Moving object boundary extraction using level set method

Myo Thida

2008

Myo, T. (2008). Moving object boundary extraction using level set method. Master' s thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/13117>

<https://doi.org/10.32657/10356/13117>

Moving Object Boundary Extraction using Level Set Method

Myo Thida

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Master of Engineering

2008

*To my parents and siblings,
for their encouragement and love.*

Acknowledgements

First and foremost, I would like to express my deepest appreciation and sincere gratitude to my supervisor, Associate Professor Kap Luk Chan for giving me confidence and support to embark on my Master's programme. This work would not be possible without his valuable advices, guidance and encouragement during the course of this research.

I would like to express my deep and sincere thanks to my co-supervisor, Dr. How Lung Eng from the Institute for Infocomm Research, Singapore. His useful advices, patient discussions and moral support have been a great value for me. I would also like to acknowledge the Institute for Infocomm Research and the Center for Signal Processing from the School of Electrical and Electronic Engineering, Nanyang Technological University for providing me with financial support and excellent research facilities.

It is my pleasure to extend my warmest thanks to my seniors, Ms. Wen Fang and Mr. Jiayin Zhou for their grateful help and valuable suggestions. Special thanks go to Ms. Yan Gao for her understanding, and sharing my stress through the hard time.

Last but not least, I wish to thank my parents and siblings, to whom this thesis is dedicated, for years of unconditional love and support.

Table of Contents

Acknowledgement	i
Table of Contents	ii
Summary	vii
List of Figures	ix
List of Tables	xiii
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	3
1.3 Major Contribution of the Thesis	4
1.4 Organization of the Thesis	5
2 Literature Review	7
2.1 Curve Evolution Theory	8

2.1.1	Examples of Curve Evolution	10
2.2	Active Contour Models: Parameterized Curve Evolution	12
2.2.1	Limitations of Active Contour Models	15
2.3	Level Set Method: Implicit Curve Evolution	15
2.3.1	Implicit Representation	16
2.3.2	Speed Functions	20
2.3.2.1	Edge Driven Speed Functions	21
2.3.2.2	Region Driven Speed Functions	23
2.3.3	Numerical Implementation of Level Set Method	25
2.3.3.1	Narrow Band Method	25
2.3.3.2	Fast Marching Method	26
2.3.3.3	Fast Level Set Implementation without Solving Partial Differential Equation	27
2.3.3.4	Discussion	30
2.4	Concluding Remarks	30
3	An Improved Moving Object Boundary Extraction Using Fast Level Set Implementation Method	32
3.1	Introduction	32
3.2	A Brief Review on Background Subtraction	33
3.3	An Improved Algorithm for Moving Objects Boundaries Extraction .	35

3.3.1	Background Subtraction Based Energy Functional	36
3.3.2	Evolution Speed Functions	37
3.3.2.1	Data Dependent Speed Function	39
3.3.2.2	Smoothness Speed Function	40
3.3.3	Coping with Low Contrast Region	40
3.3.3.1	Defining Temporal Speed Function	41
3.3.3.2	Incorporating Temporal Information Into Region Driven Energy Functional	42
3.3.4	Numerical Implementation	43
3.4	Experimental Results and Discussion	44
3.4.1	Coping with New Person Entering Into the Scene	45
3.4.1.1	Computation Time	46
3.4.2	Detection and Tracking Multiple Objects	47
3.4.3	Results in Presence of Cluttered Environment	48
3.4.4	Handling Low Contrast Objects	49
3.4.5	Comparison to Prior Work	50
3.5	Concluding Remarks	52
4	A Shape Driven Fast Level Set Method for Moving Object Boundary Extraction	53
4.1	Introduction	53

4.2	A Literature Review on Shape Prior Based Level Set Methods	54
4.3	Proposed Method	56
4.3.1	Shape Representation and Alignment for Pre-collected Shapes	58
4.3.1.1	Shape Representation	58
4.3.1.2	Shape Alignment	59
4.3.2	Shape Model Construction	60
4.3.2.1	Finding Nearest Neighbors	61
4.3.2.2	Computing Weights and Constructing Shape Prior .	62
4.3.3	Combining Shape and Image Region Information	63
4.4	Experimental Results	65
4.4.1	Comparing Proposed Method with PCA based Linear Shape Model	65
4.4.2	Evaluation of Reconstructed Shape Prior Model	66
4.4.3	Detection and Tracking a Walking Person	68
4.4.4	Computation Efficiency	73
4.5	Concluding Remark	73
5	Conclusions and Future Work	76
5.1	Conclusion	76
5.2	Future Work	77
	Author's Publications	78

Bibliography	79
A Spline Representation in Active Contour Models	85
B Adaptive Gaussian Mixture Model for Background Modeling	87

Summary

This thesis deals with the problem of automatic extraction of moving object boundary from video data. Since the level set method plays an important role in object boundary extraction, this thesis starts with a review of this method. Various models based on the level set method are presented and their advantages and disadvantages are discussed.

For a level set based method to be useful in automatic boundary extraction, two major considerations must be taken into account; 1) the formulation of an energy functional that provides desired features and 2) its high computational cost. To address these issues, the research work in this thesis focus on developing a fast level set based object boundary extraction algorithm. A region based energy functional based on background subtraction and Chan-Vese model is proposed. This energy functional has the advantage of having robustness to handle noisy and cluttered environment compared to the gradient based energy functional. A way to incorporate this energy functional into a fast level set implementation scheme is also introduced. With this scheme, level set based curve evolution can be performed in a faster way compared to the existing PDE solver based methods.

A new method to construct a shape model is proposed. The shape model is constructed based on local nearest neighbors. An energy functional based on the

constructed shape model is developed and incorporated into a region based energy functional. This extension allows recovering the shape of partially occluded objects. Shape prior modeling based on principal component analysis is also implemented for comparisons with the proposed method. Experimental results demonstrate the superior performance of the proposed method in various environmental conditions.

List of Figures

2.1	(a) General form of a curve evolution (b) Motion of a curve by speed function F in normal direction	9
2.2	Curve evolution under its curvature	10
2.3	(a) A simple closed curve (b) Not a simple closed curve	11
2.4	Curve evolution under speed function of image gradient [9]	12
2.5	Parameterized view of a curve	14
2.6	Level Set Formulation of curve Evolution: The evolution of curve (left most) and corresponding surface $\phi(x, y, t)$ at time $(t = 0, t = 1, t = 2)$ are shown [10]	17
2.7	An example of embedding a curve C as zero level set of signed distance function where curve C is shown in white: darker pixel values represent the points inside the curve while lighter pixel values represent the points outside the curve.	18
2.8	From right to left, curve at zero level set (shown in blue) splits into two separate curves while level set function remains as a single function	18

2.9	A narrow band with current estimated zero level set at center: Grid pints in gray area are members of narrow band	26
2.10	Curve Expanding in Mono-directional: Initial curve is shown in red color	27
2.11	Implicit representation of the evolution of curve C in fast level set method [5]: the value of ϕ at point A changes from positive to negative when the curve moves outward in (b)	28
3.1	Detection and Tracking without re-initialization: New person entering into the scene is ignored	45
3.2	Detection and Tracking New Person Entering:	46
3.3	Curve Evolution Time against Frame Number (image size is 352×240)	47
3.4	Tracking of multiple objects at the same time: walking in outdoor environment.	48
3.5	Tracking of walking person in the presence of cluttered background .	48
3.6	Tracking of a walking person using only image based energy term. . .	49
3.7	Tracking of a walking person with temporal constraints: the distorted shape is recovered by integrating temporal constraints	50
3.8	Proposed method versus the Prior work: Top row: experimental results obtained by the proposed method. Bottom row: results obtained by PDE-based method	51

4.1	(a) Detection result using the image-based method(Chapter 3) (b)corresponding level set function and its binary representation (c) 2 nearest neighbors of current detected shape (d) constructed shape model and (e)final detection result	57
4.2	A few examples of pre-collected shape samples	57
4.3	(a) original training samples (6 out of 22 samples) (b)corresponding binary representation	59
4.4	training samples (6 out of 22 samples) (a) reference image (b) before and (c) after alignment	61
4.5	Few instances of aligned shapes of a walking person: Walking sequence contains a large deformation of shapes	65
4.6	(a) test shape (b) shape prior model using linear PCA (c) shape model by the proposed method using 3 nearest neighbors	66
4.7	Few instances from data set contained three different types of objects: Each type contains 20 instances of shapes	67
4.8	Shape Model Reconstruction: First Row shows the test shapes with missing information and Second Row shows the reconstructed shape model using the proposed method with 3 nearest neighbors	67
4.9	(a) Unseen Test Shape (b-e) Corresponding Reconstructed Shape Models using different number of k	68
4.10	Results obtained by Purely Image Driven Term	69
4.11	Results obtained by Shape driven and Image Driven Term: Human Shapes is correctly extracted compared to Figure 4.10	70

4.12	Samples of Pre-collected Shapes	70
4.13	The first two rows show the results obtained by image driven term and the last two row show the final results obtained by the proposed method	71
4.14	Tracking Results in the presence of prominent occlusion. The first two rows show the boundary of person is accurately extracted despite the fact that the person is partially occluded. The distorted shapes in third row cannot be recovered as no reliable shape model can be constructed when more than 50 percent of shape is distorted. But the last row shows that the proposed method can extract the shape of a person once the person reappears.	72
4.15	Comparisons of the computational time with and without shape prior. (Image size is 352×240). The initialization at image border is done at the first frame of video sequence. The result in current frame is used to initialize in the proceeding frame	74
A.1	B-spline basis functions $B_i(s)$ are weighted by coefficients x_i (top) and combined linearly to form a spline function $x(s)$ (bottom). Note that the spline function follows the control polygon closely.	86

List of Tables

3.1	CPU time (in Seconds) for Images shown in Figure 3.8. Image size is	
	576×720	52

Chapter 1

Introduction

1.1 Motivation

Automated extraction of visual characteristics of an object from a scene is an important task for many computer vision problems such as in human activity analysis. Among other information such as color, texture and motion, shape characteristics leads to better understanding of an object of interest. For instance, human shape plays a key role in interpretation of human activities [1]. These shapes cannot be sufficiently described by models based on simple geometric shapes (e.g, circle, ellipse) but need to be represented by more complex models. This is why *boundary extraction* methods are popular in recent decades. These methods facilitate more precise description of shapes of interested objects. Active contour models (snakes) [2] and level set methods [3] are two well known deformable models for object boundary extraction. The idea of these methods is to evolve an initial guess of a curve or surface towards the boundaries of objects of interest.

While many *boundary extraction methods* have been developed, there are still re-

1.1 Motivation

maintaining challenges hindering to their widespread adoption in real applications. These challenges include the presence of noisy, cluttered environment, occluded objects and complex nature of deformable objects such as the human shapes. In this research, *deformable models* for extracting dynamic object boundary were studied and improved the methods to overcome the above challenges.

A popular one among the existing boundary extraction approaches was the level set method [4] which employs implicit representation of object boundary. This representation allows topological changes and it has been shown as a recent state-of-art improvement over other parameterized representation used in active contour models (snakes) [2].

Two major problems of the level set method must be considered for it to be an autonomous and robust boundary extraction algorithm. The first major problem is the formulation of an energy functional that drives the curve evolution. Ideally, minimum of the defined energy functional coincides with boundaries of interest. This is to ensure that the initial level set curve will deform to object boundaries while evolving towards the energy minimum. However, the question of ***how to define an energy functional with the desired properties*** still remains as an open question. In early works, energy functionals were defined based on image gradient. More recently, region based information of object has been considered. In all these cases, some underlying assumptions of interested objects are made. Considering the conditions at hand, one should carefully define an energy functional that can provide robustness and precision to cope with missing or misleading information from the image data.

Another problem is ***its high computational cost*** to evolve the level set function towards object boundaries. This is because the level set method requires the solving

1.2 Objectives

of partial differential equation (PDE) of a two dimensional level set function. It has been a hindrance for the deployment of the level set method in real time applications.

1.2 Objectives

The objective of this work is to explore *moving object boundary extraction methods that address both problems* afore-mentioned in the previous section. Motivated by Shi et. al,'s work [5], high computational complexity is reduced by using their fast level set implementation scheme. The algorithms are implemented based on the pixel based switching mechanism of the boundary points corresponding to the zero level set. However, the first question posted in the previous section was not being addressed in [5]. They made an assumption that the probability density function for object of interest is known beforehand. In order to achieve robust real time performance, some issues have to be considered. One of them is the possibility of automatic initialization and re-initialization of the evolving curve, both are at the beginning of the object tracking and in case of new objects entering into the scene, in the context of video surveillance applications. Another important issue is its robustness to cope with low contrast region, cluttered environment and occlusion.

All of the above-mentioned issues motivate to work on the improvement of the fast level set method. Inspired by work of [5, 6], this thesis is devoted on improving the level set method to achieve robust boundary extraction methods that, i) allow working in real-time applications running on a PC equipped with standard video hardware and, ii) achieve robustness when encountering partial occlusion.

1.3 Major Contribution of the Thesis

1.3 Major Contribution of the Thesis

The major contributions of the research work reported in this thesis can be summarized as follows:

- An improved region-based level set method for automated object boundary detection is developed. Information from background subtraction, boundary smoothness constraints and temporal consistency term are integrated within a level set framework. Region information from the background subtraction drives the initial guess of curve towards the object boundary whereas smoothness constraints ensures the curve continuity. The proposed method brings three major advantages. First, this method allows to track the object boundary with weak gradient in cluttered environment. Second, the proposed method is significantly less sensitive to the curve initialization. Third, the level set based curve evolution can be implemented using the fast level set implementation scheme [5] and is computationally more efficient than PDE based level set methods.
- The second contribution is the extension of the above method by a shape driven energy functional to handle partial occlusion. In contrast to existing shape priors in the level set framework, a new shape model is introduced using weighted local nearest neighbors. A new energy functional based on the proposed shape model is defined on the basis of a Heaviside function and combined with the image driven energy functional. Then, the shape driven level set method is implemented using the fast level set implementation scheme. To summarize, with the proposed energy functional and the fast level set implementation scheme, the proposed method provides faster and efficient results in detecting and tracking walking person.

1.4 Organization of the Thesis

1.4 Organization of the Thesis

The following chapters of this thesis are structured as follows:

Chapter 2 presents a literature review of existing boundary extraction methods. The focus of this thesis is on boundary extraction using the level set method. A review of various energy functionals used in the level set method are first presented. Then, the usefulness of this method over active contour method is highlighted and the corresponding fundamental issues of the level set method are discussed in more details.

The main contributions highlighted in the previous section are contained in chapters 3 and 4.

In Chapter 3, the improved method for moving object boundary extraction is presented. First, the background subtraction algorithms are briefly discussed. Then, the steps for constructing energy functional and implementation technique are described in details. Finally, experimental results demonstrating performance of the proposed method in various environments are presented.

Chapter 4 presents extension of the method described in Chapter 3. Specifically, an extension is proposed for dealing with *partial occlusion*. The issues of using low level information alone for energy functional is discussed and the robustness gained by incorporating high level information such as shape prior knowledge are also highlighted. Then, the steps for constructing the proposed shape model based on the local nearest neighbors are explained in details. Towards the end of Chapter 4, the robustness of the proposed method against various environmental conditions are evaluated. In addition, a comparison to other state-of-art algorithms which use the principal component analysis for shape prior model is also presented.

1.4 *Organization of the Thesis*

Finally, conclusions and possible extensions of this work are discussed in Chapter 5.

Chapter 2

Literature Review

This chapter gives a literature review on object boundary extraction methods. The active contour method and the level set method are two well known methods in this area. These methods provide object's shape in a continuous form, as opposed to disconnected edges resulting from filtering method such as canny edge detector. Filtering methods are not explored here as previous literatures have already proven that the active contour method and the level set method have better performance in extracting shapes of object of interest, which is the main interest of this research.

The other major advantage of these methods is the ability of including various types of constraints in its energy functional formulation. For instance, sizes of objects, smoothness, functional continuity and even a shape prior model can be incorporated into the functional for improving the detection. In addition, the strength of these prior assumptions can be controlled by varying the corresponding weighting parameters.

After early successes of these methods in a few applications, research in boundary extraction is now focusing on deeper and more challenging problems. The most interesting and challenging problem is to utilize these methods for human boundary

2.1 Curve Evolution Theory

extraction. The first challenging issue in this direction is to handle topological changes that occur during human movement. As a result, researchers show more interest in the level set method which brings both numerical stabilities and ease of handling of topological changes.

In this chapter, the curve evolution theory, which is the fundamental concept of above two methods, is first reviewed. The active contour method and its limitation are addressed in section 2.2. Finally, a literature review on the level set method are given in section 2.3. This section starts with the basic formulation of the level set method, and then reviews the various existing energy functionals and numerical implementation techniques. Advantages and disadvantages of the level set method are also highlighted in this section. Finally, concluding remarks are given in section 2.4.

2.1 Curve Evolution Theory

This section reviews the curve evolution theory which is a basic concept in active contour models and the level set method.

Curve evolution theory considers curves to be deformed over time. Given an initial curve C_0 , curve evolves according to the following partial differential equation:

$$\frac{\partial C}{\partial t} = F_t \vec{T} + F_n \vec{N}, \quad (2.1)$$

where $\vec{T}$ and $\vec{N}$ are tangential and inward unit normal vectors respectively as shown in Figure 2.1(a). In general, this equation means that the curve is deformed with speed function F_t in the tangential direction and F_n in normal direction. Basically, the

2.1 Curve Evolution Theory

tangential flow in equation (2.1) does not influence the geometry of curve deformation.

Hence, it can be further simplified as:

$$\frac{\partial C}{\partial t} = F \vec{N}, \quad \text{where } F = F_n. \quad (2.2)$$

This speed function F may be dependent on factors such as local curvature, normal

Figure 2.1: (a) General form of a curve evolution (b) Motion of a curve by speed function F in normal direction

to the curve, and passive advection velocity. Much of the challenges in curve evolution problems come from constructing an adequate model for speed function F . For computer vision applications such as object detection and tracking, the goal of curve evolution is to stop the curve at the boundaries of interested objects. The necessary condition for the solution of these problems is speed function F to be zero at object boundaries. The various models of speed function F have been proposed in literature depending on the applications in hand. By constructing an appropriate speed function that encapsulates all desired features, a curve evolution method can achieve robustness against image noise, boundary gaps as well as cluttered background and

2.1 Curve Evolution Theory

partial occlusions.

2.1.1 Examples of Curve Evolution

Some examples of curve evolution driven by different speed functions are given in this section. The curve evolution driven by its curvature is first discussed. Curve movement driven by curvature speed function $F(\kappa)$ plays an important role in many applications. For instance, it can be used to smooth and reshape object boundaries.

Figure 2.2: Curve evolution under its curvature

Figure 2.2 shows the evolution of a square with speed function of $F = -\kappa$. As shown in the Figure 2.2, the sharp corner of square evolves to a smooth curve and gradually shrinks to a circle. It is easier to see that the evolution of a circle driven by its curvature will collapse to a single point. More examples of curve evolution under curvature speed function can be found in [7, 8].

In the second example, curve evolution with a speed function of the form $\pm 1 - \epsilon\kappa$,

2.1 Curve Evolution Theory

Figure 2.3: (a) A simple closed curve (b) Not a simple closed curve

where ϵ is a constant, is discussed. The first constant term acts as an advection term and the second term smoothes out the curve and it has same effects as discussed earlier. This type of speed function is popularly used in image segmentation in which the goal is to stop the curve evolution at object boundary. For example, given an initial curve enclosing the object of interest, one can evolve the curve towards the object boundary by multiplying the above speed function by the term:

$$g_I(x, y) = \frac{1}{1 + |\nabla(G_\sigma * I(x, y))|}, \quad (2.3)$$

where the expression $G_\sigma * I(x, y)$ denotes the gaussian filtered image. The term $|\nabla(G_\sigma * I(x, y))|$ is essentially close to zero in uniform region whereas the value becomes very large at the region where image gradients change rapidly. Hence, the function $g_I(x, y)$ is close to unity away from boundaries, and close to zero only at significant changes in image gradient. Figure 2.4 illustrates the curve evolution driven by such a speed function. The boundaries of dimes are detected by evolving an initial curve enclosing all objects.

2.2 Active Contour Models: Parameterized Curve Evolution

Figure 2.4: Curve evolution under speed function of image gradient [9]

2.2 Active Contour Models: Parameterized Curve Evolution

This section reviews *active contour models* [2] and introduces the mathematical concepts required to derive the curve evolution equation. Then, the parametric curve representation used in active contour models, in particular, spline based curve representation adopted in [2] and its limitations are discussed.

Following the original pioneering work by Kass et al. [2], extensive research has been done on *snakes* (*active contours models*) for boundary detection. The idea is to initialize a parametric curve in an image and evolves the curve until it converges to the boundary of a searched object. The curve deformation is formulated as an energy minimization problem and the calculus of variations is used to derive the partial differential equations which leads to the solution of a minimization problem. In general, its energy functional is defined as

$$E(C) = \int_0^1 E_{int}(C_s) + E_{ext}(C_s) ds. \quad (2.4)$$

Here, $C(s)$ denotes an explicit parametric closed curve. E_{int} refers to internal energy of the curve and it imposes smoothness constraints. E_{ext} attracts the deformable contour to interesting features, such as object boundaries, in an image. The internal

2.2 Active Contour Models: Parameterized Curve Evolution

energy is generally defined as:

$$E_{int} = (v_1|C_s|^2 + v_2|C_{ss}|^2)/2, \quad (2.5)$$

where C_s and C_{ss} denote the first and second derivatives with respect to the parametric curve. The first derivative gives the rate of change of length of the curve, which means the longitudinal contraction of the curve. The coefficient v_1 allows the curve to have smaller or larger degree of contraction and therefore makes the snake act like an elastic string. The second derivative gives the rate of convexity or curvature. The coefficient v_2 regulates the rate of the change of the curve in the direction normal to its boundary. This term makes the curve like a rigid string. If the value of v_2 is high, the curve is hard and resists bending, while small values of v_2 allow the curve to develop a corner. By adjusting these two coefficients v_1 and v_2 , the curve gets an appropriate elasticity and is able to embrace the object of interest.

The external image energy terms depend on factors such as image structure, and constraints that user imposes. For instance, an energy functional of negative image gradient ($-\nabla I$) can be imposed as external energy. The evolution of the initial curve towards the minimum of this energy functional will drive the initial curve towards the high gradient parts of image (that is object boundary). It will be shown later that this model may be improved in many ways to be able to detect the boundary of a variety of objects. A necessary condition for a minimum of above energy functional (2.4) is that its first derivation must be zero and hence:

$$v_1 C_{ss} + v_2 C_{ssss} - \nabla E_{image}(C) = 0. \quad (2.6)$$

2.2 Active Contour Models: Parameterized Curve Evolution

Then, the curve evolution equation is:

$$\frac{\partial C}{\partial t} = v_1 C_{ss} + v_2 C_{ssss} - \nabla E_{image}(C). \quad (2.7)$$

The first two terms impose smoothness and rigidity while the last term drives the initial curve toward the areas of high gradient (object boundary).

To highlight the fundamental difference between the active contour models and the level set method, the parametric curve representation is described here. As shown in Figure 2.5, active contour models represents a curve in a parametric form:

$$C(s) = (x(s), y(s)), \quad 0 \leq s \leq 1, \quad (2.8)$$

where coordinates $x(s)$ and $y(s)$ are spline functions of curve parameter s . A detailed explanation of the spline representation is explained in Appendix A.

Figure 2.5: Parameterized view of a curve

2.3 Level Set Method: Implicit Curve Evolution

2.2.1 Limitations of Active Contour Models

Active contour models have been applied successfully in a wide range of applications; but, they have a few major drawbacks. First, these models rely on other mechanisms to place an initial curve somewhere near the desired contour. This becomes a major issue when there is no proper initialization algorithm exists. In addition, in situations where the desired object boundaries changes greatly in size and shape, it is difficult to prevent clustering of control points and numerical instabilities can arise. The other limitation of active contour models is that it has difficulty to handle topological changes such as splitting and merging during evolution. This difficulty is caused by the fact that re-parameterization of the curve is required whenever the topology changes occur. This increases the algorithm's complexity and computational cost.

These drawbacks of active contour models constitute motivations for exploring the level set framework as discussed in next section.

2.3 Level Set Method: Implicit Curve Evolution

The level set method is a theoretical and numerical framework for computing and analyzing the motion of a curve C in two or three dimensions. It has become a more and more popular tool for boundary detection since it was first introduced by Osher and Sethian [3, 4]. The idea of the level set method is to embed the initial position of an evolving curve as the zero level set of a higher dimensional function $\phi(x, y, t)$:

$$C(x, y, t) = \{(x, y) | \phi(x, y, t) = 0\}. \quad (2.9)$$

2.3 Level Set Method: Implicit Curve Evolution

Thus, the position of the curve C is extracted by locating the set for which ϕ vanishes. This implicit representation has numerous advantages over parametric representation, primarily because topological changes such as curve splitting and merging are well handled without explicit computation. This allows the level set method to track highly complex objects such as humans. In addition, the implicit nature of the contour representation provides numerical stability in tracking large shape variations as this method is parameter free in contrast to active contour models. However, some drawbacks exist such as heavy computation for solving partial differential equation and dependency on initialization.

Similar to active contour models, many features can be considered simultaneously such as edges, region statistics, shape and any kind of multidimensional data for construction speed function. Depending on how the propagation speed is defined, the level set method has ability of tracking a curve with sharp corners, cusps, topological changes and highly complicated shapes. Incorporating a priori knowledge of shape in the level set framework for handling partially occluded objects is recently popular and it is one of the major contributions of this thesis.

This section gives a literature review on the state-of-art of the level set method. It begins with basic concepts of the level set method and discusses a few examples of existing speed functionals. Finally, it is followed by an explanation of various numerical schemes used for implementing the level set method.

2.3.1 Implicit Representation

Instead of representing a curve C parametrically, the level set method introduces an extra dimension to the problem. For example, given a curve $C(x, y)$ bounding a region

2.3 Level Set Method: Implicit Curve Evolution

Figure 2.6: Level Set Formulation of curve Evolution: The evolution of curve (left most) and corresponding surface $\phi(x, y, t)$ at time $(t = 0, t = 1, t = 2)$ are shown [10]

Ω , the level set method defines a smooth function $\phi(x, y, t)$ that represents the curve as the level set where $\phi(x, y, t) = 0$. Rather than moving the curve directly, the level set method evolves the function ϕ as shown in Figure 2.6. This figure illustrates the surface evolution with time t . The position of the curve C at any time t is captured by finding the zero level set of the evolution function ϕ .

In general, the initial value of the level set function ϕ is defined based on signed distance d from a grid location (x, y) to the initial contour C . A positive sign is chosen if the grid point (x, y) lies outside the curve and a negative if it is located inside. Mathematically,

$$\phi(x, y, t) = \begin{cases} -d(x, y, C), & (x, y) \in \Omega; \\ +d(x, y, C), & (x, y) \in \bar{\Omega}; \\ 0, & (x, y) \in \partial\Omega = C(t). \end{cases} \quad (2.10)$$

2.3 Level Set Method: Implicit Curve Evolution

where $d(x, y)$ is minimum distance from curve C to the grid location (x, y) . Ω represents the internal region whereas $\bar{\Omega}$ represents the region outside the initial contour C . The representation of moving contour as a zero level set of implicit function ϕ

Figure 2.7: An example of embedding a curve C as zero level set of signed distance function where curve C is shown in white: darker pixel values represent the points inside the curve while lighter pixel values represent the points outside the curve.

Figure 2.8: From right to left, curve at zero level set (shown in blue) splits into two separate curves while level set function remains as a single function

2.3 Level Set Method: Implicit Curve Evolution

brings several major advantages.

- ◇ The most important advantage is handling topological changes occurring during the evolution such as curve merging and splitting. Figure 2.8 illustrates how the level set method handles topology changes naturally. The initial curve shown in blue color, embedded at the zero level set of function ϕ , will deform and split into two curves, (Figure 2.8), without changing topology of the embedding level set.
- ◇ The parameter free representation provides natural and accurate ways of extracting object boundary with more complex geometry, such as objects with sharp corners and cusps.
- ◇ Several numerical advantages such as the easy computation of normal and curvatures are also due to the level set representation. Later it will be seen that these computations will play a major role in curve propagation.

In order to evolve the level set function ϕ with time and match the zero level set of ϕ with the evolving curve, it must be always true that:

$$\phi(x(t), y(t), t) = 0. \quad (2.11)$$

Then, by computing the time derivative of above equation using the chain rule derives:

$$\frac{\partial \phi}{\partial t} + \nabla \phi(x(t), y(t), t) \cdot \frac{d\mathbf{x}}{dt} = 0, \quad (2.12)$$

where $\mathbf{x} = (x(t), y(t))$ and ∇ is a gradient operator.

By defining F as the speed function in the normal direction of the surface, the

2.3 Level Set Method: Implicit Curve Evolution

speed function F can be written as:

$$F = \pm \frac{d\mathbf{x}}{dt} \cdot \mathbf{n}, \quad (2.13)$$

where $\mathbf{n} = \nabla\phi/|\nabla\phi|$ is a local unit normal vector.

This yields an evolution equation for ϕ :

$$\frac{\partial\phi}{\partial t} = -\frac{d\mathbf{x}}{dt} \cdot \nabla\phi = \pm F|\nabla\phi|, \quad \text{given } \phi(x, y, t = 0). \quad (2.14)$$

The above equation is called initial partial differential equation since it describes the time evolution of a function on the basis of an initial condition. With the different speed functions F , a curve can move differently. In the work presented here, the level set method is used to detect object boundary through propagation of a contour with speed function that is defined based on the image properties. In the variational approach, speed function F is derived by minimizing an energy functional. The energy functional is defined to capture all criteria associated with a condition that:

$$C_{\text{true-boundary}} = \arg \min_C \{E(C)\}, \quad (2.15)$$

where the energy functional E involves local properties as well as global properties of image. The various types of existing speed functions will be discussed in details in next section.

2.3.2 Speed Functions

As mentioned in the last section, the behavior of the curve propagation process depends on the definition of the speed function F . Since the focus of this work is the

2.3 Level Set Method: Implicit Curve Evolution

application of the level set method for boundary extraction, some examples of speed functions that are specific to this application will be presented here.

2.3.2.1 Edge Driven Speed Functions

An algorithm for boundary extraction using the level set method was first introduced by Caselles [11] and it is well known as Geodesic Active Contours (GAC). The strategy is to reformulate the classical snakes model [2] into a problem of geodesic curve computation in a Riemannian space. A geodesic is a local minimum distance path between two points and finding the minimum distance in Riemannian space is equivalent to the solution of particular snakes model given as:

$$E(C) = \int_0^1 g(|\nabla I(C(q))|) |C'(q)| dq, \quad (2.16)$$

where g is a positive decreasing function such that $g(r) \rightarrow 0$ as $r \rightarrow \infty$. $C'(q)$ is the first derivation of the parametric curve C .

Compared to the Euclidean length of a curve given by:

$$L = \int_0^1 |C'(q)| dq. \quad (2.17)$$

the above energy term in equation (2.16) gives the weighted length of Euclidean distance by $g|\nabla I|$. The essence is that GAC finds a geodesic curve through a function g that best takes account of the desired image features (edges).

This energy functional is minimized by solving the Euler Lagrange equation. Details can be found in [11]. The curve evolution equation toward the minima of energy

2.3 Level Set Method: Implicit Curve Evolution

functional in (2.16) is given by:

$$\frac{\partial C}{\partial t} = g(\nabla I)\kappa\vec{N} - (\nabla g(|\nabla I|) \cdot \vec{N})\vec{N}. \quad (2.18)$$

Representing the curve C in the level set function ϕ , this equation becomes:

$$\frac{\partial \phi}{\partial t} = g(\nabla I)\kappa|\nabla \phi| + \nabla g(|\nabla I|) \cdot \nabla \phi, \quad (2.19)$$

where the curvature κ is computed on the level set function ϕ . This corresponds to speed function:

$$F = g(\nabla I)\kappa + \nabla g(|\nabla I|) \cdot \vec{N}. \quad (2.20)$$

The first term is the curvature flow multiplied by image dependent function g . This term will stop curve evolution at object boundary. The second term adjusts the position of the contour when close to real object boundary.

Numerous extensions [12, 13, 14] of this speed function for moving object boundary detection have been proposed. In [12], boundaries of moving objects are extracted in a two step approach. First, an initial curve evolves towards moving areas of image by minimizing an energy functional that depends on temporal change detection. Second, speed function based on the image gradient drives the curve further until it coincides with exact boundary of moving objects. The idea in [14] is to incorporate shape information into the above energy functional. Since energy functional depends on both information of the image gradient and shape of interest, the boundary of the object can be extracted either by larger magnitude of image gradient or by prior knowledge of its shape. However, it should be noted that edge driven speed functions do not provide satisfactory results in the presence of noise and cluttered background. Another limitation of these speed functions is the initial contour must be either enclosing the

2.3 Level Set Method: Implicit Curve Evolution

object or to be completely inside the object of interest.

2.3.2.2 Region Driven Speed Functions

One of the early region based speed functions is proposed by Chan and Vese [15] for *image segmentation*. They have presented a region based model to extract boundaries of static objects by using weak formulation of the Mumford-Shah functional [16]. A gray-level image u_0 is separated by minimizing an energy functional defined as:

$$\begin{aligned} E(\phi, c_1, c_2) = & \mu \int_{\Omega} \delta(\phi) |\nabla \phi| + \nu \int_{\Omega} H(\phi) dx dy + \lambda_1 \int_{\Omega} |u_0 - c_1|^2 H(\phi) dx dy \\ & + \lambda_2 \int_{\Omega} |u_0 - c_2|^2 (1 - H(\phi)) dx dy, \end{aligned} \quad (2.21)$$

where c_1 and c_2 are mean values of each region (inside and outside of interested objects). Ω is the image domain and $H(\phi)$ and $\delta(\phi)$ are Heaviside step function and one dimensional Dirac measure defined as:

$$H(\phi) = \begin{cases} 1, & \text{if } \phi \leq 0; \\ 0, & \text{if } \phi > 0; \end{cases} \quad (2.22)$$

$$\delta(\phi) = \frac{d}{d\phi} H(\phi). \quad (2.23)$$

As in edge driven speed functions, the resulting evolution equation toward the minimum of an energy functional in (2.21) is:

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \left[\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - v - \lambda_1 (u_0 - c_1)^2 + \lambda_2 (u_0 - c_2)^2 \right] \text{ in } \Omega \quad (2.24)$$

$$\frac{\delta(\phi)}{|\nabla \phi|} \frac{\partial \phi}{\partial n} = 0 \text{ on } \partial \Omega. \quad (2.25)$$

2.3 Level Set Method: Implicit Curve Evolution

For more details on derivation of energy functional and definition of each term can be found in [15]. Another popular model of region-based speed function in *image segmentation* is the Geodesic Active Regions introduced by [17]. The main assumption is that the probability density functions for both objects and background are available. From image region properties, an energy functional is defined as:

$$\begin{aligned}
 E(\partial R) = & (1 - \alpha) \int_0^1 g(p_C(\mathbf{I}(\partial R(p)))) |\partial \dot{R}(p)| dp \\
 & + \alpha \int \int_{R_A} \log[p_A(\mathbf{I}(x, y))] dx dy \\
 & + \alpha \int \int_{R_B} \log[p_B(\mathbf{I}(x, y))] dx dy,
 \end{aligned} \tag{2.26}$$

where p_C is the probability of a pixel (x, y) belonging to the contour C whereas p_A and p_B represent the probabilities of a pixel (x, y) belonging to object and background. R_A and R_B correspond to object and background regions respectively. The first term is edge driven and adopted from GAC. The last two terms aim at finding the optimal region by minimizing the conditional probabilities for each region R_A and R_B . Then, the speed function that evolves the curve towards the minimum of above energy functional is given as:

$$F = -\alpha \cdot \log \left(\frac{p_B(\mathbf{I}(x, y))}{p_A(\mathbf{I}(x, y))} \right) + (1 - \alpha) (g\kappa + \nabla g \cdot \vec{N}). \tag{2.27}$$

Compared to edge driven speed function in equation (2.20), this speed function has one extra term that tries to expand or shrink the curve in the direction driven by probability density functions. For instance, in a case of (x, y) is a point belongs to R_B ,

$$p_B(\mathbf{I}(x, y)) > p_A(\mathbf{I}(x, y)) \Rightarrow -\alpha \cdot \log \left[\frac{p_B(\mathbf{I}(x, y))}{p_A(\mathbf{I}(x, y))} \right] < 0. \tag{2.28}$$

2.3 Level Set Method: Implicit Curve Evolution

This negative speed term shrinks the curve and otherwise acts to expand it. As a result, region driven speed functions relax requirements of good initialization compared to edge driven speed functions. Extensions to the above region based speed functions have been proposed to solve the problem of moving object boundary extraction [5, 18, 19].

2.3.3 Numerical Implementation of Level Set Method

As mentioned earlier the level set method represents each image as a $2D$ matrix of grid points where each grid point holds a value of level set function. In a direct implementation approach, all grid points over the entire domain are updated corresponding to the motion of contour by solving its level set based partial differential equation. This produces new level set values at all grid points over the domain and changes the position of the zero level set. The advantage of a direct implementation approach is that its operations are easier to understand and it is easy to develop the program. However, this approach is computationally inefficient in many applications due to its high computational cost in updating all grid points over the entire image. To speed up the level set method, a few approaches are being considered.

2.3.3.1 Narrow Band Method

The *Narrow Band Method* is the most popular scheme for implementing the level set method. It was introduced by [20] and extensively analyzed in [21]. The essence is to reduce updating domain of level set function into a narrow band around the zero level set. Figure 2.9 illustrates the placement of a narrow band with the zero level set at the center. During curve evolution, only those values of level set function within

2.3 Level Set Method: Implicit Curve Evolution

the narrow band are updated according to the speed function. This method results in a significant cost reduction. But, the challenge is the contour position changes dramatically and one must ensure that zero level set lies within a user defined narrow band. One way to overcome this problem is to re-build a new band around the current estimation of zero level set during evolution. This rebuilding process is called as *re-initialization* and challenging part is to decide how frequent a new band should be re-built. One mechanism is to trigger a re-initialization process when the zero level set is close to borders of the narrow band.

Figure 2.9: A narrow band with current estimated zero level set at center: Grid points in gray area are members of narrow band

2.3.3.2 Fast Marching Method

Fast Marching Method, proposed by [22], is meant for moving curves with a constantly positive (or negative) speed function. This constant speed function leads to a mono-directional moving curve, i.e. a curve is either expanding or shrinking with time as shown in Figure 2.10. Under this limitation, fast marching method provides a very fast implementation scheme but it has a limited set of applications. The main drawback is that this scheme cannot be used to implement curvature dependent speed function, which is frequently encountered in many applications such as the ones which

2.3 Level Set Method: Implicit Curve Evolution

are considered in this thesis. Due to its limited applications, the fast marching method is not discussed in details here. Interested reader are referred to [4, 22]. In brief, fast

Figure 2.10: Curve Expanding in Mono-directional: Initial curve is shown in red color

marching method constructs *an arrival time function* $T(x, y)$, the time that the curve reaches a grid point (x, y) , using the following equation:

$$|\nabla T|F(x, y) = 1, \quad \Gamma = \{(x, y) | T(x, y) = 0\}, \quad (2.29)$$

where Γ is the starting position of curve and $|\nabla T|$ is the gradient of T . The above equation is known as Eikonal equation and it states the simple fact that speed is inversely proportional to the magnitude of gradient of function T . The fast marching method evolves the curve from smaller values of T to larger ones.

2.3.3.3 Fast Level Set Implementation without Solving Partial Differential Equation

A fast level set implementation method is proposed in [5]. In contrast to previous approaches, this scheme implements the level set based curve evolution without solving the partial differential equation. The essence is to evolve a curve by observing discontinuity of level set function. By embedding the current position of curve as zero level set of the implicit function, the curve movement occurs at discontinuity of

2.3 Level Set Method: Implicit Curve Evolution

function ϕ (sign changes of ϕ) as shown in Figure 2.11. For instance, the curve at

Figure 2.11: Implicit representation of the evolution of curve C in fast level set method [5]: the value of ϕ at point A changes from positive to negative when the curve moves outward in (b)

grid point A moves outward when the value of ϕ at point A changes from positive to negative. On the other hand, the curve moves inward at grid point B as its ϕ value changes from negative to positive. Based on the above observation, the fast level set implementation method evolves a curve by switching grid points within two lists of neighbor points along the current position of the curve. A detailed description of fast level set implementation is as follows:

- **Initialization Procedure** Two lists of neighboring points, L_{in} and L_{out} are specified based on the signed distance function in a narrow band around the current position of curve. Their definitions are as follows:

$$L_{in} = \{\mathbf{x} | \phi(\mathbf{x}) < 0 \text{ and } \exists \mathbf{y} \in N_4(\mathbf{x}) \text{ such that } \phi(\mathbf{y}) > 0\}, \quad (2.30)$$

$$L_{out} = \{\mathbf{x} | \phi(\mathbf{x}) > 0 \text{ and } \exists \mathbf{y} \in N_4(\mathbf{x}) \text{ such that } \phi(\mathbf{y}) < 0\}, \quad (2.31)$$

where $N_4(\mathbf{x})$ is 4-connected discrete neighborhoods of a grid point $\mathbf{x}$ with coordinates of (x,y) . Those grid points inside the initial curve but not in L_{in} are

2.3 Level Set Method: Implicit Curve Evolution

defined as *interior points* and those points outside the curve but not in L_{out} are defined as *exterior points*. With these definitions, level set function is re-defined locally as follows:

$$\phi(\mathbf{x}) = \begin{cases} 3, & \text{if } \mathbf{x} \text{ is exterior point;} \\ 1, & \text{if } \mathbf{x} \text{ is in } L_{out}; \\ -1, & \text{if } \mathbf{x} \text{ is in } L_{in}; \\ -3 & \text{if } \mathbf{x} \text{ is interior point.} \end{cases} \quad (2.32)$$

- **Evolution Procedure** At each iteration, the speed function in the grid points L_{in} and L_{out} are first computed. Then, the curve evolution is performed in two steps as follows:

1. Grid points with positive speed values $F(x, y) > 0$ in L_{out} are switched to L_{in} and new ones from exterior points that satisfy equation (2.31) are added into L_{out} . Then, values of level set function of both switched points and new added ones are updated according to equation (2.32). This leads the moving curve to outward direction as explained in Figure 2.11.
2. The curve is moved inward direction by switching grid points with negative speed function $F(x, y) < 0$ in L_{in} to L_{out} . Similar to step one, the new grid points that satisfy equation (2.30) are added into L_{in} and its values of level set function are updated according to equation (2.32).

This allows the curve to move in both directions (inward and outward).

2.4 Concluding Remarks

2.3.3.4 Discussion

A literature review on various implementation schemes of the level set method has been presented in this section. The *Narrow band method* deals with only those grid points in a narrow band around the current position of the zero level set function. Level set function is updated by solving the partial differential equation in each iteration. *Narrow band method* allows the curve to move freely but still left considerable computational burden. Fast marching method moves the curve outward (or inward), that is, from small values of $|T|$ to larger ones, by solving the equation of $|\nabla T|F = 1$. This leads to a faster implementation than the narrow band method but the fast marching method is restricted to a specific model of curve evolution, i.e., curve moving is restricted in a single direction only. The *fast implementation method without solving partial differential equation* bridges this gap by allowing the curve to move in both directions (inward and outward) while significant speed-up is achieved.

2.4 Concluding Remarks

This chapter has presented a literature review on active contour models and the level set method. The level set method possesses a significant advantage for detecting and tracking highly complex object boundary. Major advantages of the level set method over active contour models are highlighted in section 2.3. However, it should be noted that the algorithms will not work as expected without considering the application context in designing speed functions. On the other hand, selecting a proper implementation scheme is a necessary condition for many applications. For instance, *fast implementation without solving partial differential equation* will be the best suited scheme for applications that require real time performance.

2.4 Concluding Remarks

In the following chapters, extensions to above speed functions discussed in section 2.3.2 will be studied and new energy functionals for moving objects boundary extraction will be developed.

Chapter 3

An Improved Moving Object Boundary Extraction Using Fast Level Set Implementation Method

3.1 Introduction

This chapter presents an *improved real time contour tracking algorithm* using fast level set implementation scheme discussed in section 2.3.3.3. This research is highly motivated by work presented in [5]. In [5], authors showed that fast level set implementation scheme can be used for real time objects tracking. In their method, the curve is evolved by speed function obtained based on prior knowledge of feature distribution of objects. By assuming that objects of interest have already been extracted in the first frame of video sequence, *fast implementation scheme* requires only a few number of iterations to achieve tight object boundaries. However, this requirement for boundary extraction in the first frame can be difficult in practice as it requires

3.2 A Brief Review on Background Subtraction

user interaction or other mechanism for object boundary extraction. In addition, their method does not incorporate any motion detection module and as a result, it does not handle new objects entering into the scene.

In this work, the moving object boundaries extraction method using fast level set implementation scheme has been developed. A *new speed function* by incorporating background subtraction into Chan-Vese model (2.21) is proposed. The proposed method removes dependency on other mechanism or user interaction. Furthermore, automatic initialization and re-initialization function based on motion detection module gives the possibility of tracking new persons entering into the scene. In short, the proposed method addresses the shortcomings in [5] with a new proposed speed function and additional features while real time performance is still maintained. Another improvement is incorporating temporal information to cope with low contrast objects.

Before describing the proposed method in details, a brief review on background subtraction methods is presented in next section. After that, formulation of the proposed region based speed function in the fast level set implementation framework is explained in section 3.3. Finally, section 3.4 gives a discussion on efficiency of the proposed method evaluated under different environments.

3.2 A Brief Review on Background Subtraction

In this framework, the output of background subtraction process is used for modeling a region based speed function. Here, a review on background subtraction methods is briefly described. Background subtraction is a basic component of motion detection for an image sequence against static background. The main assumption behind this method is that a representation of the scene background can be modeled and

3.2 A Brief Review on Background Subtraction

maintained. In the literature, a variety of background modeling techniques have been reported, including averaging [23, 24, 25], parametric [26, 27, 28] and non-parametric [29] statistical modeling. In any indoor or outdoor scene, there are changes that occur over time and may be classified as changes to the scene background. A good background reference model must tolerate these scene changes either by being invariant to them or by adapting to them. As the objective here is *to obtain an adaptive background model that could be used in real time level set based curve evolution*, the review emphasize only on those adaptive background models that can achieve real time performance.

A simple background modeling method is proposed in [23] where the first reference background is initialized with the first frame of video sequence, i.e., $B_0 = I_0$. Then, background model is updated using first order recursive filter such as:

$$B_{t+1} = (1 - \alpha)B_t + \alpha I_t, \quad (3.1)$$

where α is a learning rate coefficient and I_t is current image information. The larger the α is, the faster the changes in the scene are integrated to the background frame. Extensions of this model have been proposed in literature. Some models used different updating schemes for foreground pixels and background pixels whereas some update only those pixels belonging to background model. These models are sensitive dynamic changes such as repetitive noise, sudden illumination changes, etc.

In [26], authors use a statistical background model where each pixel is represented with its minimum (M), maximum (N) and the largest inter-frames absolute difference (D) values. These values are observed during initial training period where the scene contains no moving objects and updated periodically for those parts of the scene where the system determines as non-moving parts. A pixel in current frame is classified as

3.3 An Improved Algorithm for Moving Objects Boundaries Extraction

moving if it satisfies:

$$|M(x, y) - I_t(x, y)| > D(x, y) \quad \text{or} \quad |N(x, y) - I_t(x, y)| > D(x, y). \quad (3.2)$$

Stauffer and Grimson [27] proposed an adaptive online background mixture model that can deal with lighting changes, and slow moving objects to some extent. The idea is to model the values of a particular pixel as a mixture of Gaussian instead of modeling the values of all pixels as one particular type of distributions. This model is being used in this work for constructing an energy functional. This is selected due to their real time adaptive property. A detail explanation of this method is given in Appendix B.

3.3 An Improved Algorithm for Moving Objects Boundaries Extraction

A new region based speed function is proposed to derive level set based curve evolution. This speed function is obtained by minimizing a background subtraction based energy functional. In contrast to [5], the proposed method relaxes extraction of objects in first frame. Instead, a motion detection module is incorporated for automatic initialization and re-initialization. Setting image border as initial curve in every frame will detect all person in the scene but this will slow down the performance of algorithm. The research here is to have a real time performance of moving objects boundaries extraction. Based on these considerations, a module to detect number of moving persons in the scene is integrated. Upon reporting changes from this module, the level set curve is re-initialized at image border.

3.3 An Improved Algorithm for Moving Objects Boundaries Extraction

3.3.1 Background Subtraction Based Energy Functional

An adaptive background mixture model in [27] is used in this work. A reference background image is modeled and updated according to equations (B.3 - B.6) in Appendix B. Then, the difference image is obtained by gray-level differencing between the current frame and the reference background model. Ideally, the observed difference image is a mixture distribution of two regions. Assuming that background is static, distribution corresponding to background region will have zero mean whereas object region can be interpreted as a distribution with a larger mean and larger variance. This property is employed to define a region based energy model and curve evolution speed function is derived.

Denote the difference image by

$$I_d(x, y) = I(x, y) - B(x, y), \quad (3.3)$$

where $I(x, y)$ and $B(x, y)$ are current image and background image respectively. By exploiting the property of difference image into Chan-Vese model (2.21), an energy functional is defined as

$$\begin{aligned} E_{image}(\phi_t) &= \lambda_1 \int_{\Omega} (I_d(x, y) - c_1)^2 H(\phi(x, y, t)) dx dy \\ &+ \lambda_2 \int_{\Omega} (I_d(x, y) - c_2)^2 (1 - H(\phi(x, y, t))) dx dy \\ &+ \mu \int_{\Omega} |\nabla H(\phi(x, y, t))| dx dy. \end{aligned} \quad (3.4)$$

where $\lambda_1, \lambda > 0$ and $\mu > 0$ are fixed parameters. ϕ is a *time dependent level set function* satisfying equation (2.32).

On each frame, the initial guess of the level set function $\phi(x, y, t = 0)$ evolves

3.3 An Improved Algorithm for Moving Objects Boundaries Extraction

such that above energy functional given in equation (3.4) is minimized. Initialization of ϕ at time zero is arbitrary and denoted as ¹ $\phi(x, y, t = 0) = \phi_0(x, y)$. The two parameters c_1 and c_2 are average intensities of difference image I_d over the region for which $\phi_t < 0$ and $\phi_t > 0$ and updated with level set function ϕ . Mathematically, these parameters for current level set function $\phi_t = \phi(x, y, t)$ are defined as:

$$\begin{aligned} c_1(\phi_t) &= \frac{\int_{\Omega} I_d(x, y) H(\phi_t) dx dy}{\int_{\Omega} H(\phi_t) dx dy}, \\ c_2(\phi_t) &= \frac{\int_{\Omega} I_d(x, y) (1 - H(\phi_t)) dx dy}{\int_{\Omega} (1 - H(\phi_t)) dx dy}, \end{aligned}$$

where $H(\phi_t)$ is Heaviside function and defined as

$$H(\phi_t(x, y)) = \begin{cases} 1, & \phi_t(x, y) < 0; \\ 0, & \phi_t(x, y) > 0. \end{cases} \quad (3.5)$$

These two parameters are dynamic in nature as they are updated with ϕ . The first two terms in above functional represent the region fitting energy and will be minimized only if the curve is at the boundary of object being tracked. The last term in E_{image} gives a constraint on length of object boundary and control its smoothness.

3.3.2 Evolution Speed Functions

Object boundaries are extracted by evolving an initial guess of contour in each frame, such that the final energy given in equation (3.4) is minimized. The curve evolution equation driving the curve towards the minimum of energy functional (3.4) is given

¹Please note that , in level set method, t describes *evolution time* of level set equation in each frame and it does not relate to physical time. For consistency of notations used, here, frame number such as current frame and previous frames are denoted as I^n and I^{n-1} .

3.3 An Improved Algorithm for Moving Objects Boundaries Extraction

as:

$$\frac{\partial \phi}{\partial t} = \delta(\phi) [\mu \kappa - \lambda_1 (I_d(x, y) - c_1)^2 + \lambda_2 (I_d(x, y) - c_2)^2], \quad (3.6)$$

where κ is curvature of level set function ϕ and defined as:

$$\kappa = \operatorname{div}\left(\frac{\nabla \phi}{|\nabla \phi|}\right). \quad (3.7)$$

In order to extend the evolution to all level sets of ϕ , Dirac function $\delta(\phi)$ can be replaced with $|\nabla \phi|$. Hence, the above equation (3.6) becomes

$$\frac{\partial \phi}{\partial t} = F |\nabla(\phi)|, \quad (3.8)$$

with speed function F :

$$F = \mu \cdot \kappa - \lambda_1 (I_d(x, y) - c_1)^2 + \lambda_2 (I_d(x, y) - c_2)^2, \quad (3.9)$$

where $\mu > 0$ and $\lambda_1, \lambda_2 > 0$ are fixed parameters.

The first term in speed function minimizes the length of zero level set function and works as smoothness regularizer. The last two terms come from image data and evolve the curve toward object boundary. Based on this, speed function is divided into data dependent and smoothness speed functions as follows:

$$F_d = -\lambda_1 (I_d(x, y) - c_1)^2 + \lambda_2 (I_d(x, y) - c_2)^2, \quad (3.10)$$

$$F_s = \mu \cdot \kappa. \quad (3.11)$$

3.3 An Improved Algorithm for Moving Objects Boundaries Extraction

Hence, level set based curve evolution equation becomes:

$$\frac{\partial \phi}{\partial t} = (F_d + F_s)|\nabla \phi|, \quad (3.12)$$

where F_d drives the curve towards the desired boundary and F_s smoothes the curve.

3.3.2.1 Data Dependent Speed Function

In equation (3.10), $I_d(x, y)$ approximates to zero in areas where moving objects are absent. Thus, the parameter c_2 will be approximately zero when the curve lies outside the object and the second term in F_d will be zero. As a result, the resultant F_d will be negative if a pixel lies outside the moving object. On the other hand, the first term in F_d will be smaller than the second term and the resultant F_d will be positive if a pixel lies inside the moving object. The negative and positive values of F_d correspond to shrinking and expanding forces of the curve.

The numerical value of F_d controls the movement of curve along the normal direction. However, this actual value of F_d is not required in this application where the goal is to extract the final object boundary. In the fast level set implementation, the expanding and shrinking of curve movement occurs only when the sign of speed function changes. Based on this condition, data dependent speed function is redefined as:

$$F_d = \begin{cases} -1, & \text{if } A(x, y, c) < 0; \\ 1, & \text{if } A(x, y, c) > 0. \end{cases} \quad (3.13)$$

where $A(x, y, c) = -\lambda_1(I_d(x, y) - c_1)^2 + \lambda_2(I_d(x, y) - c_2)^2$.

This allows to evolve the level set function using switching mechanism between two lists of neighboring points which is explained in section 2.3.3.3. As explained

3.3 *An Improved Algorithm for Moving Objects Boundaries Extraction*

earlier, this implementation reduces expensive computational cost of PDE based level set implementation approaches and allows level set based contour tracking method to run in real time. The convergence performance of fast level set implementation using switching mechanism has been proven by Shi et. al. [5]. He has showed that the average distance between results of using switching mechanism and PDE based approaches are less than one pixel.

3.3.2.2 Smoothness Speed Function

According to equation (3.11), the smoothness speed function comes from the length constraints and is the evolution of curvature of level set function ϕ . The smoothness speed function is defined as in [5].

$$F_s = \text{sign}(G \otimes \phi(x, y)), \quad (3.14)$$

where G is defined as 3×3 2D Gaussian function. This smoothness process using Gaussian filtering accelerates minimization process. This is because this process cares only about the sign of the output of Gaussian filter and avoid evaluating curvature which is computationally expensive.

3.3.3 Coping with Low Contrast Region

A number of models have been proposed to deal with low contrast region or partial occlusion. A general trend among researchers is to incorporate shape prior information to constraint the curve evolution [14], [30], [31]. The prior shape is constructed by modeling a training set of level set functions using principal component analysis. Here, the interest of current research is to obtain a method that can recover the

3.3 An Improved Algorithm for Moving Objects Boundaries Extraction

deformed shape with no prior shape information. The only assumption for this proposed method is that the shapes of the object in two consecutive frames of sequence are quite similar. This is a valid assumption most of the time for tracking of objects as the shapes of the object in the image sequence normally contain strong temporal correlations.

Based on the above assumption, the previous model is extended to handle low contrast regions by incorporating temporal constraints into data dependent speed function.

3.3.3.1 Defining Temporal Speed Function

E_{temp} is speed function based on temporal constraints and is calculated by minimizing the following energy function:

$$E_{temp} = \int \int_{\Omega} (H(\phi^n(x, y)) - H(\phi^{n-1}(x, y)))^2 dx dy, \quad (3.15)$$

where ϕ^{n-1} and ϕ^n are corresponding level set functions in current frame and previous frame. H is Heaviside function as defined in equation (3.5). Thus, minimizing E_{temp} derives the evolving level set function ϕ to find similar shape with respect to the result in previous frame. To account for translation in the image, the above shape similarity measure E_{temp} is further extended as:

$$E_{temp} = \int \int_{\Omega} (H(\phi^n(x, y)) - H(\phi^{n-1}(x + T_x, y + T_y)))^2 dx dy, \quad (3.16)$$

where the translational vector $T = \begin{pmatrix} T_x \\ T_y \end{pmatrix}$ is obtained by computing distance between centroid of two consecutive frames.

3.3 An Improved Algorithm for Moving Objects Boundaries Extraction

Then, the curve evolution equation towards the minimum of temporal information driven energy functional(3.16) is given as:

$$\frac{\partial \phi}{\partial t} = F_{temp} |\nabla \phi|, \quad (3.17)$$

where speed function is given by:

$$F_{temp} = -(H(\phi^n) - H(\phi^{n-1})). \quad (3.18)$$

3.3.3.2 Incorporating Temporal Information Into Region Driven Energy Functional

Combining equation (3.17) with previous evolution equation (3.12), the final expression of surface evolution becomes:

$$\frac{\partial \phi}{\partial t} = ((1 - \tau)F_d + F_s + \tau F_{temp}) |\nabla \phi|. \quad (3.19)$$

For sake of clarity, we now define speed function that integrates data dependent and temporal information as external speed function and smoothness term as internal speed function as follow:

$$F_{ext}(x, y) = (1 - \tau(x, y))F_{d(x, y)} + \tau(x, y)F_{temp}(x, y), \quad (3.20)$$

$$F_{int}(x, y) = F_s(x, y). \quad (3.21)$$

3.3 An Improved Algorithm for Moving Objects Boundaries Extraction

where τ is a balancing factor between image information and temporal constraints.

Hence, final evolution equation becomes:

$$\frac{\partial \phi}{\partial t} = (F_{ext} + F_{int})|\nabla \phi|, \quad (3.22)$$

In equation (3.20), τ determines how much temporal information should be incorporated. In our proposed method, we define τ based on strength of data dependent speed function that determines association of each pixel to background or object. If the underlying data dependent speed function is very weak, image information is not reliable and temporal information should be incorporated. On the other hand, temporal information is not required when underlying speed function is very strong. Hence, the balancing factor between image information and temporal information is defined as:

$$\tau(x, y) = \begin{cases} 1, & |F_d(x, y)| < \rho; \\ 0, & \text{otherwise.} \end{cases} \quad (3.23)$$

where the parameter ρ is a threshold and determined based on the variance of data dependent speed function $F_d(x, y)$.

3.3.4 Numerical Implementation

Using *fast implementation scheme* [5] explained in the previous chapter, the level set based curve evolution is implemented in two cycles. During the first cycle, the curve is evolved based on external speed function F_{ext} (3.20) as follows:

- F_{ext} is computed for every point $(x, y) \in L_{in}$. The point is moved from L_{in} to L_{out} if $F_{ext}(x, y) < 0$.

3.4 Experimental Results and Discussion

- F_{ext} is computed for every point $(x, y) \in L_{out}$. The point is moved from L_{out} to L_{in} if $F_{ext}(x, y) > 0$.

The above two steps are iterated until one of the following conditions is satisfied:

$$(a) \quad \begin{aligned} F_{ext}(x, y) &\leq 0 & \forall (x, y) \in L_{out} \\ F_{ext}(x, y) &\geq 0 & \forall (x, y) \in L_{in} \end{aligned}$$

- (b) a pre-specified number of iterations, N_a is reached.

The condition in (a) satisfies that all the points in L_{in} have positive speed values (i.e., they are in the object) and all the points in L_{out} have negative speed values (i.e., they are in the background). However, in the case of noisy or cluttered environment, the condition in (a) will not be satisfied. Hence, the maximum number of iterations N_a is required to be pre-defined to ensure that the curve evolution reaches the convergence in the case of noisy data. In the second cycle, the curve evolves based on the internal speed function using similar procedures in cycle one. The number of iterations for second cycle, N_s is defined to be smaller than N_a .

3.4 Experimental Results and Discussion

In this section, a number of experimental results are chosen to highlight different properties of the proposed method. Four video sequences chosen involves both indoor and outdoor and highly cluttered environment. In all experiments, the first few frames of the video are assumed to have no moving objects and assigned to initialize a reference background model. For the subsequent frames, the difference image between the current frame and background model is used as input to the algorithm while

3.4 Experimental Results and Discussion

background model is updated with time. All experiments are performed on 2.4 GHz PC.

3.4.1 Coping with New Person Entering Into the Scene

The first experiment is performed on the *hall monitor sequence* which has been used as a base of test for many tracking algorithm using level set method. This experiment shows the ability of the proposed method to track new person entering into the scene. For comparison, results obtained without re-initialization is shown in Figure 3.1.

Figure 3.1: Detection and Tracking without re-initialization: New person entering into the scene is ignored

The new person entering into the scene is ignored (this is the case in Shi. et al. [5] approach). This is overcome in the proposed method with a re-initialization module which re-initializes the curve to be at the image border upon detecting the changes of the number of moving objects. The results of the proposed method is shown in Figure 3.2. Not only the existing person but also new person entering into the scene can be detected and tracked.

The first row shows the detection results. By initializing the curve at the image border, the moving person is detected in 243 iterations. In the second row, the subsequent tracking results are displayed. By using previous result as the initial curve in the current frame, only a few number of iterations is required to obtain tight object boundaries. The detection and tracking of new person is shown in last row. Note

3.4 Experimental Results and Discussion

Figure 3.2: Detection and Tracking New Person Entering:

that one part of level set function is expanding while the other part is shrinking. This demonstrates that the proposed method with fast level set implementation scheme allows the level set based curve to expand and shrink at the same time.

3.4.1.1 Computation Time

The real time performance of *fast level set implementation scheme* over the narrow band method is already discussed in [5]. The computational efficiency affected by the proposed re-initialization scheme is highlighted here. Figure 3.3 shows the computation time required at each frame of results shown in figure 3.2. Note that two peaks occur at frame number 20 and 84 due to our re-initialization scheme. At these two frames, the curve is re-initialized to track the new person entering into the scene (see in first and last row of figure 3.2). As a result, more number of iterations are required to reach the convergence stage and hence, high a computational cost. However, this gives the advantage of detecting and tracking of new persons and relax the requirement of closed initialization in first frame. Average tracking time of our

3.4 Experimental Results and Discussion

proposed method for *Hall monitor sequence* is 0.0651 sec per frame. This is much faster than those level set based methods using other implementation schemes which require at least 2 to 3 sec per frame. A detailed comparison of computational cost in implementing level set based curve evolution can be found in [32].

Figure 3.3: Curve Evolution Time against Frame Number (image size is 352×240)

3.4.2 Detection and Tracking Multiple Objects

In the second experiment, the robustness of our algorithm in the outdoor environment is evaluated. This is an outdoor sequence in which three persons are walking in close proximity. The results in Figure 3.4 shows that though two persons are grouping into one when they are almost in contact with each other, they are detected separately later. This example also shows the ability of level set based method which can handle the topological changes naturally.

3.4 Experimental Results and Discussion

Figure 3.4: Tracking of multiple objects at the same time: walking in outdoor environment.

3.4.3 Results in Presence of Cluttered Environment

The tracking of the objects in the presence of cluttered background is presented in Figure 3.5. Notes that the background is highly cluttered and gradient-based level

Figure 3.5: Tracking of walking person in the presence of cluttered background

3.4 Experimental Results and Discussion

set method will easily fail in this environment. The experimental results show that the proposed method is able to track the walking persons in the sequence against the cluttered background.

3.4.4 Handling Low Contrast Objects

This experiment illustrates the tracking of object entering into the low contrast region. For comparison, the moving person is first tracked without using temporal constraints. As shown in the second row of Figure 3.6, the shapes of the person become distorted when the background has similar statistical properties of the person being tracked. This is indeed a challenging task and additional constraints are required to recover the shape. Figure 3.7 shows the experimental results obtained

Figure 3.6: Tracking of a walking person using only image based energy term.

with the modified speed function (refers to equation (3.20)) that integrates data dependent and temporal information. Clearly, exploiting temporal information recovers the missing information.

3.4 Experimental Results and Discussion

Figure 3.7: Tracking of a walking person with temporal constraints: the distorted shape is recovered by integrating temporal constraints

3.4.5 Comparison to Prior Work

As discussed in section 2.3.2, well-known region-based approaches in the level set framework, such as Geodesic Active Region [17], are based on probability modeling of region property. In addition, these methods implement the curve evolution by solving PDE equation. But, such probability modeling and solving of PDE require high computational cost. In this thesis, the work has focused on achieving a method that can rapidly extract moving objects.

Experimental results in Figure 3.8 compares the performance of the proposed method with the original Chan-Vese model which is implemented by solving the PDE equation. Chan-Vese model using PDE-solver is implemented on the Matlab platform while the proposed method is implemented in Visual C++ platform. It is understandable that this may not be a very fair comparison. But, this comparison at least highlights that the proposed method provides near real time performance while the PDE solving methods do not. Figure 3.8 shows the results of the proposed method (the upper row) and the original Chan-Vese model (the lower row) implemented by solving PDE equation. The CPU time for these images are listed in Table 3.1. It is obvious that the proposed method is significantly faster than the PDE-based method.

3.4 Experimental Results and Discussion

As already reviewed in section 2.3.3, Shi et al. [5] has also proven that the fast level set implementation scheme is much faster than any PDE solving approach. Hence, in this comparison, the focus is emphasized on their final segmentation results. As can be seen in Figure 3.8, the performance of the proposed method is superior or at least comparable to the PDE solving approach. To summarize, with the proposed energy functional and fast implementation scheme, the proposed method provides more efficient results.

Figure 3.8: Proposed method versus the Prior work: Top row: experimental results obtained by the proposed method. Bottom row: results obtained by PDE-based method

3.5 Concluding Remarks

	Img 1	Img 2	Img 3
Proposed Method (using C++ platform)	0.515	0.608	0.812
PDE-based Method (using Matlab platform)	706.087	1060.877	1271.369

Table 3.1: CPU time (in Seconds) for Images shown in Figure 3.8. Image size is 576×720

3.5 Concluding Remarks

In this chapter, an improved real time contour detection and tracking algorithm combining both image features and temporal information has been represented. The image features-based energy functional is obtained by background subtraction and Chan-veese Model (2.21). The motion detection result is combined into the Chan-Vese model and the model is reformulated so that it can be used in fast level set implementation framework. The temporal information based energy functional is also incorporated for handling low contrast region. The PDE based curve evolution equation is obtained by minimizing the energy functional. Real time performance is achieved by implementing the obtained PDEs with methods described in [5].

To validate the proposed method, the method is tested with a number of real video sequences. These sequences involve both indoor and outdoor environment as well as cluttered background. Experimental results illustrate the robustness of the proposed method. Both single and multiple objects are accurately tracked despite its environment. In addition, the new person entering into the scene is also tracked with the re-initialization scheme. The experimental result also demonstrates the ability of recovering the distorted shape using temporal constraints. A comparison with prior work (PDE based approach) has also shown the favorable performance of the proposed method.

Chapter 4

A Shape Driven Fast Level Set Method for Moving Object Boundary Extraction

4.1 Introduction

An autonomous boundary extraction method using the fast level set implementation scheme is proposed in Chapter 3. The method minimizes the region based energy functional to drive the initial guess of curve toward the object boundary. This functional relaxes the requirement of having good contour initialization (refers to section 3.3) and provides better performance for *detecting low contrast object boundary in cluttered background* and performance is close to *real time*.

However, in many real applications, the object of interest may not be easily discriminated from the background by its intensity alone. Utilizing image information

4.2 A Literature Review on Shape Prior Based Level Set Methods

alone often leads to poor performance in the presence of noise, scene clutter or occlusion. High level information such as prior knowledge of shape is useful to recover the boundary of object.

This chapter presents a *shape driven* approach to moving object boundary extraction using *fast level set implementation scheme*. A new method to construct a shape model based on local nearest neighbors is proposed. In contrast to existing approaches of shape priors in the level set framework, the proposed shape model allows to handle larger shape variations. In addition, the proposed shape driven level set method can be performed in a faster way compared to existing PDE solver approaches.

Section 4.2 describes a brief literature review on incorporating shape prior constraint into the level set method. In section 4.3, the proposed method is explained in details. Experimental results on real video sequences and comparisons to linear PCA are given in section 4.4. Finally, conclusion is followed in section 4.5.

4.2 A Literature Review on Shape Prior Based Level Set Methods

The incorporation of shape prior information into PDE based level set method has been shown to be an effective way to deal with partial occlusion [14, 33, 34, 35]. The essence is to add a constraint based on a similarity measure between the current evolving shape and a *prior* one. Leventon et. al [33] proposed to incorporate a shape prior model into the geodesic active contour (an edge based level set method) to enforce the contour evolution towards the direction of a prior shape. The shape prior is modeled using principal component analysis under the assumption that shapes in

4.2 *A Literature Review on Shape Prior Based Level Set Methods*

training set are distributed according to a Gaussian function. A similar technique was used in [34] where all training shapes are represented by their level set functions as discussed in section 2.3. In [35], authors proposed a method to incorporate shape information into geodesic active region (a region based level set method). An alternative shape prior model based on spline representation is proposed by Chen et al. [14]. She proposed to use the average of a training set of aligned curves where each curve is represented using the spline curve. Then, the shape prior model is imposed on the zero level set of an implicit function ϕ .

The above methods focus more on the detection of a single shape pattern at a time and hence they cannot be applied to a set of training samples that contain several different shape patterns. Recently, Cremers [36, 37] proposed to use an energy functional which is a weighted combination of multiple shape priors. Each shape in the training data is weighted by a factor which decays exponentially based on the distance from the evolving shape. However, this method suffers from high computational burden as it considers all training samples at each step of curve evolution.

In [30], Cremers proposed to use a dynamic labeling function to select the region where a prior model needs to be enforced. This method also allows handling of multiple object segmentation. Another method to integrate non-linear shape statistics into the problem of image segmentation is proposed in [38]. In their approach, a density estimation in the feature space of nonlinearly transformed data based on kernel principal component analysis (kPCA) is performed. This method shows that the resulting shape prior has better ability than linear PCA to deal with nonlinear shape variation. Although this method can handle multiple shape patterns, it is not well suited for detecting a walking person which can have many deforming shapes.

In this chapter, a new shape model which allows the handling of larger shape

4.3 Proposed Method

variations is developed. The training shapes contain multiple different patterns of a walking person. The local nearest neighbors of current evolving shape are computed by minimizing the reconstruction error given by the local covariance matrix. Then, a shape model is constructed using the weighted nearest neighbors and the curve is evolved under the shape constraint.

In addition to handling large shape variations, the computational cost of implementation process is also considered. This is an important issue, especially, in the video surveillance application. Hence the fast level set implementation scheme discussed in section 2.3.3.3 is utilized in the curve evolution process. With this scheme, the obtained shape driven curve evolution can be performed in a faster way compared to existing PDE solver approaches.

4.3 Proposed Method

Figure 4.1 illustrates the idea of the proposed method, which comprises the image driven curve evolution, shape model construction and followed by shape driven curve evolution. The object boundary is first extracted using the image driven method proposed in Chapter 3. The binary representation of the detection result is then obtained for shape model construction. Then, the proposed shape model is constructed using $k = \{1, 2, 3, \dots, n\}$ nearest neighbors of the current detection result. In the last step, the current level set function is further evolved by incorporating constructed shape model.

In the following, the proposed method will be explained in details. Section 4.3.1 discusses on shape representation and alignment of pre-collected shapes. The construction of shape model and incorporating shape information into fast level set

4.3 Proposed Method

method are explained in section 4.3.2.

Figure 4.1: (a) Detection result using the image-based method(Chapter 3) (b)corresponding level set function and its binary representation (c) 2 nearest neighbors of current detected shape (d) constructed shape model and (e)final detection result

Figure 4.2: A few examples of pre-collected shape samples

4.3 Proposed Method

4.3.1 Shape Representation and Alignment for Pre-collected Shapes

This step does not require to learn shape variation in the training stage. The only thing required is a set of shapes of the deformable object. These shapes are referred as *pre-collected shapes* in this thesis. In this section, the shape representation as well as the shape alignment for pre-collected shapes are discussed. By pre-aligning the shapes, a reference orientation for the pre-collected shapes is obtained. The test shape is only required to align with the reference orientation and position.

4.3.1.1 Shape Representation

Most research on statistical shape modeling in the level set framework has been devoted to implicit shape representation. However, the concept of representing shapes as signed distance function has its own limitation. The space of signed distance function is not a linear space [37]. Neither a mean shape nor a linear combination of signed distance functions will in general correspond to a signed distance function. It can lead to shapes that are very different from the learned shape as explained in [39].

In this work, the binary image representation is used for pre-collected shapes. In particular, each of the N shapes in the sample set is represented by **1** inside and **0** outside of the object as shown in Figure 4.3. Note that a shape represented by level set function ϕ can be easily converted to binary image using the Heaviside function $H(\phi)$:

$$H(\phi_s) = \begin{cases} 1, & \phi < 0; \\ 0, & \phi \geq 0. \end{cases} \quad (4.1)$$

The left figure in Figure 4.1 (b) shows the level set function. The darker pixels

4.3 Proposed Method

correspond to negative distance while lighter ones corresponds to positive distance function. Zero level set is highlighted with white color for better visualization. The right one in Figure 4.1 (b) shows its binary representation using Heaviside function.

(a)

(b)

Figure 4.3: (a) original training samples (6 out of 22 samples) (b)corresponding binary representation

4.3.1.2 Shape Alignment

Before constructing a shape prior model, pre-collected shapes are aligned in order to eliminate the differences due to similarity transformation. One image is fixed as reference image I_r and the rest I_j where $j = 1, \dots, N$ are aligned to I_r using an Euclidean transformation scheme such that:

$$I_j^{aligned} = \alpha R I_j + T, \quad (4.2)$$

where $T = [T_x \ T_y]^t$ is a (x, y) translation vector, α is a scaling factor and R is a rotation matrix with respect to angle Θ . These transformation parameters are learnt

4.3 Proposed Method

as follows:

$$T = \sqrt{(x_r - x_j)^2 + (y_r - y_j)^2}, \quad (4.3)$$

$$\alpha = \frac{\int \int_{\Omega} H(\phi_r) dx dy}{\int \int_{\Omega} H(\phi_j) dx dy}, \quad (4.4)$$

$$R = \begin{pmatrix} \cos \Theta & \sin \Theta \\ -\sin \Theta & \cos \Theta \end{pmatrix}, \quad (4.5)$$

where $H(\phi)$ is a Heaviside function (4.1), Θ is a rotational angle measured between major axes of reference image I_r and I_j while (x_r, y_r) and (x_j, y_j) refer to the centers of gravity.

Figure 4.4(a) shows the reference shape while 4.4(b) illustrates a few of unaligned training shapes out of the 22 instances of a walking person. The aligned shapes are shown in Figure 4.4(c).

4.3.2 Shape Model Construction

In this work, the detection result obtained by the method in Chapter 3 is used as a test shape to construct the shape model. Thus, the test shape is first given as the level set function and its binary image is obtained by equation (4.1). Then, the current level set function is further evolved by incorporating this shape model. There are two critical steps in this proposed shape modeling. The first step finds the k nearest neighbors of a test shape. In the second step, the weights for candidate neighbors are computed by minimizing the reconstruction error and a shape model is constructed as a combination of weighted neighbors.

4.3 Proposed Method

Figure 4.4: training samples (6 out of 22 samples) (a) reference image (b) before and (c) after alignment

4.3.2.1 Finding Nearest Neighbors

A critical component of finding nearest neighbors of a test shape is to have a distance measure between shapes. A various of shape similarity measures is being used in level set based shape prior modeling [40, 35, 41, 42]. In this work, a distance measure between two shapes based on Jaccard distance [43] is defined as:

$$d_{sj} = \frac{\int \int_{\Omega} (H(\phi_s) - H(\phi_j)) \, dx dy}{\int \int_{\Omega} H(\phi_s) \, dx dy + \int \int_{\Omega} H(\phi_j) \, dx dy}. \quad (4.6)$$

where $H(\phi_s)$ refers to test shape and $H(\phi_j)$ is equivalent to binary image I_j , one of the pre-collected shapes. This measure corresponds to union of two shapes minus its intersection divided by the sum of areas. The normalized measure is used to achieve its value varies within the range $0 \leq d \leq 1$ where $d = 0$ corresponds to the best match between two shapes. By using the method discussed in section (4.3.1.2), a

4.3 Proposed Method

rigid registration is performed between the test shape and training shapes before d_{sj} is computed.

Based on the above distance metric (4.6), the pairwise distances between the test shape and all pre-collected shapes are obtained. Then, k number of nearest neighbors are selected to define a matrix U size of $n \times k$. Here, n is dimension of feature vector and obtained by rearranging the binary image I (size of $M \times N$) into a column vector.

4.3.2.2 Computing Weights and Constructing Shape Prior

The second step is to compute the weights for candidate neighbors, $W = \{w_1, w_2, \dots, w_k\}$.

These weights are computed by minimizing the reconstruction error:

$$\varepsilon(W) = \left(H(\phi_s) - \sum_{j=1}^k w_j H(\phi_j) \right)^2, \quad (4.7)$$

where the weight w_j gives the contribution of j^{th} shape to the test shape. In the following, we will denote $H(\phi_s)$ as H_s and $H(\phi_j)$ as H_j for the sake of simplicity. By introducing local covariance matrix, this error can be written as:

$$\begin{aligned} \varepsilon(W) &= \left(\sum_{j=1}^k w_j (H_s - H_j) \right)^2 \\ &= \sum_{j=1}^k \sum_{m=1}^k w_j w_m C_{jm}, \end{aligned} \quad (4.8)$$

where w_j, w_m are weights for candidate neighbors and C_{jm} is a local covariance matrix which defined as:

$$C_{jm} = (H_s - H_j)^T (H_s - H_m). \quad (4.9)$$

4.3 Proposed Method

Minimizing the above error with the constraint of $\sum_{j=1}^k w_{sj} = 1$ leads to closed form solution [44, 45] as follows:

$$w_j = \frac{\sum_{m=1}^k R_{jm}}{\sum_{p=1}^k \sum_{q=1}^k R_{pq}}, \quad (4.10)$$

where R_{jm} is inverse matrix of C_{jm} and defined as $(C_{jm} + rI)^{-1}$. Similar definition is applied to R_{pq} . Here, we introduce a small multiple of the identity matrix I before computing inverse matrix to avoid singularity.

Then, a shape prior for the test shape $H(\phi_s)$ can be constructed as a linear combination of weighted neighbors:

$$H(\phi_{prior}) = \sum_{j=1}^k w_j H(\phi_j), \quad (4.11)$$

where $H(\phi_j), j = 1, 2, \dots, k$ are the k nearest neighbors of testing shape and w_j gives the contribution of j^{th} shape to the test shape.

4.3.3 Combining Shape and Image Region Information

In Chapter 3, motion detection is incorporated into the Chan-Vese model to address the problem of moving object detection. The objective has been achieved by driving the initial guess of curve toward the minimum of the energy functional given below

$$\begin{aligned} E_{image}(\phi_t) = & \lambda_1 \int_{\Omega} (I_d(x, y) - c_1)^2 H(\phi) dx dy \\ & + \lambda_2 \int_{\Omega} (I_d(x, y) - c_2)^2 (1 - H(\phi)) dx dy \\ & + \mu \int_{\Omega} |\nabla H(\phi)|. \end{aligned} \quad (4.12)$$

4.3 Proposed Method

with respect to the level set function ϕ . Here, $I_d(x, y)$ represents the gray level intensity of background subtracted image. In the following, a shape driven energy functional is proposed and integrated into above functional (4.12). Mathematically, the total energy functional is defined as:

$$E_{total} = \beta_1 E_{image} + \beta_2 E_{shape}, \quad (4.13)$$

where β_1, β_2 balance the weights of shape and image information and the shape driven energy functional is defined as follow:

$$E_{shape}(\phi) = \int \int_{\Omega} (H(\phi_t) - H(\phi_{prior})) dx dy. \quad (4.14)$$

where $H(\phi_{prior})$ is constructed shape model. In the above equation of E_{shape} (4.14), the constructed shape model $H(\phi_{prior})$ is aligned to the local coordinate associated with the current shape $H(\phi_t)$ using the method in section (4.3.1.2).

The complete curve evolution equation leading to a minimum of the functional (4.13) is given by:

$$\frac{\partial \phi}{\partial t} = (\beta_1 F_{image} + \beta_2 F_{shape}) |\nabla \phi|, \quad (4.15)$$

where

$$F_{image} = \delta(\phi) [\mu\kappa - \lambda_1(d(x, y) - c_1)^2 + \lambda_2(d(x, y) - c_2)^2], \quad (4.16)$$

$$F_{shape} = -(H(\phi_t) - H(\phi_{prior})). \quad (4.17)$$

Compared to purely image driven evolution in equation (3.6), equation (4.15) contains an additional shape based term. This knowledge driven term drives the current shape $H(\phi_t)$ towards the shape model constructed by a linear combination of its weighted

4.4 Experimental Results

neighbors. As in section 3.3, the PDE-based curve evolution (4.15) is implemented using the *fast level set implementation scheme* in order to address the time constraints.

4.4 Experimental Results

A number of experiments have been conducted to evaluate the performance of the proposed method. Here, a few results obtained are presented to highlight the key features of the proposed shape model. In section 4.4.1, the proposed shape model is compared to the result obtained by linear PCA which is a popular approach in the area of level set based shape prior reconstruction. The evaluation of proposed method in reconstructing unseen shape is provided in section 4.4.2. Lastly, section 4.4.3 presents the performance of proposed method on the problem of contour extraction of a partially occluded walking person.

4.4.1 Comparing Proposed Method with PCA based Linear Shape Model

Figure 4.5: Few instances of aligned shapes of a walking person: Walking sequence contains a large deformation of shapes

In this section, the comparison between shape prior model obtained by linear

4.4 Experimental Results

PCA and the result of the proposed method are presented. For a fair comparison, the same set of training shapes are used for both linear PCA and the proposed method. Few instances of a walking person are shown in Figure 4.5. Figure 4.6(a) shows a previously unseen shape with a slight variation from the training set. Figure 4.6(b) presents the reconstructed shape prior using the linear PCA and the result of the proposed method is shown in Figure 4.6(c). As seen, the result obtained by the proposed method is more faithful to the test shape compared to the result obtained by linear PCA.

Figure 4.6: (a) test shape (b) shape prior model using linear PCA (c) shape model by the proposed method using 3 nearest neighbors

4.4.2 Evaluation of Reconstructed Shape Prior Model

In this experiment, the proposed method is used to re-construct an unseen shape. To assess the ability of handling complex shape distribution of the proposed method, the shapes are collected from three different types of objects as shown in Figure 4.7. It is obvious that the pre-collected shapes do not form a Gaussian distribution even for the same type of object, especially, in human shapes. Linear PCA or kernel PCA will not suffice for this data set.

The proposed method is tested on all three different types of objects. Figure 4.8, the first row shows the partially distorted test images. These test images are

4.4 Experimental Results

Figure 4.7: Few instances from data set contained three different types of objects: Each type contains 20 instances of shapes

Figure 4.8: Shape Model Reconstruction: First Row shows the test shapes with missing information and Second Row shows the reconstructed shape model using the proposed method with 3 nearest neighbors

obtained by using the algorithm explained in section 3.3 with no shape prior information. Second row shows the reconstructed shape priors using the proposed method. The results illustrate that the valid reconstructed shapes can be obtained with the proposed method when a given data set contains large shape variations. But, this leaves an open question of how many nearest neighbors to choose. If k is too small, the local variation will not reflect to the reconstructed shape. On the other hand, as

4.4 Experimental Results

Figure 4.9: (a) Unseen Test Shape (b-e) Corresponding Reconstructed Shape Models using different number of k

k is large, the algorithm sees the entire data set as local neighborhood and behaves like linear PCA. In this work, the optimal number of nearest number k is empirically set. For instance, $k = 3$ is suffice for the data set presented above as shown in Figure 4.9.

4.4.3 Detection and Tracking a Walking Person

This section presents some experimental results conducted on the problem of moving object boundary detection and tracking.

In the first experiment, the proposed model is tested on the detection of a partially occluded walking person. Firstly, the images from 20 percent of the original video sequence are manually binarized and used as pre-collected shapes. Then, a horizontal bar is introduced in a video sequence of walking person to create partial occlusion. The object boundary is then detected by evolving the initial guess of curve

4.4 Experimental Results

Figure 4.10: Results obtained by Purely Image Driven Term

by image driven speed function (3.12). Figure 4.10 shows that the purely image driven speed function is not capable to separate the actual shape of humans from the occluded bar. The results obtained by the proposed method (4.15) are shown in Figure 4.11. A comparison to results from image-driven term shown in Figure 4.10 demonstrates the performance of the proposed method. The proposed shape model allows to reconstruct the shapes with missing information.

In the second experiment, an unseen sequence is used where the walking person is different from the one from pre-collected shapes. Figure 4.12 shows some examples of pre-collected shapes. The extraction results obtained by image driven term are shown in the first two rows and final results of applying the proposed method are given in the last two rows of Figure 4.13. With the balancing factors (between image

4.4 Experimental Results

Figure 4.11: Results obtained by Shape driven and Image Driven Term: Human Shapes is correctly extracted compared to Figure 4.10

Figure 4.12: Samples of Pre-collected Shapes

driven and shape information), the proposed method provides faithful results.

While the first two experiments evaluated the proposed method in situations where

4.4 Experimental Results

(a) Results obtained by Image Driven Term

(b) Results obtained by Image Driven plus Shape Model

Figure 4.13: The first two rows show the results obtained by image driven term and the last two row show the final results obtained by the proposed method

the shapes of interest are partially occluded, in the third experiment, the performance of the proposed method is assessed on objects with prominent occlusions. The first 5 percent of video sequence is hand-segmented and used as pre-collected shapes. Figure 4.14 shows the results obtained using the proposed method with the shape

4.4 Experimental Results

Figure 4.14: Tracking Results in the presence of prominent occlusion. The first two rows show the boundary of person is accurately extracted despite the fact that the person is partially occluded. The distorted shapes in third row cannot be recovered as no reliable shape model can be constructed when more than 50 percent of shape is distorted. But the last row shows that the proposed method can extract the shape of a person once the person reappears.

driven term. With no assumption on object transformation, the reconstructing shape prior needs to rely only on the information obtained by the image driven term. As a result, no reliable shape model can be reconstructed when more than 50 percent of

4.5 Concluding Remark

shape is distorted. This leads the method to rely only on image information and as a consequence, corrupted shapes are not recovered, as shown in the third row of Figure 4.14. However, the proposed method is capable to extract the shapes, once over 50 percent of walking person reappears as shown in the last row of Figure 4.14. These results can be improved by incorporating transformation parameters or assuming constant translational motion as in section 3.3.3.

4.4.4 Computation Efficiency

In order to quantify the influence of shape information on the computational efficiency measured by curve evolution time, the comparison is done between computational time with and without shape model. To this end, we ignored the computational time required to construct a shape model in the current comparisons. Figure 4.15 gives an example of run-time behavior of the proposed method on image sequence of Figure 4.13. The payoff for using shape information is that more number of iterations are required to achieve convergence and, as a result, more computational time is required. On average, the proposed method with shape model takes about 0.1399 seconds to finish the whole process. This is an increase about 3 times compared to evolution without shape information. But, this still provides faster performance compared to traditional implementation schemes such as narrow band which requires on average 2 to 3 seconds to reach convergence.

4.5 Concluding Remark

In this chapter, a new method has been presented to incorporate the shape information into a region based level set method. In contrast to existing shape priors in the

4.5 Concluding Remark

Figure 4.15: Comparisons of the computational time with and without shape prior. (Image size is 352×240). The initialization at image border is done at the first frame of video sequence. The result in current frame is used to initialize in the proceeding frame

level set framework, a new shape model is introduced using local nearest neighbors. A new energy functional based on the proposed shape model is defined on the basis of a Heaviside function and combined with the image driven energy functional. Then, the curve evolution equation leading to the minimum of the combined energy functional is derived. This evolution equation drives the initial guess of curve towards the object boundary.

Experimental results confirm that the proposed shape model out-performs the linear shape prior when dealing with the large shape variations as in shapes of a walking person. The current work used the all pre-collected shapes for searching k nearest neighbors of the test shape. As a consequence, the computational time increases with the number of pre-collected shapes. The constructing shape model can be further improved by reducing the search space in finding k nearest neighbors. For instance, the clustering of the pre-collected N shapes can be first done off-line. A hierarchical search algorithm can be used. This will reduce search space and hence, computational cost will decrease. Future research will focus on above discussed issue and learning

4.5 Concluding Remark

shape transformation parameters over time to incorporate into the proposed shape model in order to accommodate more complex situations.

Chapter 5

Conclusions and Future Work

5.1 Conclusion

This thesis focuses on automatic extraction of moving object boundary. The various types of the level set methods are first evaluated and an improved contour extraction method is proposed based on region information and fast level set implementation scheme. The improved boundary extraction method can handle cluttered background and low contrast region. The energy functional is modified to incorporate an adaptive weight which changes according to local region contrast. Using the fast level set implementation scheme, the level set based curve evolution can be performed in close to real time. Experiments were performed which showed that the intended objective was achieved.

A new method for incorporating the shape model into the fast level set implementation scheme is proposed. Shape model is constructed based on local nearest neighbors so that the model can handle more complex shape variations. To incorporate this shape model with the fast level set implementation scheme, a new shape

5.2 Future Work

based energy functional, on the basis of Heaviside function, is proposed. The curve evolution leading to the minimum of this energy functional drives the curve towards the constructed shape model. With weighting factor defined based on strength of image region information and shape model, the influence of shape model and image information is balanced. Experimental results demonstrate the advantages and limitations of the proposed method.

5.2 Future Work

This research suggests various directions for future work. Two of possible directions are highlighted below:

- The current research focused on incorporating a shape prior based on local nearest neighbors into the fast level set implementation scheme. It will be beneficial to incorporate shape dynamic over time into the current framework as this conveys important aspects in mapping from current shape to the shape in next frame. Integrating shape and dynamics of shape variation will help to estimate a shape in severe occlusion.
- Being able to extract a sequence of shapes over time and model its dynamics is useful in many applications. One of the future directions is to learn human activities and detect abnormal events using *non-stationary shapes* from video data. The shape analysis has recently been popular [46, 47]. But, in these works, explicit shape representation is adopted. Advantages of the level set representation over the point based explicit representation have been discussed in Chapter 2. These properties will result in better performance for understanding and interpreting the human activities.

Author's Publications

1. M. Thida, K. L. Chan, and H. L. Eng, "An improved real-time contour tracking algorithm using fast level set method," *Advances in Image and Video Technology, Lecture Notes in Computer Science, LNCS*, vol. 4319, pp. 702–711, Springer Berlin / Heidelberg, December 2006.
2. Annamalai Lakshmanan, Myo Thida, Kap Luk Chan, and Jiayin Zhou, "Incorporation of active contour without edges in the fast level set framework for biomedical image segmentation," in *International Conference on Biomedical and Pharmaceutical Engineering (ICBPE'06)*, (Singapore), December 2006.

Bibliography

- [1] A. Veeraraghavan, A. R. Chowdhury, and R. Chellappa., “Role of shape and kinematics in human movement analysis,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2004.
- [2] M. Kass, A. Witkin, and D. Terzopolous, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, pp. 321–331, 1988.
- [3] S. Osher and J. A. Sethian, “Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations,” *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.
- [4] J. Sethian, *Level set methods and fast marching methods : evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge University Press, 1999.
- [5] Y. Shi and W. Karl, “Real-time tracking using level sets,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2005.
- [6] A. Blake and M. Isard, *The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.
- [7] “Level set framework for fronts propagating with curvature dependent speeds.” www.math.sfu.ca/~bwo/papers/curvature/curvature.pdf.

- [8] “Example of curve evolutions.” <http://www.mpi-inf.mpg.de/~shin/Research/CCurve/node17.html>.
- [9] “Level set methods for curve evolution and image segmentation.” http://www-static.cc.gatech.edu/~delfin/research_projects/8903-sp03/.
- [10] “Level set methods and fast marching methods.” http://math.berkeley.edu/~sethian/2006/level_set.html.
- [11] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic active contours,” in *IEEE International Conference on Computer Vision*, pp. 694–699, 1995.
- [12] N. Paragios and R. 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, pp. 266–280, March 2000.
- [13] A.-R. Mansouri, A. Olivier, and J. Konrad, “Topology-independent region tracking with level sets,” in *International Conference on Image Processing*, vol. 3, pp. 66–69, 2000.
- [14] Y. Chen, H. Tagare, S. Thiruvenkadam, F. Huang, D. Wilson, and E. Geiser., “On the incorporation of shape priors into geometric active contours,” in *IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pp. 145–152, 2001.
- [15] T. Chan and L. Vese, “Active contours without edges,” *IEEE Transactions on Image Processing*, vol. 10, pp. 266–277, February 2001.
- [16] D. Mumford and J. Shah, “Optimal approximation by piecewise smooth functions and associated variational problems,” *Communications on Pure and Applied Mathematics*, vol. 42, pp. 577–685, 1989.
- [17] N. Paragios and R. Deriche, “Geodesic active regions for texture segmentation,” Tech. Rep. 3440, INRIA, France, 1998. <http://www.inria.fr/rapports/sophia/RR-3440.html>.

- [18] A. Mansouri, “Region tracking via level set PDEs without motion computation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 947–961, July 2002.
- [19] A. Yilmaz, X. Li, and M. Shah, “Contour-based tracking with occlusion handling in video acquired using mobile cameras,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, November 2004.
- [20] D. L. Chopp, “Computing minimal surfaces via level set curvature flow,” *Journal of Computational Physics*, vol. 106(1), pp. 77–91, May 1993.
- [21] D. Adalsteinsson and J.A.Sethian, “A fast level set method for propagating interfaces,” *Journal of Computational Physics*, vol. 118(2), 1995.
- [22] J. Sethian, “A fast marching level set method for monotonically advancing fronts,” in *National Academic Science USA*, vol. 93, pp. 1591–1694, 1996.
- [23] J. Heikkila and O. Silv, “Real-time system for monitoring of cyclists and pedestrians,” in *Second IEEE Workshop on Visual Surveillance*, pp. 74–81, July 1999.
- [24] B. Lo and S. Velastin, “Automatic congestion detection system for underground platforms,” in *2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, pp. 158–161, September 2001.
- [25] M. P. R. Cucchiara, C. Grana and A. Prati, “Detecting moving objects, ghosts and shadows in video streams,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1337–1342, October 2003.
- [26] I. Haritaoglu, D. Harwood, and L. Davis, “W4: A real time system for detecting and tracking people,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 962–967, 1998.
- [27] C. Stauffer and E. Grimson, “Adaptive background mixture models for real-time tracking,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 246–252, 1999.

- [28] P. KaewTraKulPong and R. Bowden, “An improved adaptive background mixture model for real time tracking with shadow detection,” in *Second European Workshop on Advanced Video Based Surveillance Systems*, September 2001.
- [29] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, “Background and foreground modeling using nonparametric kernel density for visual surveillance,” in *IEEE*, vol. 90, pp. 1151–1163, July 2002.
- [30] D. Cremers, N. Sochen., and C. Schnorr, “Towards recognition-based variational segmentation using shape priors and dynamic labeling,” in *IEEE International Conference on Scale Space Theories in Computer Vision*, vol. 2695, pp. 338–400, 2003.
- [31] D. Cremers, “Dynamical statistical shape priors for level set based tracking,” *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 28(8), pp. 1262–1273, August 2006.
- [32] N. Paragios and R. Deriche, “A PDE-based level-set approach for detection and tracking of moving objects,” in *IEEE International Conference on Computer Vision*, pp. 1139–1145, 1998.
- [33] M. Leventon, W. Grimson, and O. Faugeras, “Statistical shape influence in geodesic active contours,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 316–323, 2000. Hilton Head Island, SC.
- [34] A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, A. Grimson, and A. Willsky., “Model-based curve evolution technique for image segmentation,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 463–468, 2001.
- [35] M. Rousson and N. Paragios., “Shape priors for level set representation,” in *European Conference on Computer Vision*, pp. 78–92, 2002.

- [36] D. Cremers, S. J. Osher, and S. Soatto, “Kernel density estimation and intrinsic alignment for knowledge-driven segmentation: Teaching level sets to walk,” in *Lecture Notes in Computer Science, LNCS*, vol. 3175, pp. 36–44, 2004.
- [37] D. Cremers, S. J. Osher, and S. Soatto, “Kernel density estimation and intrinsic alignment for shape priors in level set segmentation,” *International Journal of Computer Vision*, vol. 69(3), pp. 335–351, September 2006.
- [38] D. Cremers, T. Kohlberger, and C. Schnorr, “Shape statistics in kernel space for variational image segmentation,” *Pattern Recognition*, vol. 36(9), pp. 1929–1943, 2003.
- [39] S. Dambreville, Y. Rathi, and A. Tannenbaum, “A shape-based approach to robust image segmentation,” in *International Conference on Image Analysis and Recognition*, vol. 1, pp. 173–183, 2006.
- [40] T. Zhang and D. Freedman, “Tracking objects using density matching and shape priors,” in *IEEE International Conference on Computer Vision*, vol. 2, pp. 1056–1062, October 2003.
- [41] Y. Chen, H. Tagare, S. Thiruvenkadam, F. Huang, D. Wilson, K. Gosinath, R. Briggs, and E. Geiser., “Using shape priors in geometric active contours in a variational framework,” *International Journal of Computer Vision*, vol. 50(3), pp. 315–328, 2002.
- [42] D. Cremers and S. Soatto, “A pseudo-distance for shape priors in level set segmentation,” in *IEEE Workshop on Variational , Geometric and Level Set Methods in Computer Vision*, October 2003.
- [43] http://en.wikipedia.org/wiki/Jaccard_index.
- [44] S. Roweis and L. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, pp. 2323–2326, 2000.

- [45] D. Ridder and R. Duin, “Locally linear embedding for classification,” *Technical Report PH-2002-01, Pattern Recognition Group, Delft University of Technology, Delft, The Netherlands*, 2002.
- [46] A. Veeraraghavan, A. Roy-Chowdhury, and R. Chellappa., “Matching shape sequences in video with applications in human motion analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1896–1909, December 2005.
- [47] N. Vaswani, A. Roy-Chowdhury, and R. Chellappa., “‘shape activity’: A continuous state hmm for moving/deforming shapes with application to abnormal activity detection,” *IEEE Transactions on Image Processing*, pp. 1603–1616, October 2005.

Appendix A

Spline Representation in Active Contour Models

In spline representation, a curve is represented in a parametric form as

$$C(s) = (x(s), y(s)), \quad 0 \leq s \leq 1, \quad (\text{A.1})$$

where coordinates $x(s)$ and $y(s)$ are spline functions of curve parameter s . B-splines are often used as computationally convenient representations for spline functions. A spline function $x(s)$ is constructed as a weighted sum of N basis functions B_i where $i = 0, 1, \dots, N - 1$. Mathematically,

$$x(s) = \sum_{i=0}^{N-1} x_i B_i(s). \quad (\text{A.2})$$

These spline functions can be expressed compactly in matrix notation as:

$$\begin{aligned} x(s) &= \mathbf{B}^T(s) \mathbf{Q}^x, \\ y(s) &= \mathbf{B}^T(s) \mathbf{Q}^y, \end{aligned}$$

A. Spline Representation in Active Contour Models

where $\mathbf{B}(s) = (B_0(s), B_1(s), \dots, B_{N-1}(s))^T$ is a vector of B-spline functions. Q^x and Q^y are control vectors and are defined as:

$$Q^x = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{pmatrix}^T \quad \text{and} \quad Q^y = \begin{pmatrix} y_0 \\ y_1^y \\ \vdots \\ y_{N-1} \end{pmatrix}^T. \quad (\text{A.3})$$

In practice, the number N of control points is fixed to a value which provides sufficient contour detail.

Figure A.1: B-spline basis functions $B_i(s)$ are weighted by coefficients x_i (top) and combined linearly to form a spline function $x(s)$ (bottom). Note that the spline function follows the control polygon closely.

Appendix B

Adaptive Gaussian Mixture Model for Background Modeling

In Staufer's model [27], a recent history of each pixel is modeled by a mixture of K Gaussian distributions. At any time, t , the history of a pixel (x, y) is defined as a time series:

$$\{X_1, X_2, \dots, X_t\} = \{I(x, y, i) : 1 \leq i \leq t\}, \quad (\text{B.1})$$

where I refers to the image sequence and X_i is the intensity value (scalar for gray images and vector for color images) of the pixel (x, y) at time instance i . Then, the probability of observing a certain pixel value is

$$P(X_t) = \sum_{k=1}^K \omega_{k,t} \star \eta(X_t, \mu_{k,t}, \Sigma_{k,t}), \quad (\text{B.2})$$

where K is the number of Gaussian components and $\omega_{k,t}$ is an estimate of the weight on the k^{th} Gaussian in the mixture at time t . η is a Gaussian probability density function with two parameters: $\mu_{k,t}$ is mean value and $\Sigma_{k,t}$ is covariance matrix of k^{th} Gaussian in the mixture at time t .

An online mean approximation is used to update the parameters of distribution

B. Adaptive Gaussian Mixture Model for Background Modeling

as new information is gained from incoming frame. Every new pixel value, X_t is compared against the existing K Gaussian distributions and defined that it is matched to a distribution if its value is within σ_c standard deviations. If none of the K distributions match to the current pixel value, the least probable distribution (smallest ω) is replaced with a new distribution with the current pixel value as its mean, an initially large variance and low weight, $\omega_{k,t}$. Otherwise, mean and variance of the matched distribution are updated as follows:

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \quad (\text{B.3})$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t), \quad (\text{B.4})$$

where $\rho = \alpha\eta(X_t|\mu_k, \sigma_k)$. Here, α is a user defined parameter called learning rate which defined the learning speed of the background updating. The prior weights on K distribution belonging to the new pixel, X_t , are also updated as:

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha M_{k,t}, \quad (\text{B.5})$$

where α is a learning rate and $M_{k,t}$ is 1 for matched distribution and 0 otherwise.

As the parameters of the mixture model of each pixel changes, the K Gaussian distributions for each pixel are ordered by the factor ω/σ . Thus, the most likely background distributions are listed on top and the least probable ones gravitates towards the bottom and are eventually replaced by new distributions. Then, the first B distributions that have the most supporting evidence (large weight parameters) and small variance are defined as components of background model. Mathematically,

$$B = \arg \min_b \left[\left(\sum_{k=1}^b \omega_k \right) > T \right], \quad (\text{B.6})$$

where T is a user defined threshold that determines minimum portion of data that should be accounted as background. If a smaller T is chosen, it means that background is modeled using only the most probable distribution. This saves processing time

B. Adaptive Gaussian Mixture Model for Background Modeling

but background model may not tolerate small changes due to leaves on a tree and illumination variation. On the other hand, if T is large, background model adapts the small background changes but it could lead to detecting slow moving objects as background. It is often useful to use hypothesis in determining T according to application context.