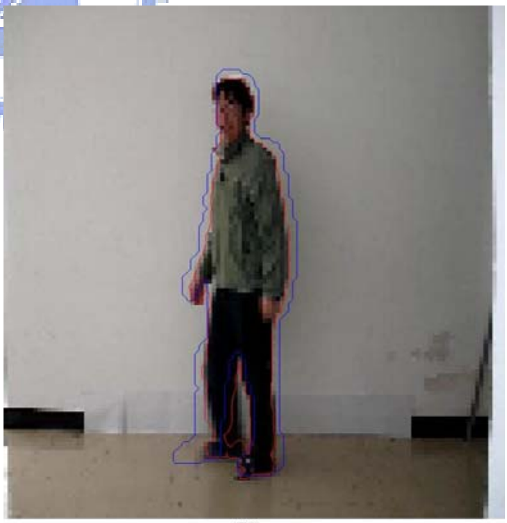
 The previous frame (Red contour is the final contour of the result)	 The current frame (the red contour is its initial contour obtained from the previous frame).

Fig 6.24 Initial contour

 (34)	 (74)

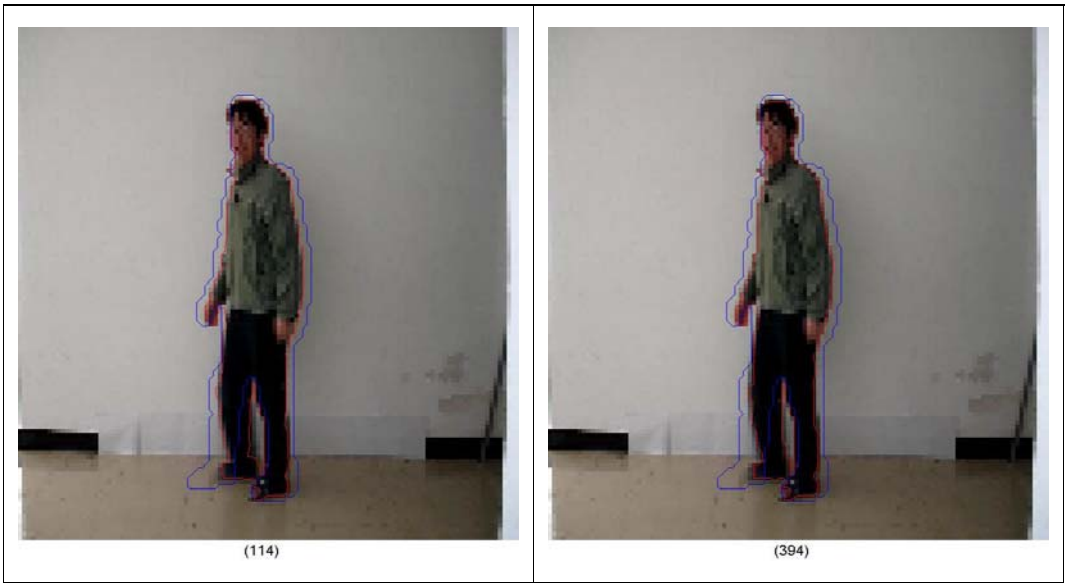
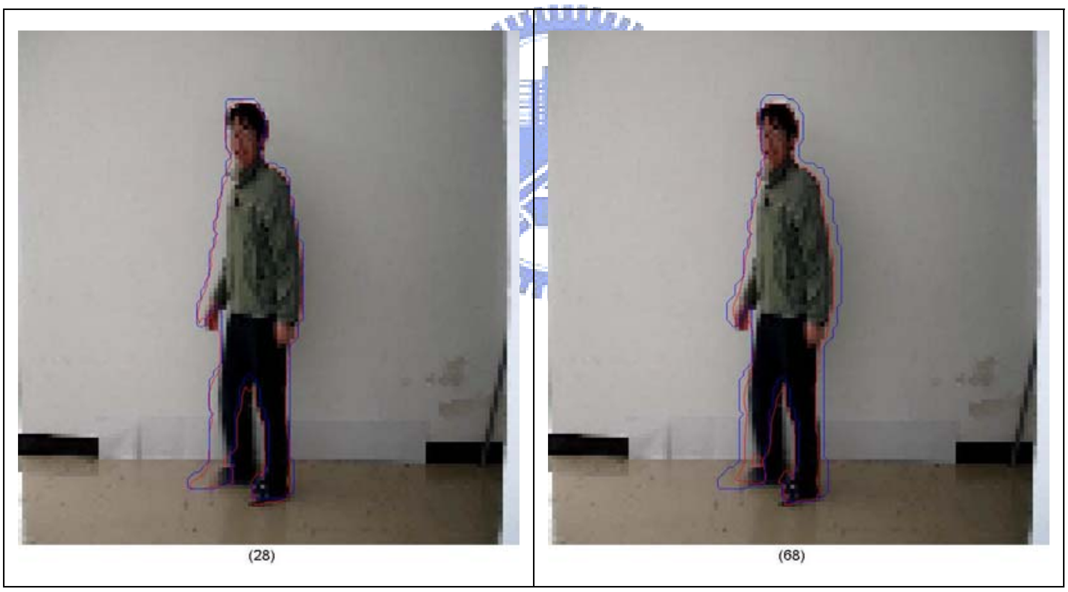
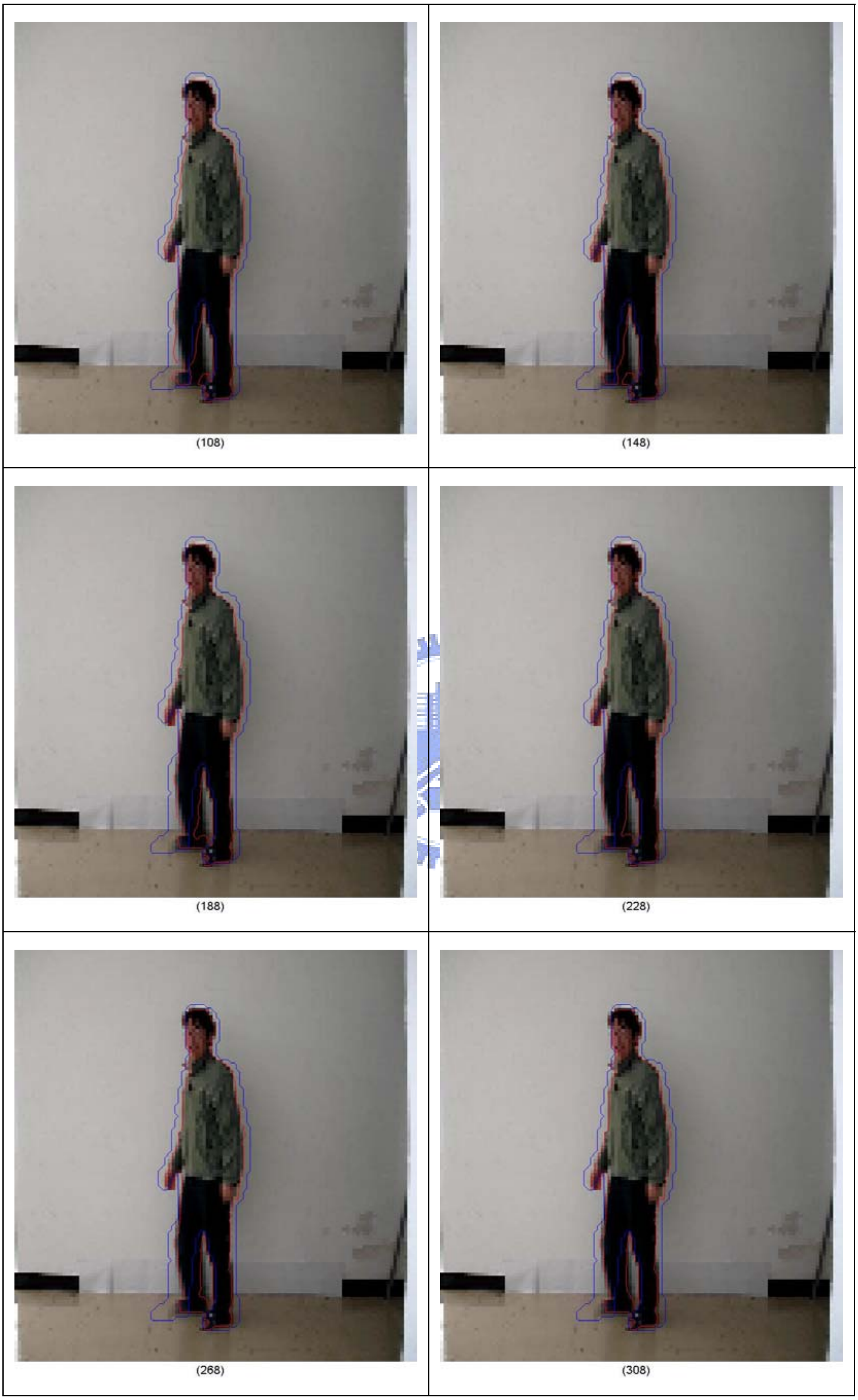

(114)
(394)

Fig 6.25 α = 500

The intermediate results of the tracking process with weights α = 500, and β =1.

The last diagram indicates the final contour in red.


(28)
(68)



(108)

(148)

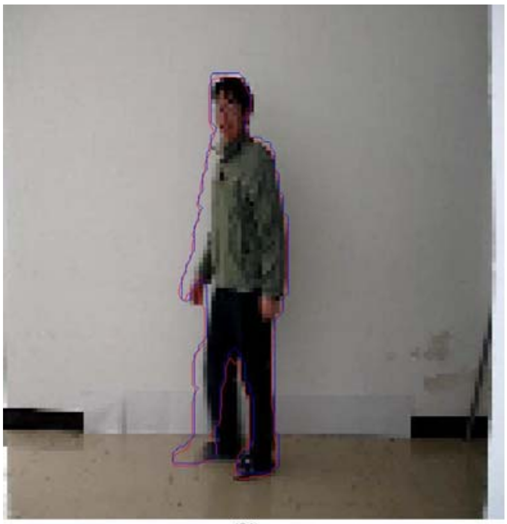
(188)

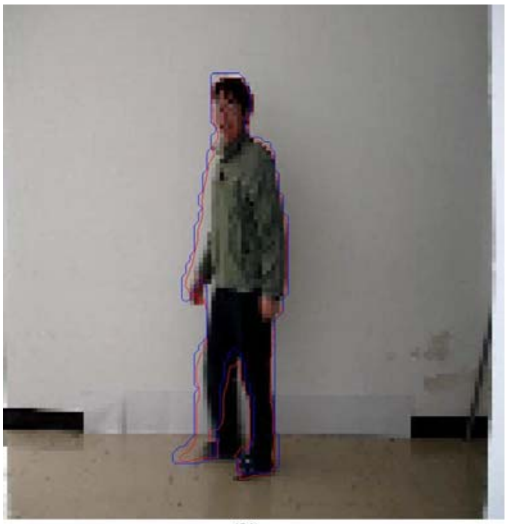
(228)

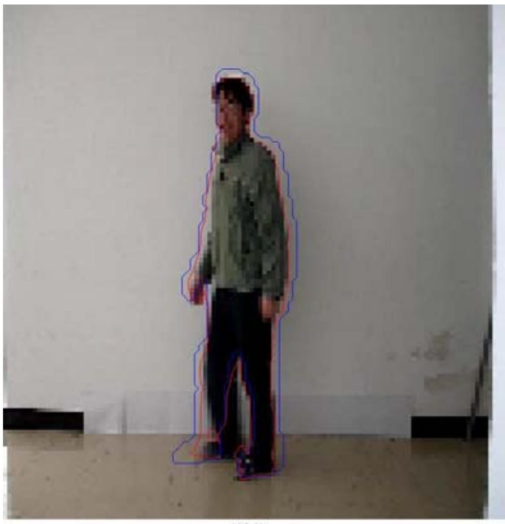
(268)

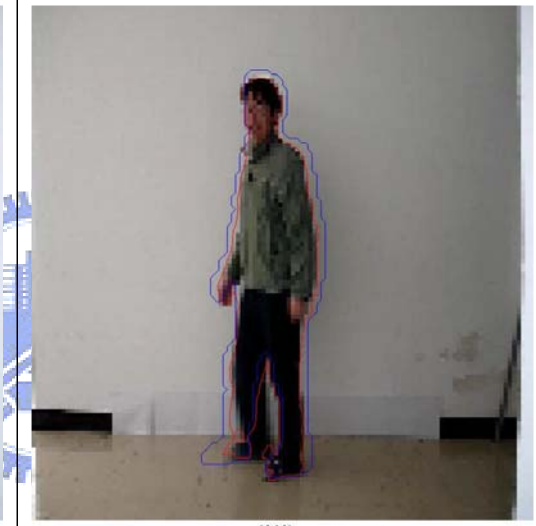
(308)

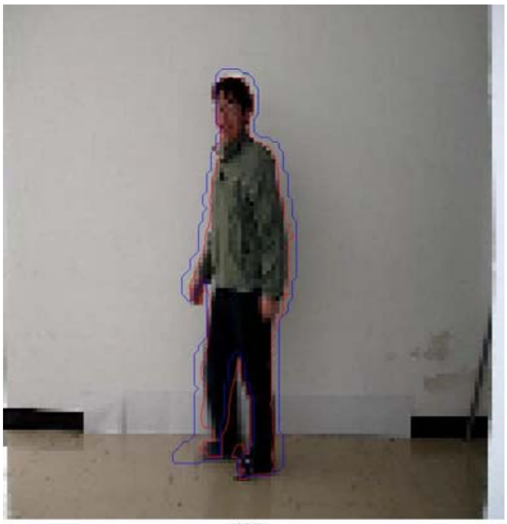
Fig 6.26 α = 1000

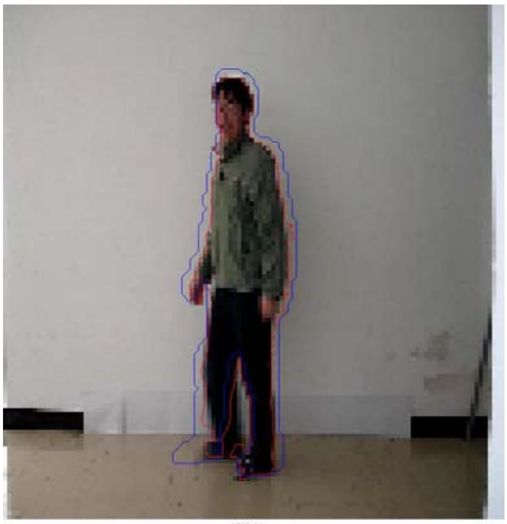

(21)

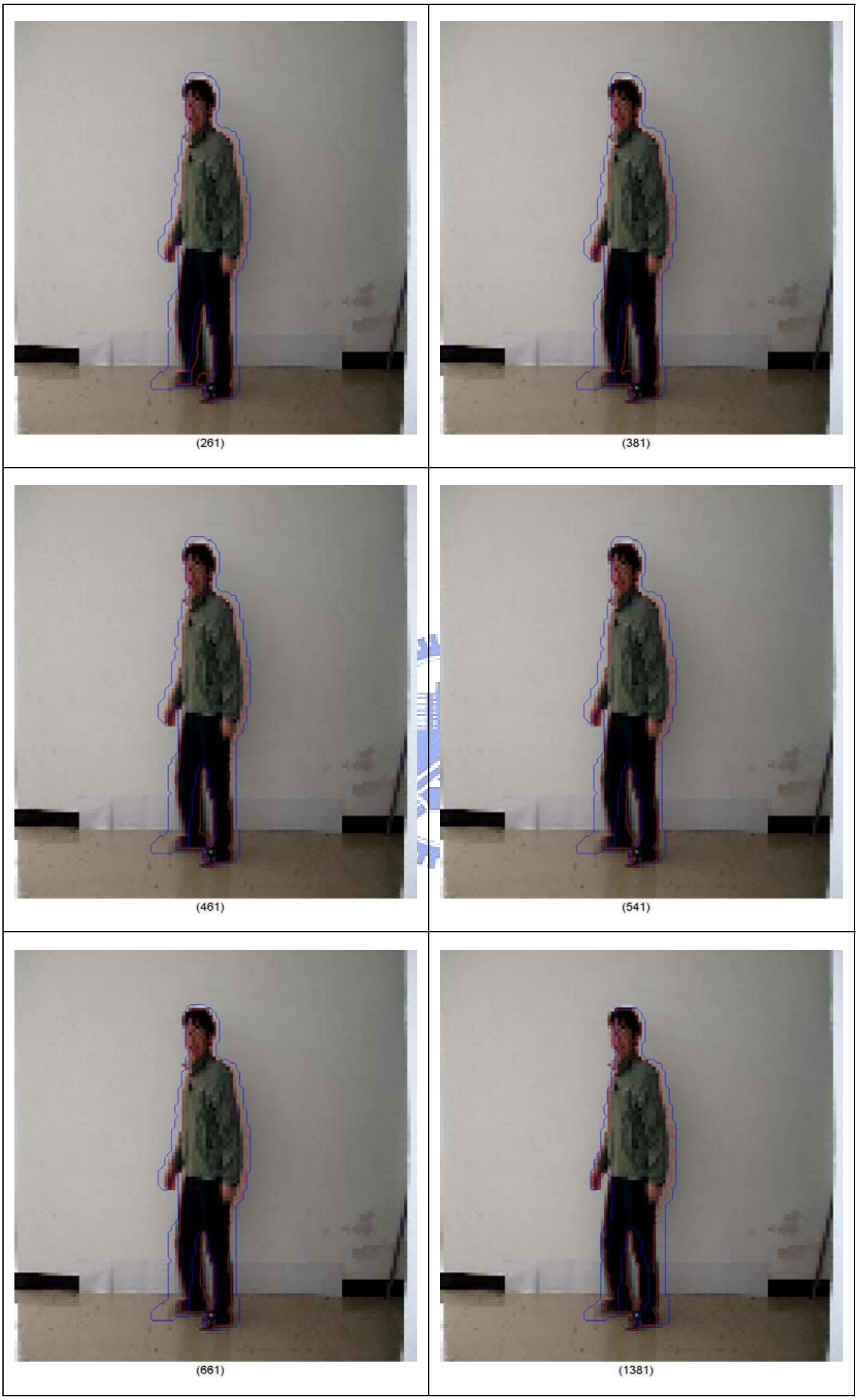

(61)


(101)


(141)


(181)


(221)



(261)

(381)

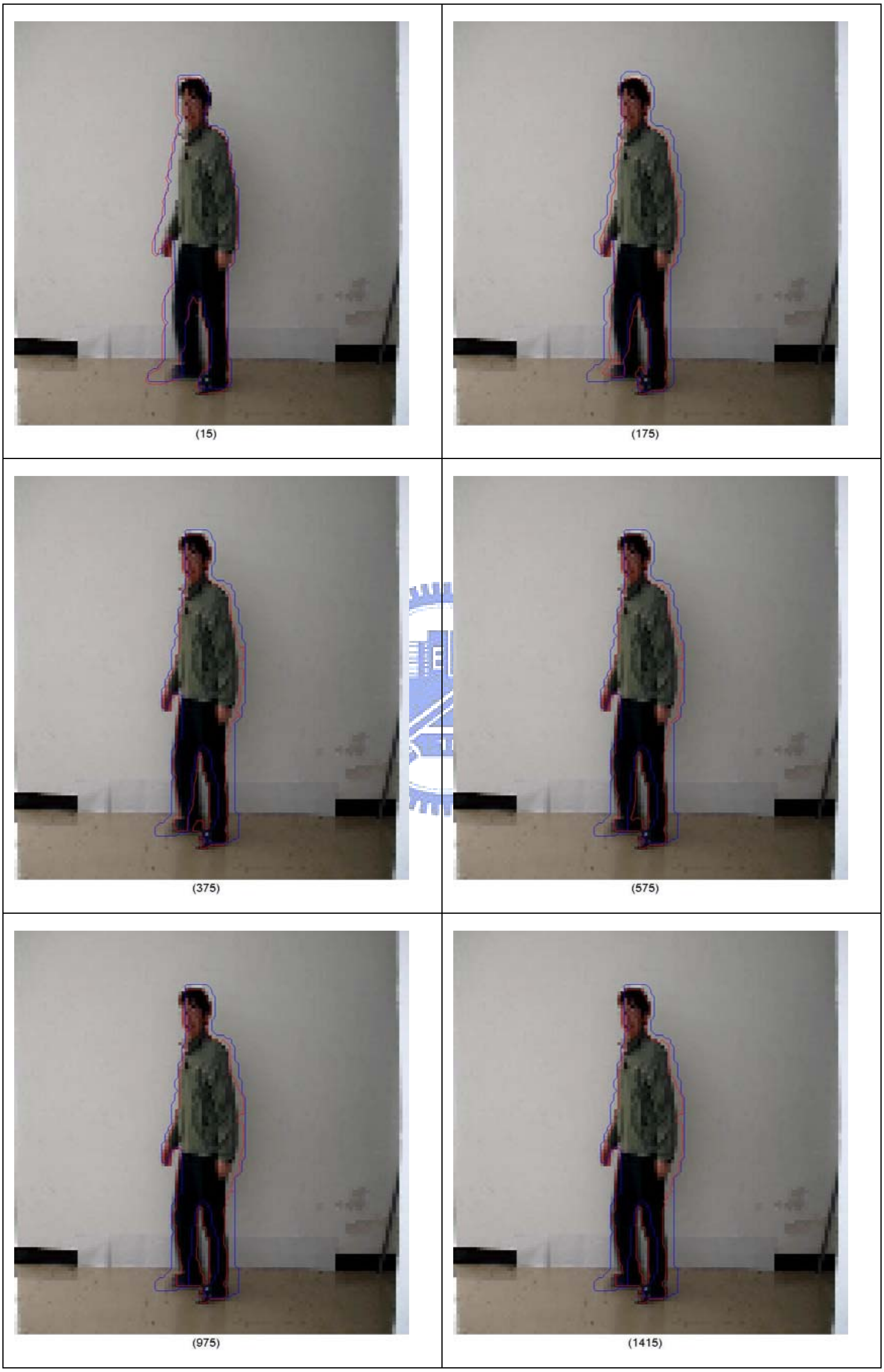
(461)

(541)

(661)

(1381)

Fig 6.27 α = 1500



(15)

(175)

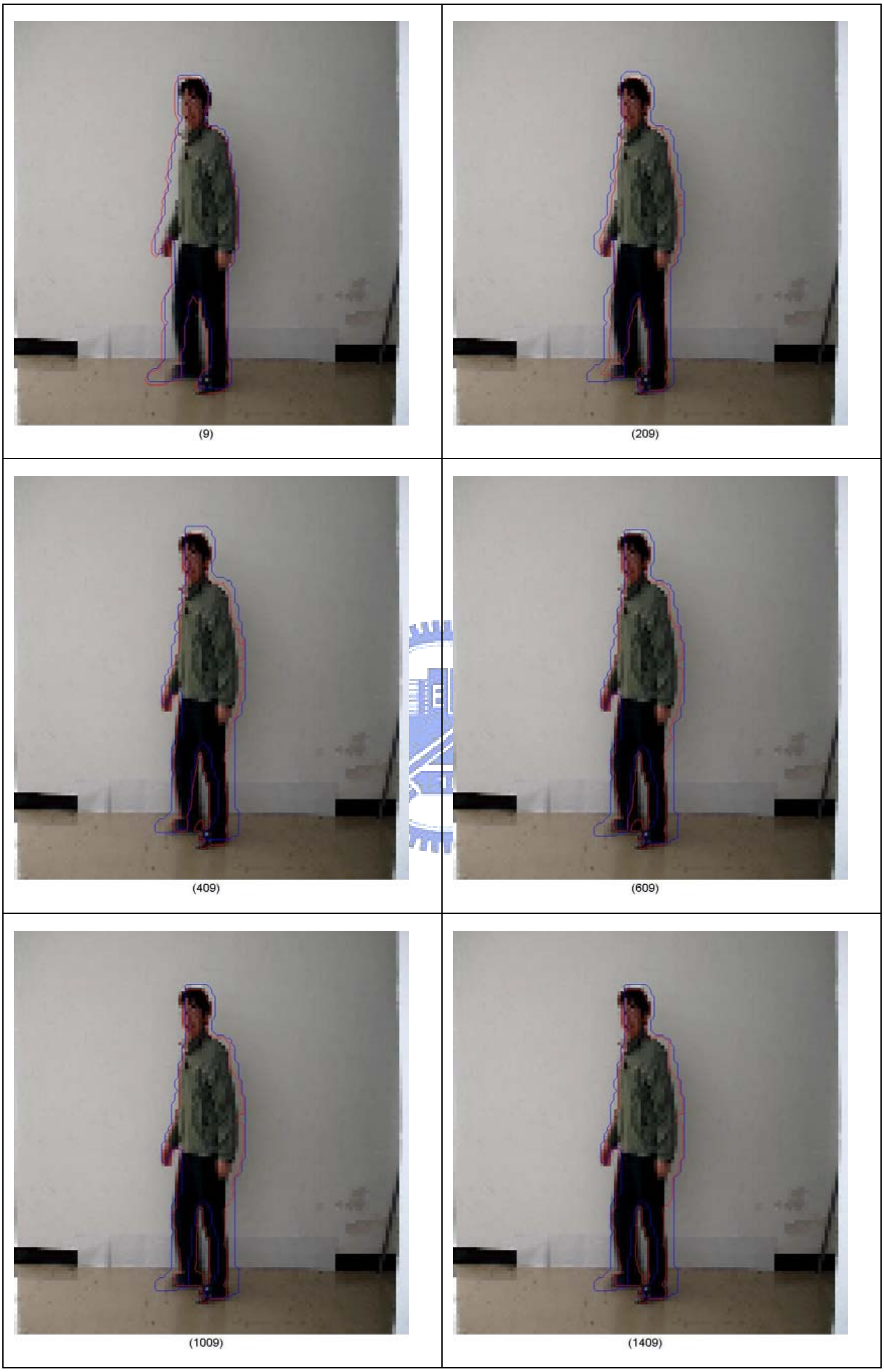
(375)

(575)

(975)

(1415)

Fig 6.28 α = 2000



(9)

(209)

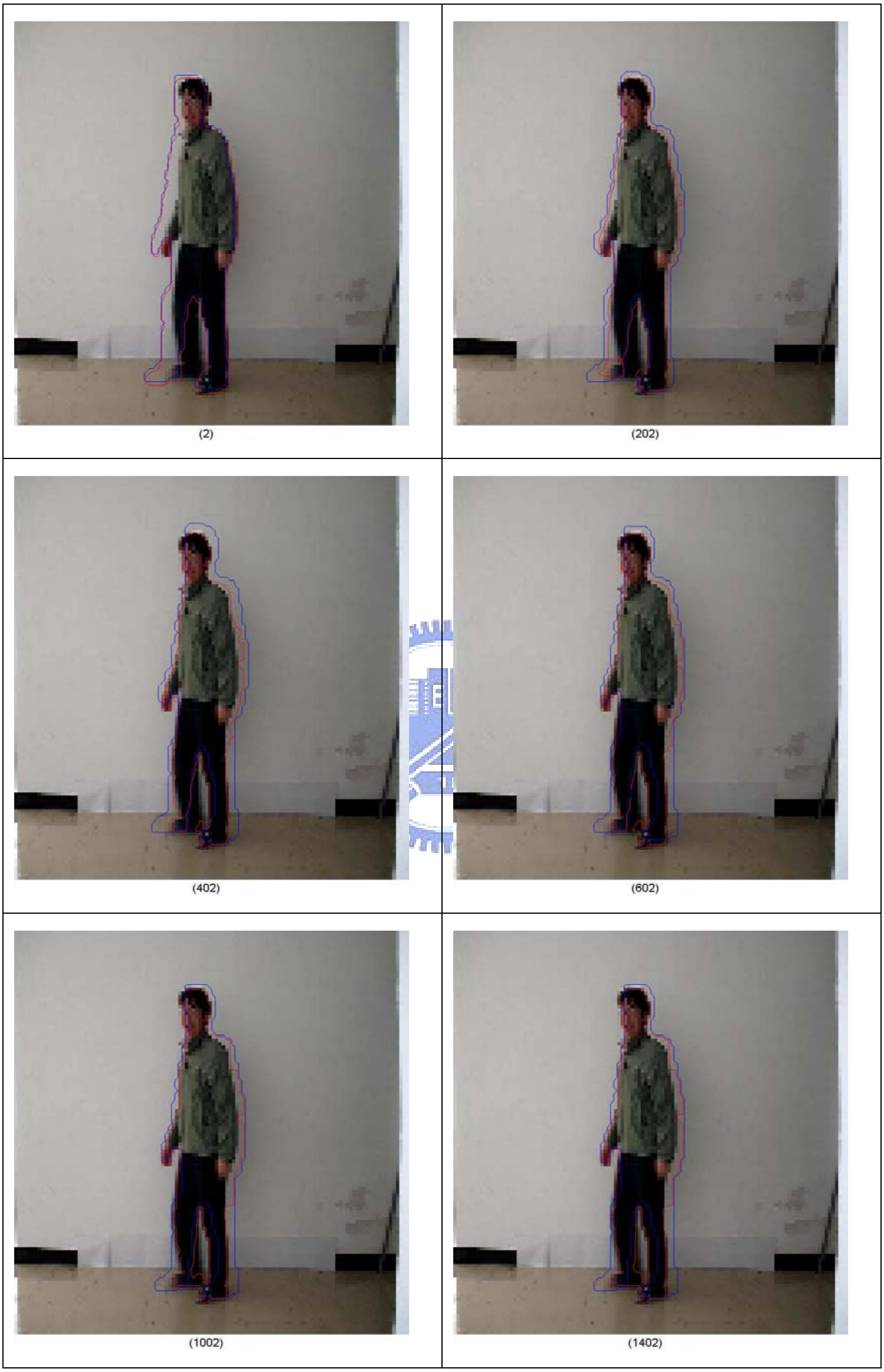
(409)

(609)

(1009)

(1409)

Fig 6.29 α = 2500



(2)

(202)

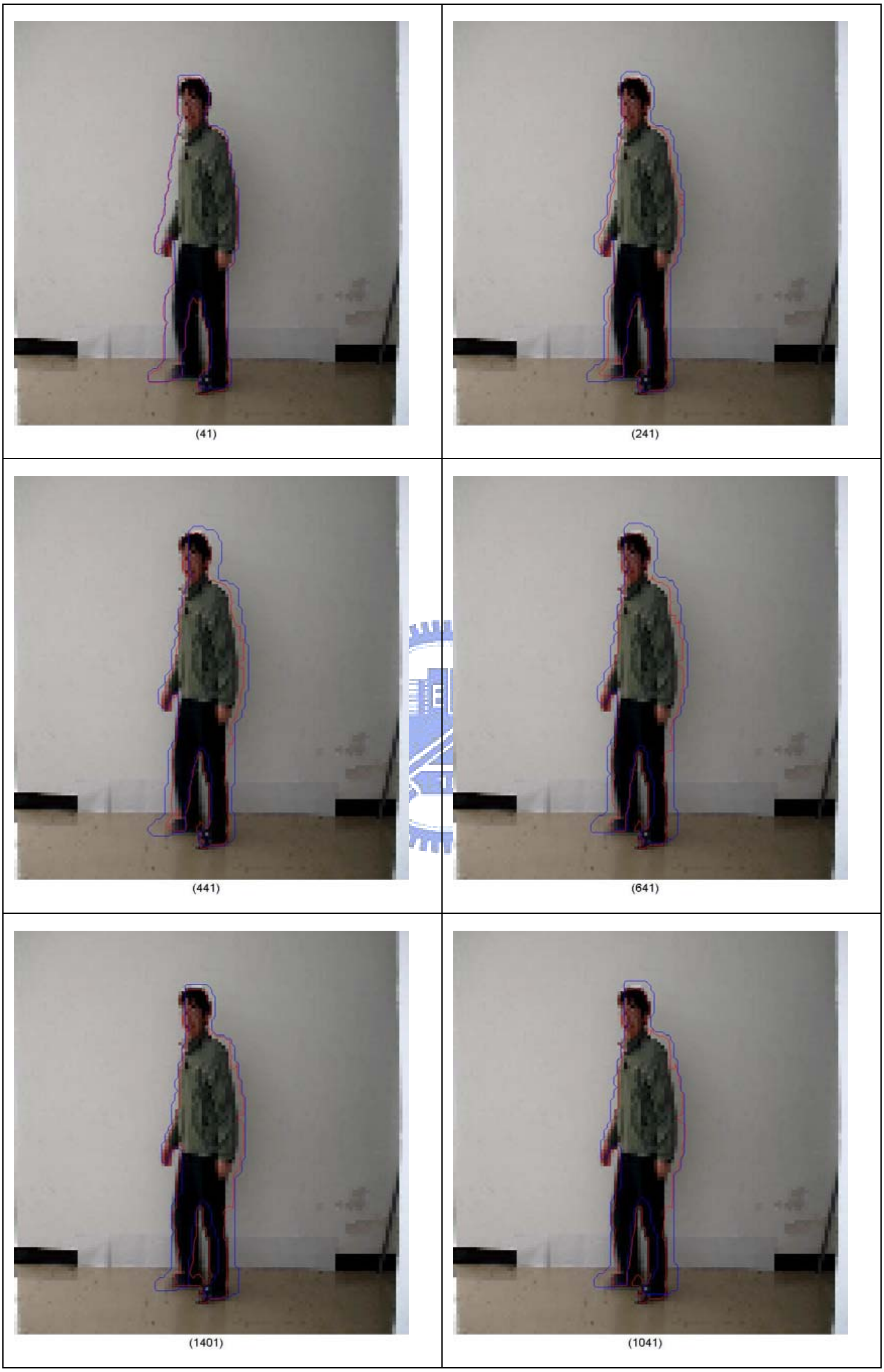
(402)

(602)

(1002)

(1402)

Fig 6.30 α = 3000



(41)

(241)

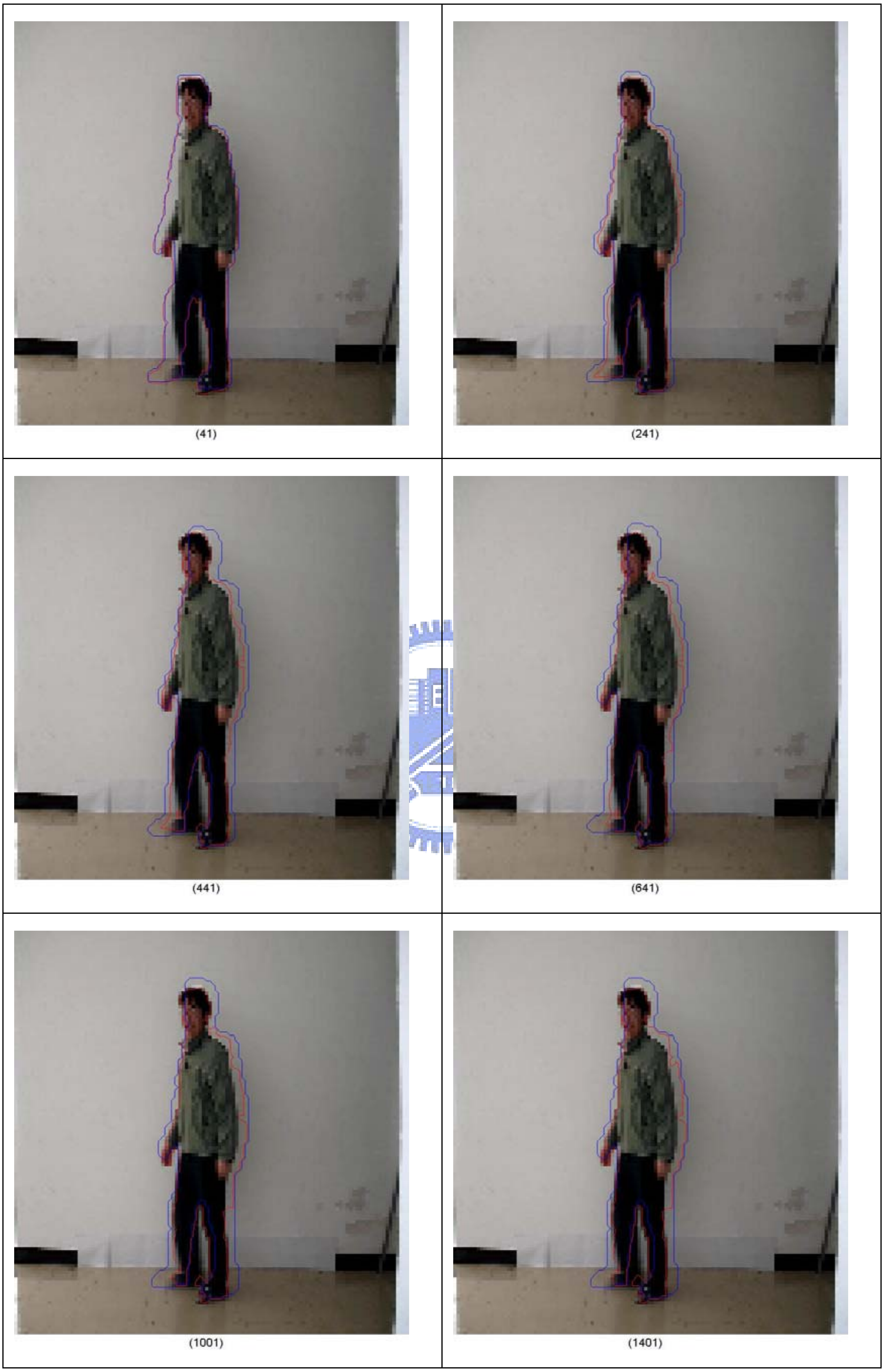
(441)

(641)

(1401)

(1041)

Fig 6.31 α = 3500



(41)

(241)

(441)

(641)

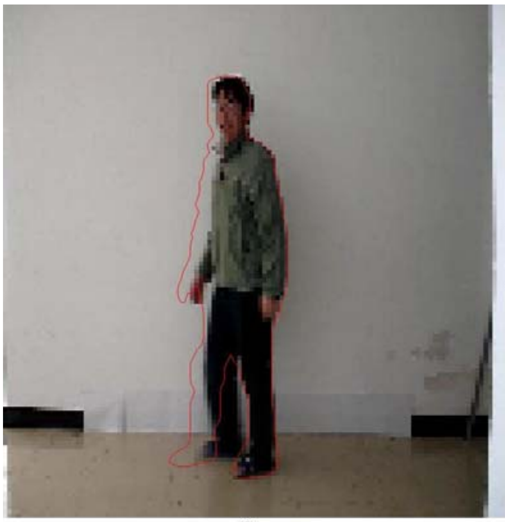
(1001)

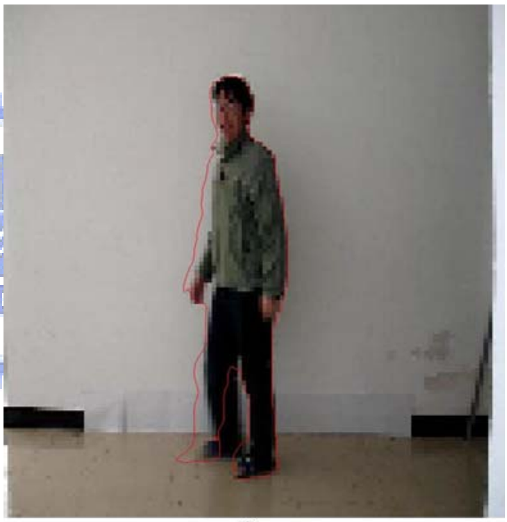
(1401)

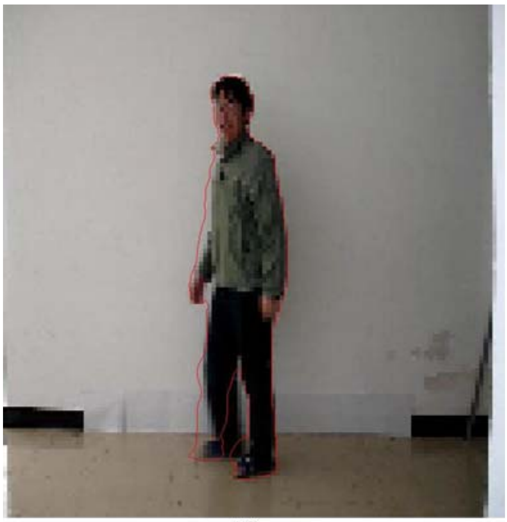
Fig 6.32 α = 4000

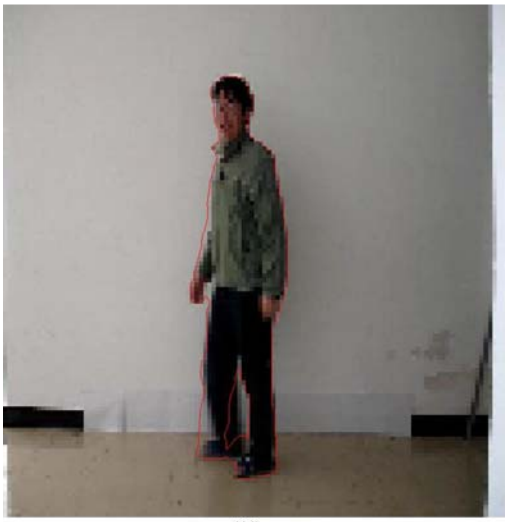
The Curvature Term

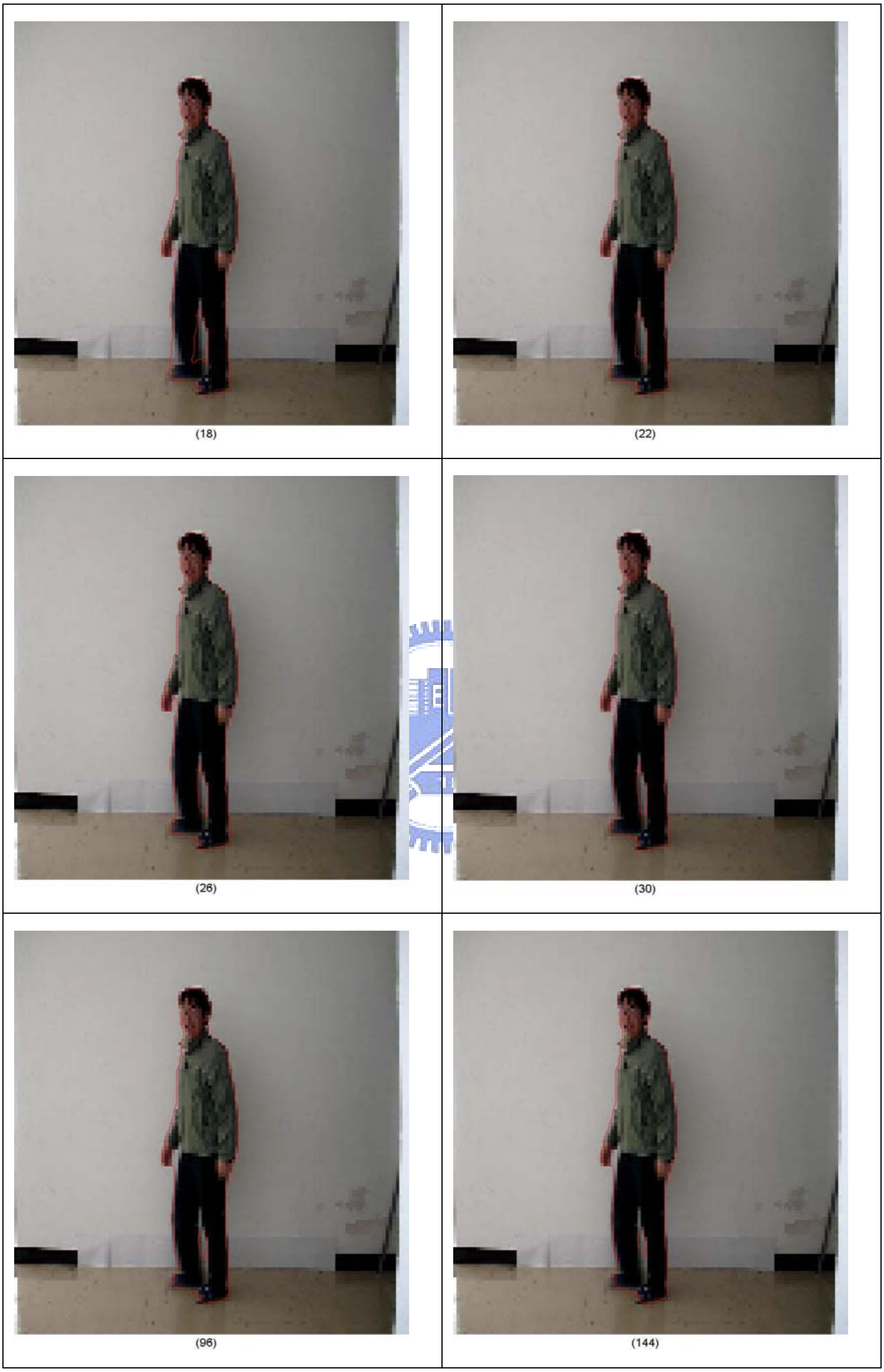
This experiment presents the effect of the curvature term. We adjust λ from 0.01 to 0.13 and have set α = 0 and β = 1 for all figures. The initialization of the contour is the same as previous sub experiment. The purpose of the curvature term is to smooth the contour. We can see that when λ =0.13, the hole of the contour between two legs is dominated by the curvature term and cannot capture correctly anymore. Also, several other parts are over smoothed such that the contour cannot deep to the detailed concave part of the object.


(2)


(6)


(10)


(14)



(18)

(22)

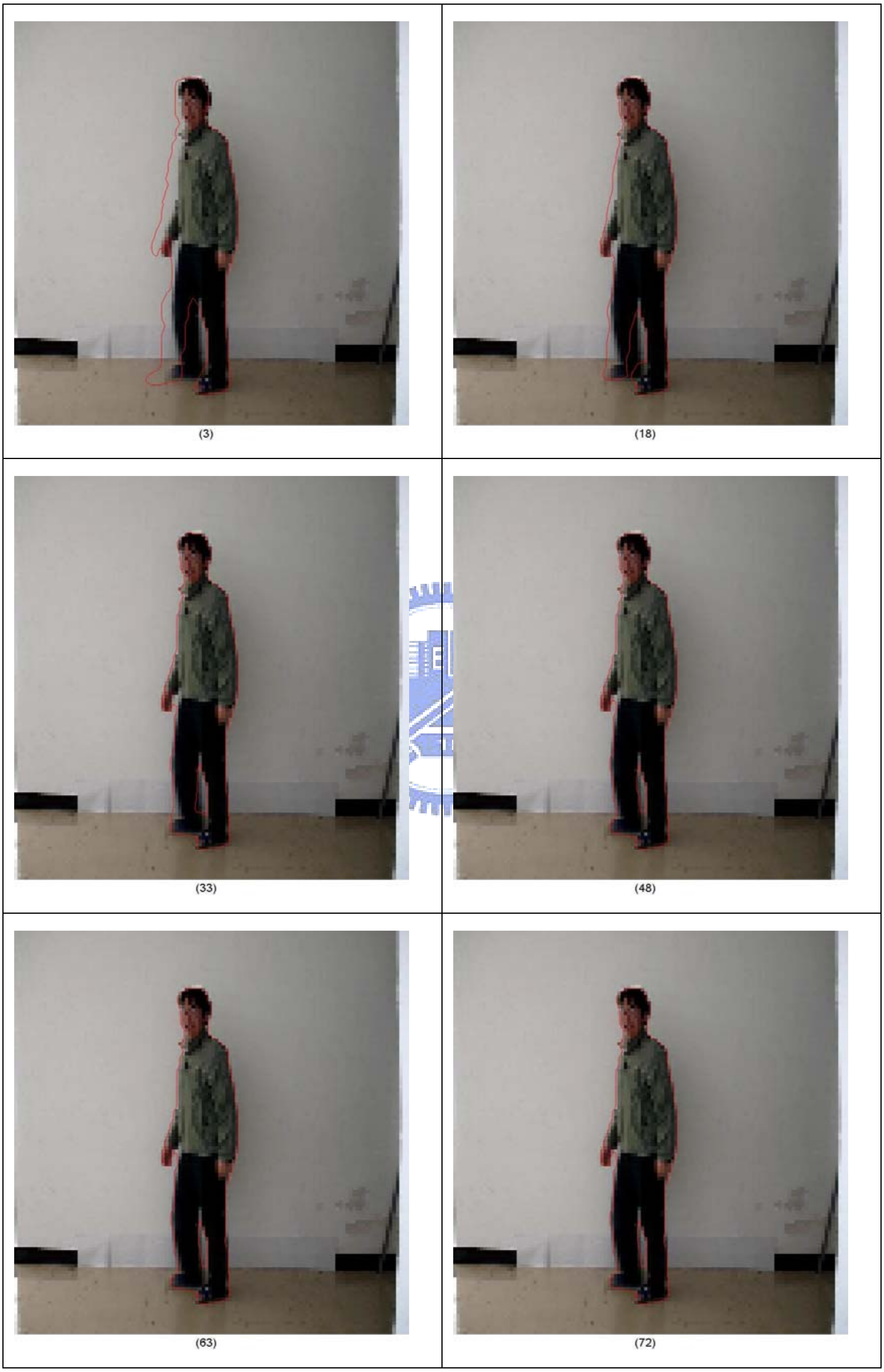
(26)

(30)

(96)

(144)

Fig 6.33 λ =0.01



(3)

(18)

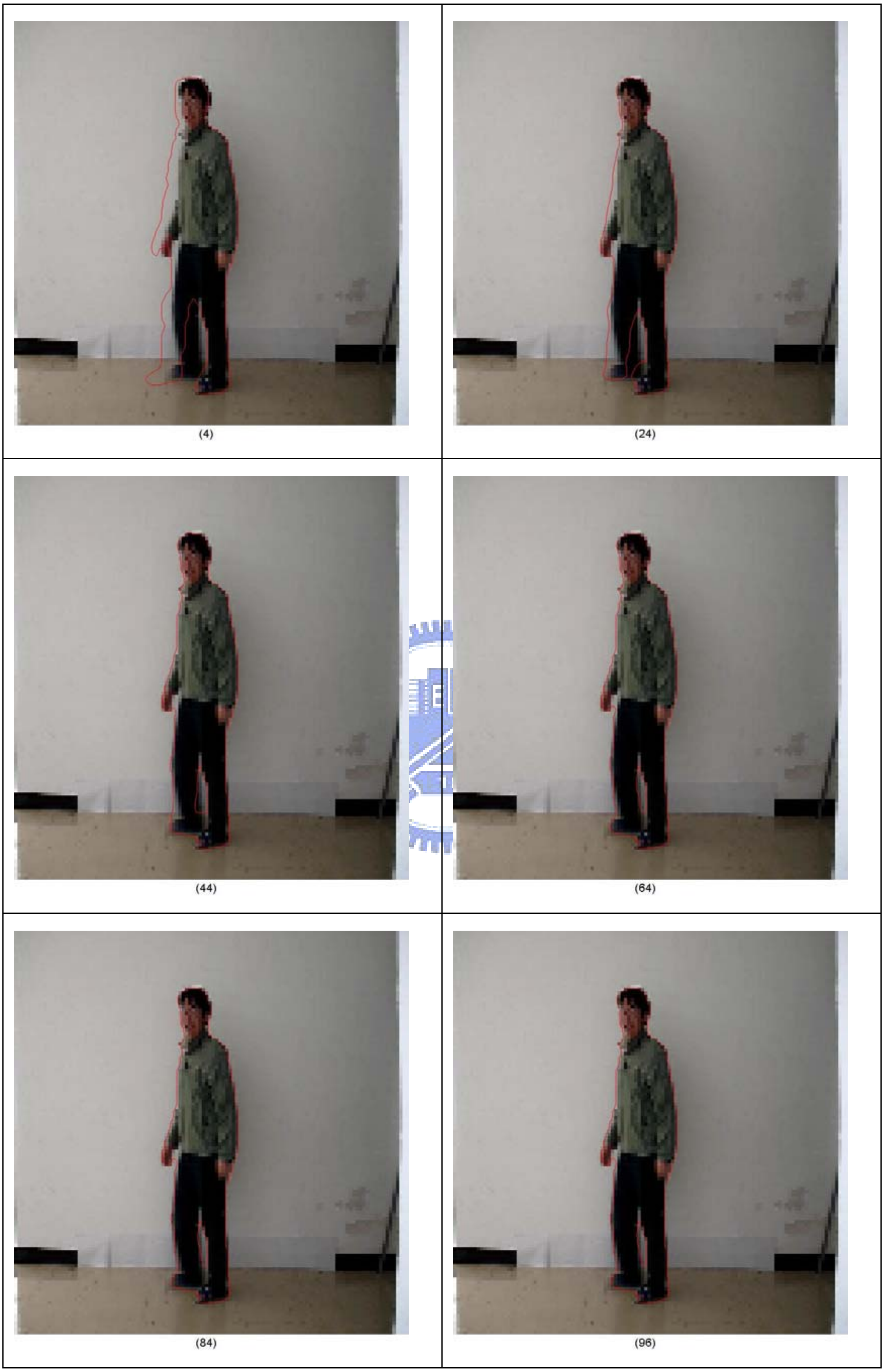
(33)

(48)

(63)

(72)

Fig 6.34 λ =0.03



(4)

(24)

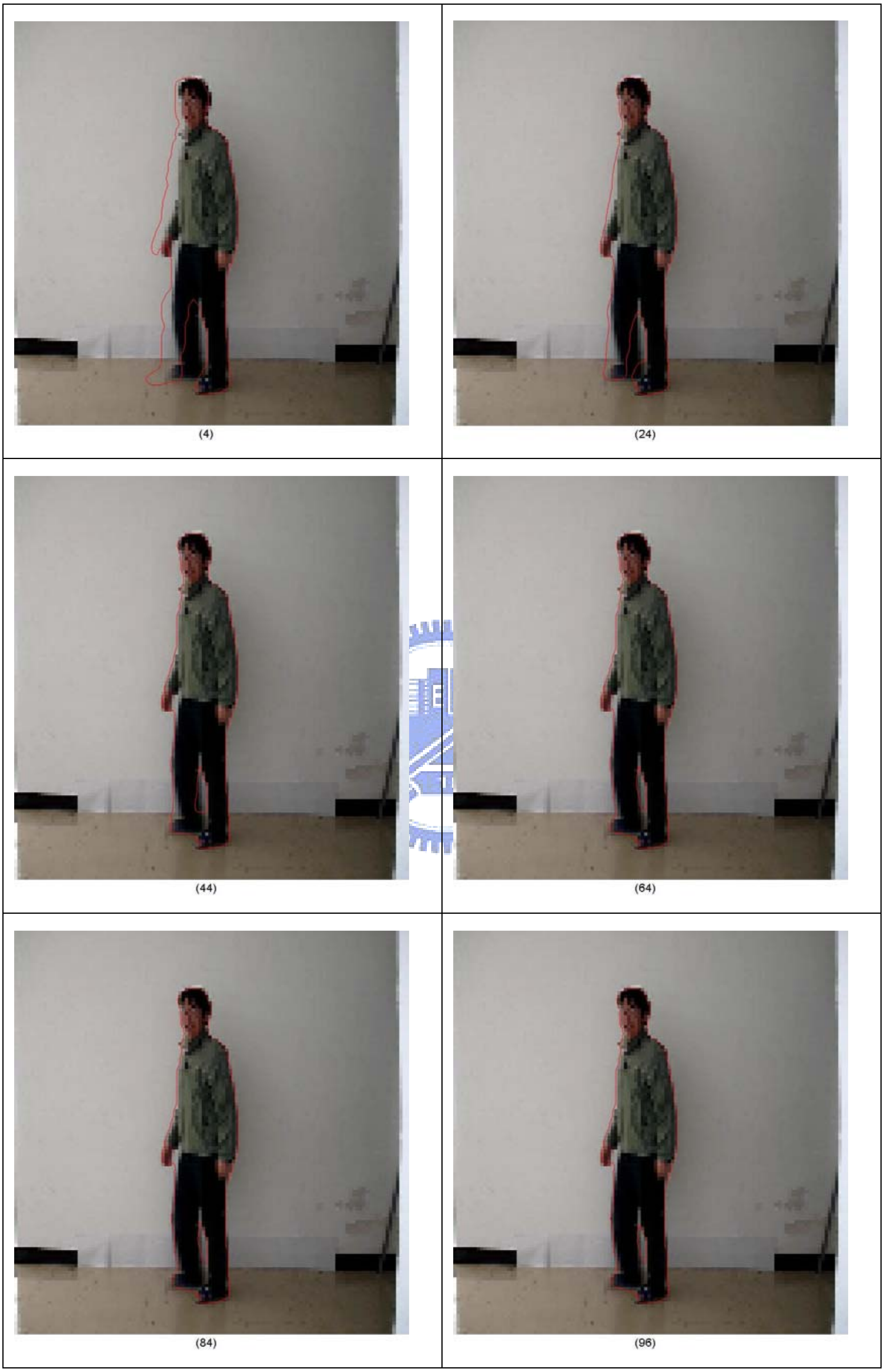
(44)

(64)

(84)

(96)

Fig 6.35 λ =0.05

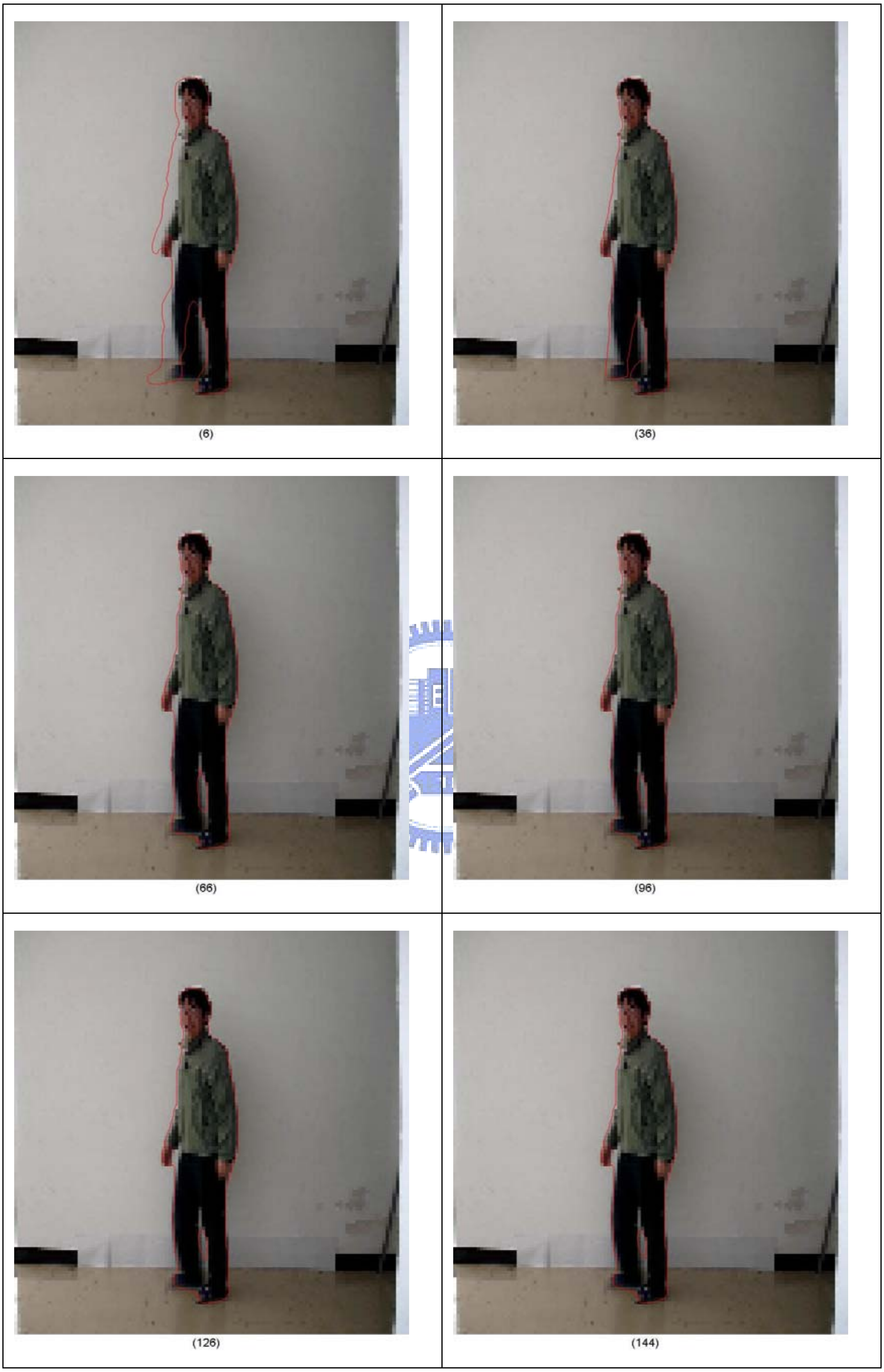


(4) (24)

(44) (64)

(84) (96)

Fig 6.36 λ =0.07



(6)

(36)

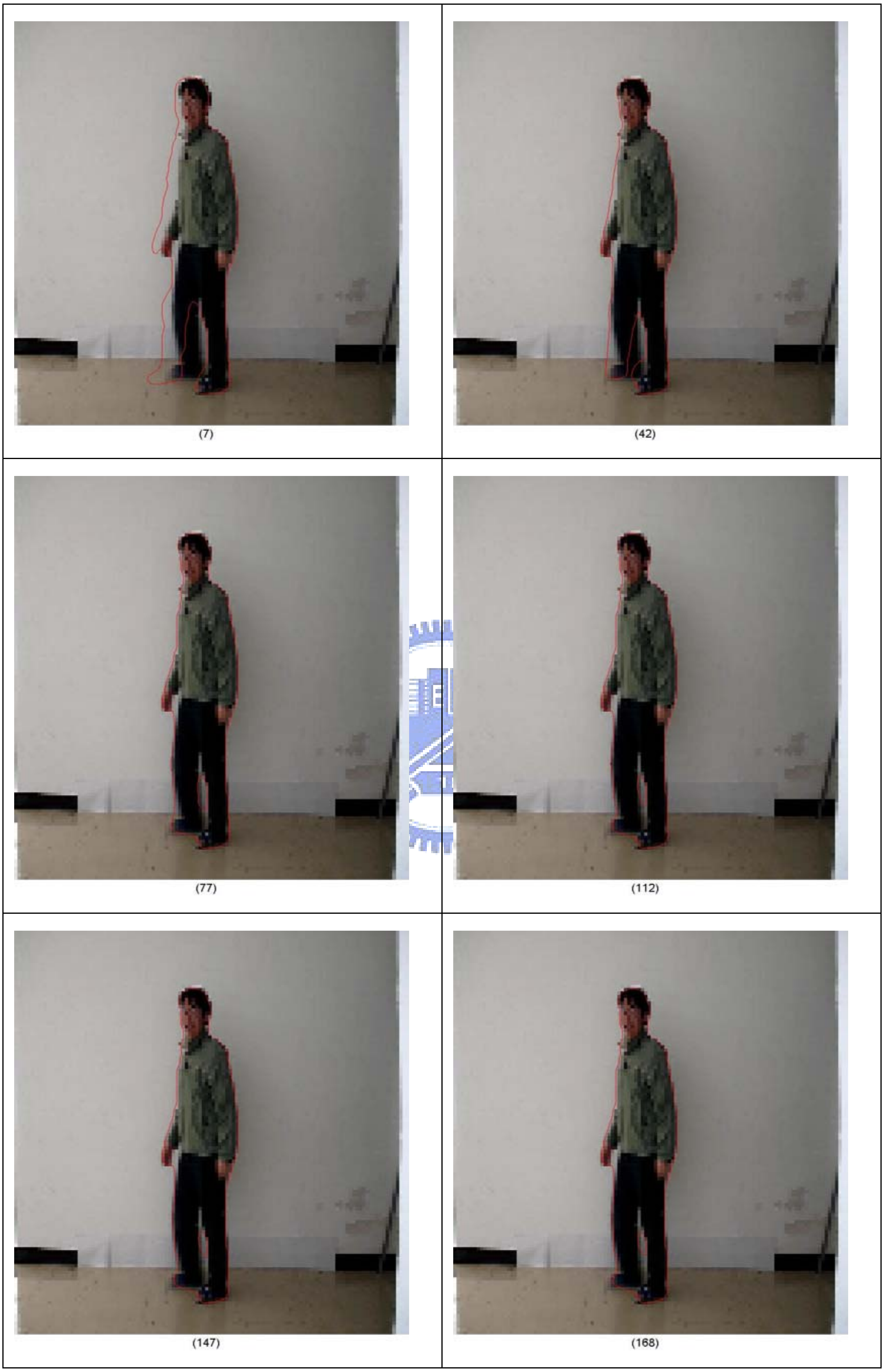
(66)

(96)

(126)

(144)

Fig 6.37 λ =0.09



(7)

(42)

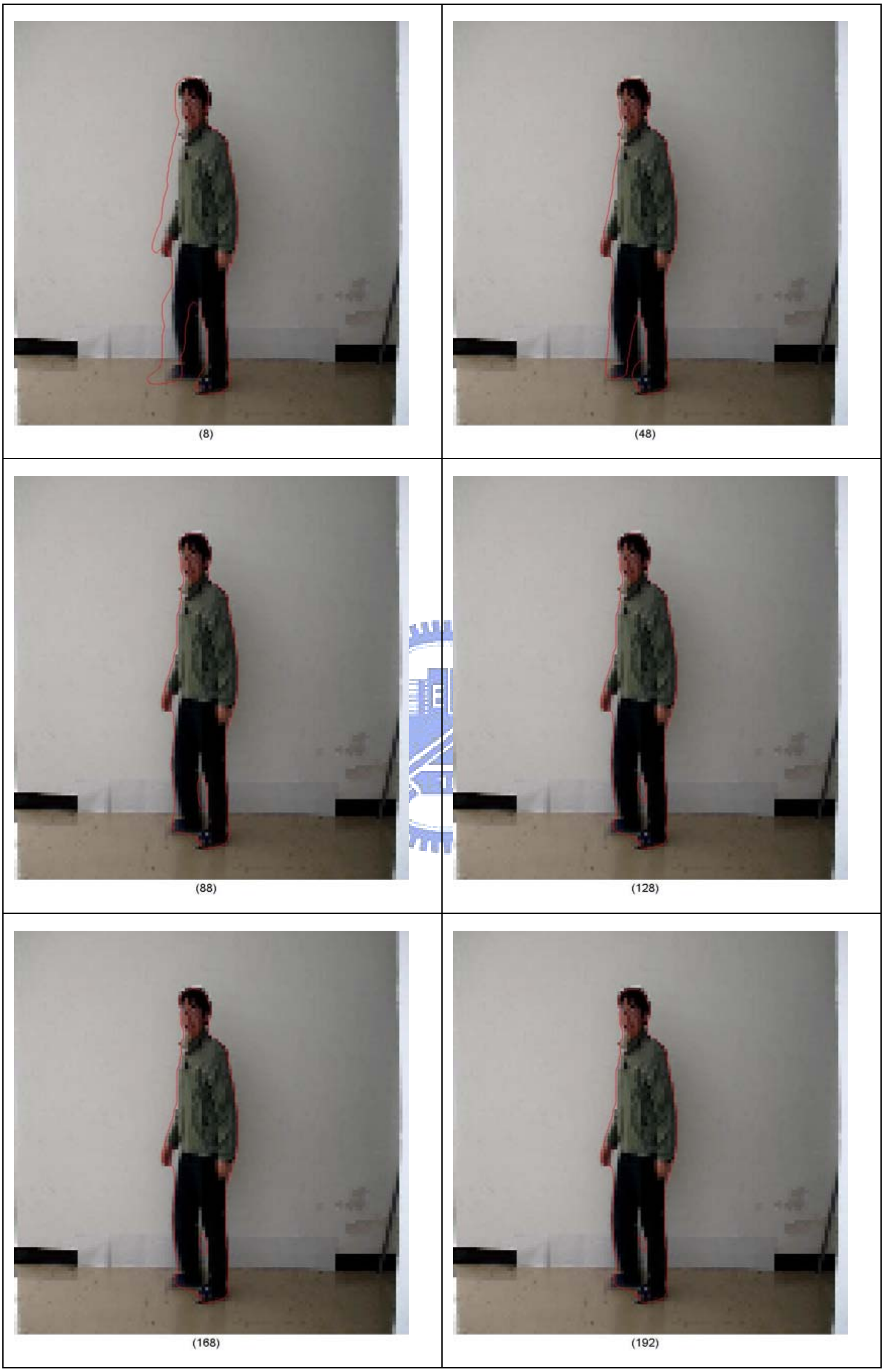
(77)

(112)

(147)

(168)

Fig 6.38 λ =0.11



(8)

(48)

(88)

(128)

(168)

(192)

Fig 6.39 λ =0.13

7. Conclusions and Future Work

7.1. Conclusions

We present a tracking method for object contour tracking in image sequence. The presented approach is combined Abdol-Reza's model with a shape prior. The shape prior is based on geodesic active contour model and will align with the contour during propagation process. When the object being tracked canno be partially discriminated, the shape prior.

7.2. Future Work

1. The shape prior can use a statistical training data set and be implemented as multiple shape priors.
2. The stop condition can be checked automatically by energy criterion instead of a specified number of iterations.
3. We believe a region based discriminate analysis is expected to perform better than pixel-wise based algorithms.

References

[1] C. Xu and J. L. Prince, ``Snakes, Shapes, and Gradient Vector Flow," *IEEE Transactions on Image Processing*, 7(3), pp. 359-369, March 1998.

[2] Vicent Caselles, Ron kimmel and Guillermo Sapiro, “Geodesic Active Contours” , International Journal of Computer Vision 22(1), 61－79 (1997).

[3] Malladi, R., Sethian, J.A., and Vemuri,”Shape modeling with front propagation: A level set approach,” *IEEE Trans. On PAMI*, 17:158–175, 1995.

[4] M. Bertalmio, G. Sapiro, and G. Randall, “Morphing Active Contours: A Geometric Approach to Topology-Independent Image Segmentation and Tracking,” Proc. Int'l Conf. Image Processing, vol. III, pp. 318-322, 1998.

[5] M. Bertalmio, G. Sapiro, and G. Randall, “Morphing active contours,” *PAMI*, vol. 22, no. 7, pp. 733–737, 2000.

[6] J. A. Sethian and Berkeley,” Level Set Methods and Fast Marching Methods (2nd Edition)”, Cambridge University Press, 1999.

[7] Stanley J. Osher and Ronald P. Fedkiw, “Level Set Methods and Dynamic Implicit Surfaces”, Springer.

[8] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer, “Kernel-Based Object Tracking,” *IEEE Trans. On PAMI*, VOL. 25, NO. 5, MAY 2003

[9] Tony F. Chan and Luminita A. Vese, “Active Contours Without Edges”, IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 10, NO. 2, FEBRUARY 2001.

[10] Kaleem Siddiqi, Yves B’ erub’ e Lauzi`ere, Allen Tannenbaum, and Steven W. Zucker, “Area and length minimizing flows for shape segmentation”, IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 7, NO. 3, MARCH 1998

[11] Guillermo Sapiro and Allen Tannenbaum,” Area and length preserving geometric invariant scale-spaces ",IEEE Transactions on PAMI, Vol. 17, No. 1, Jan 1995

[12] Luminita A. Vese and Tony F. Chan,”A multiphase level set framework for image segmentation using the Mumford and Shah model",UCLA CAM Report 01-25, September 2001

[13] T. F. Cootes, C. J. Taylor, D. H. Cooper, et al., " Active Shape Models - Their Training and Application," Computer Vision and Image Understanding, 61(1): 38-59, January 1995.

[14] Nikolaos Paragios and Rachid Deriche,”A PDE-based Level-Set Approach for Detection and Tracking of Moving Objects” Sixth International Conference on Computer Vision (ICCV'98) January 04 - 07, 1998 Bombay, India. p. 1139.

[15] Nikos Paragios and Rachid Deriche,“ Geodesic Active Contours and LevelSets for Detection and Tracking of Moving Objects ", IEEE Transactions on PAMI, VOL. 22, NO. 3, MARCH 2000

[16] Nikos Paragios and Rachid Deriche, “Geodesic Active Regions and Level Set Methods for Supervised Texture Segmentation,” *IJCV*, vol. 46, no. 3, pp. 223–247, 2002.

[17] A.R. Mansouri, “Region tracking via level set PDEs without motion computation,” PAMI, vol. 24, no. 7, pp. 947–961, 2002.

[18] Alper Yilmaz, Xin Li, Mubarak Shah,“ Contour-Based Object Tracking with

Occlusion Handling in Video Acquired Using Mobile Cameras," IEEE TPAMI, Volume 26, Number 11, pp. 1531-1536, November 2004.

[19] G. Tsechpenakis, K. Rapantzikos, N. Tsapatsoulis and S. Kollias, "A Snake Model for Object Tracking in Natural Sequences," Elsevier, Signal Processing: Image Communication, Volume 19, Issue 3, March 2004, pp. 219-238.

[20] Daniel Cremers, Florian Tischhäuser, Joachim Weickert* and christoph schnörr, "Diffusion Snakes: Introducing Statistical Shape Knowledge into the Mumford-Shah Functional", in International Journal of Computer Vision 50(3), 295–313, 2002.

[21] X.M. Pardo *, V. Leborán, R. Dosil," Integrating prior shape models into level-set approaches," Pattern Recognition Letters, vol. 25, pp. 631–639, 2004.

[22] Dainiel Cremers and Stefano Soatto, "A Pseudo-distance for Shape Priors in Level Set Segmetation," in N. Paragios (Ed.), 2nd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision, Nice, Oct 2003.

[23] Michael E. Leventon , W. Eric L. Grimson and Olivier Faugeras ,"Statistical shape influence in geodesic active contours," In Proc. Conf. IEEE CVPR, volume 1, pages 316–323, Hilton Head Island, South Carolina, June 13–15, 2000.

[24] David E. Breen, and Ross T. Whitaker,"A Level-Set Approach for the Metamorphosis of Solid Models" , IEEE Transactions on Visualization and Computer Graphics, Vol. 7, No. 2, pp. 173-192, April-June 2001.

[25] K. Museth, DE Breen, RT Whitaker and AH Barr, "Level Set Surface Editing Operators, " ACM Transactions on Graphics(Proc. SIGGRAPH), Vol. 21, No. 3, July 2002, pp. 330-338.

[26] Stanley Osher and Ronald P. Fedkiw,” Level Set Methods: An Overview and Some Recent Results”, IPAM GBM Tutorials March 27 － April 6, 2001