NON-LOCAL ACTIVE CONTOURS

A Thesis
Presented to
The Academic Faculty

by

Vikram Appia

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
August 2012

NON-LOCAL ACTIVE CONTOURS

Approved by:

Dr. Anthony Yezzi, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Patricio Vela
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Allen Tannenbaum
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Christopher Barnes
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. John Oshinski
Department of Radiology
Emory University

Dr. Rajesh Narasimha
Texas Instruments

Date Approved: 9 May 2012

To my parents.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Anthony Yezzi, for his constant support and guidance. He has been a truly inspirational mentor. I appreciate the time he has taken out of his busy schedule to answer all of my questions patiently. I would also like to thank Dr. Russell Mersereau for being my Master's Thesis advisor and introducing me to the joys of research, which led me to pursue a PhD.

I would like to thank my parents for being very supportive in my endeavours. Last but not the least, I would like to thank my friends who have made my stay at Georgia Tech a wonderful and memorable experience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis deals with image segmentation problems that arise in various computer vision related fields such as medical imaging, satellite imaging, video surveillance, recognition and robotic vision. More specifically, this thesis deals with a special class of image segmentation technique called *Snakes or Active Contour Models*. In active contour models, image segmentation is posed as an energy minimization problem, where an objective energy function (based on certain image related features) is defined on the segmenting curve (contour). Typically, a gradient descent energy minimization approach is used to drive the initial contour towards a minimum for the defined energy. The drawback associated with this approach is that the contour has a tendency to get stuck at undesired local minima caused by subtle and undesired image features/edges. Thus, active contour based curve evolution approaches are very sensitive to initialization and noise.

The central theme of this thesis is to develop techniques that can make active contour models robust against certain classes of local minima by incorporating global information in energy minimization. These techniques lead to energy minimization with global considerations; we call these models — 'Non-local active contours'.

In this thesis, we consider three widely used active contour models: 1) Edge- and region-based segmentation model, 2) Prior shape knowledge based segmentation model, and 3) Motion segmentation model. We analyze the traditional techniques used for these models and establish the need for robust models that avoid local minima. We address the local minima problem for each model by adding global image considerations.

In Chapter 3, we develop a segmentation model, which constrains the evolving active

contour to be a global minimum (geodesic) with respect to an edge-based energy throughout the evolution process and minimize the region-based energy under this edge-based constraint. Since, the contour remains a geodesic through the entire ‘active’ state, we call this model the *active geodesic* model. We use a minimal path approach, based on Fast Marching technique, to obtain edge-based geodesics. We also discuss the Fast Marching Method in detail in Chapter 2.

In Chapter 4, we present the traditional level set based shape prior model and then develop a localized shape prior based segmentation model. The localized shape prior model utilizes a weighted learning algorithm to learn localized shape variations in global shapes. By clustering correlated local shape variations, we maximize the ability of our training model to learn localized shape variations. Using localized shape priors, we generate locally accurate segmentation curves that avoid a wider class of local minima when compared to existing global shape prior model. Finally, we combine these locally accurate segmentation curves to obtain a single hybrid globally accurate segmentation.

In Chapter 5, we consider active contour segmentation models that use motion of independently moving objects in any given image sequence to detect object boundaries. Traditional models strictly incorporate the optical flow constraint, which assumes that the intensity of a pixel, moving in time, remains constant. Strictly following this constraint leads to occlusion handling problems caused by regions in the background that are occluded by the moving foreground object. Traditional models impose generic constraints for motion estimation and curve evolution in these regions, which can generate erroneous pixel correspondences, causing the contour to get stuck in certain local minima. We develop a relaxed motion segmentation model that uses smooth global approximation of images to relax the strict optical flow constraint, and thereby avoiding local minima in motion-based curve evolution.

CHAPTER I

INTRODUCTION

Image segmentation has been widely researched since the advent of image processing. In a broad sense, image segmentation can be defined as the task of dividing an image into disjoint regions, which exhibit homogeneous properties or delimit object boundaries. Early approaches for image segmentation were based on detecting edges using filters, such as Canny filter [21] and Sobel filter [44]. Object boundaries were identified by classifying the pixels as edge/non-edge pixels based on a threshold. Region growing schemes, split-and-merge schemes and certain histogram-based techniques were also developed for image segmentation [46]. These approaches are heuristic in nature, and thus they need to be tailored to each specific application. Further, these approaches are ill-suited for various imaging applications with noisy images, weak/missing edges, cluttered data, and occlusions.

Curve evolution based image segmentation models were developed later. In curve evolution based approaches image segmentation is posed as an optimization problem. In these models, an image feature-based energy functional is defined on the curve. Minimizing the energy drives the curve towards near-by image features generating desired segmentation results. Such energy optimization based curve evolution models have been used extensively for image segmentation since their introduction by Kass *et al.* [51] in their seminal paper, *"Snakes: Active Contour Models"*. Further, these models also treat images as continuous rather than discrete objects, which make them independent of image discretization. These curve evolution models are also effective over a broad class of images.

The basic idea behind the active contour model developed in [51] is to minimize an energy functional of the form

$$E_{snake} = \int_0^1 \{E_{int}(C(s)) + E_{ext}(C(s))\} ds, \tag{1}$$

where $C(s) = (x(s), y(s))$ is the deformable curve with parameter $s \in [0, 1]$. E_{int} represents internal energy associated with the curve that governs the smoothness of the curve, and E_{ext} represents external energy that depends on image features or constraints added by the user. The energy in (1) can be written as

$$E(C) = \int_0^1 \{A_1 \|C''(s)\|^2 + A_2 \|C'(s)\|^2 + P(C(s))\} ds, \tag{2}$$

where C' and C'' are derivatives of the curve with respect to the parameter s, $P \geq 0$ is a cost function associated with image features, and $A_1 \geq 0$, $A_2 \geq 0$ are scalar constants. The first two terms in the integral in Equation (2) are internal forces associated with the curve, and the third term is the external image dependent force.

The active contour model in [51] uses a local gradient based edge detector to stop the evolving curve on object boundaries. A major drawback associated with this model is that the curve has a tendency to get stuck at subtle (undesired) image features/edges. These subtle features create undesired "local minima" in energy minimization. This drawback makes the model sensitive to initialization and noise. Further, an explicit parameterization scheme is used to represent the evolving curve. This explicit scheme requires re-parameterization of the grid points to avoid over-lapping of marker particles. Also, the evolving contour does not have the flexibility to undergo topological changes to segment disconnected objects. To summarize, the classical snakes model has the following three major drawbacks:

1. *Parametrization*,
2. *Difficulty in handling topological changes*,
3. *Susceptibility to local minima*.

Level set [70] based implicit curve evolution approaches were developed in [22, 53, 54, 65], to overcome two of the major drawbacks of the classical snakes model: *parametrization* and *difficulty in handling topological changes*. But, these techniques are still sensitive to local minima and initialization. So active contour models are typically initialized by the user with a contour near the desired object of interest to avoid local minima.

The local minima problem has received considerable attention within the field in the past two decades. One of the earliest approaches proposed to overcome these local minima and drive the curve towards desired image features was proposed by Cohen in [31]. Cohen proposed a simple solution to deal with the problem. He proposed including an additional balloon force to either shrink or expand the contour to drive the contour towards desired edges. But, one requires prior knowledge of whether the object is inside or outside the initial contour. Besides, the final contour will be biased towards smaller and larger segmentation while using the shrinking and expanding balloon forces, respectively. Since the introduction of active contours and the subsequent introduction of the balloon force model, several variations and modifications have been proposed to alleviate the local minima problem [11, 18, 25, 30, 31, 69, 73, 76, 81, 93, 99, 104].

In this thesis, we incorporate global image information in certain active contour models to avoid the above-mentioned undesired local minima in curve evolution. We consider the following three kinds of active contour models:

1. *Edge/Region-based segmentation model*,
2. *Prior shape knowledge based segmentation model*,
3. *Motion segmentation model*.

We discuss these models in detail and establish the necessity to address the local minima problem for each model. We overcome the issue of local minima in each of these models by using global image considerations. Hence the thesis is titled “Non-local Active Contours”.

1.1 Edge- and Region-Based Segmentation

If a monotonically decreasing, positive function of the gradient of the image ($g(\nabla I)$) is used as the cost function in (2), the minimum of the energy drives the curve towards edges in the image. As mentioned earlier, the final segmentation varies with initialization and parameterization of the curve. Caselles *et al.* [22] modify the energy functional in (2) by defining the parameter s as the arc length of the contour. This modification yields a geometric interpretation to the classical energy minimization problem in (2). Since s denotes the arc length of the curve, we have $\|C'(s)\| = 1$. Neglecting the term associated with the second derivative $C''(s)$, we modify (2) as

$$E(C) = \int_{\Omega} \{A_1 + P(C(s))\} ds = \int_{\Omega} \{\tau(C(s))\} ds, \tag{3}$$

where $\tau = P + A_1 (\Rightarrow \tau \geq 0)$. For the given length of the curve, L, the domain becomes $\Omega = [0, L]$. The internal forces are now included in the cost function τ. Thus, the energy minimization problem is reduced to finding the minimum path length of a contour in a Riemannian space.

Using the active contour model described in (3) removes the dependence on parameterization of the curve, but the final segmentation still depends on initialization. Edge-based segmentation algorithms are very sensitive to noise and spurious edges because the cost function is defined on the local image gradients. Thus, one has to initialize segmentation with a contour that lies in close proximity to the desired segmentation.

Cohen and Kimmel [30] converted the energy minimization problem in (3) into a cost accumulation problem. By treating τ as the traveling cost associated with each point on the image plane, one can find the minimum of the energy by finding the accumulated cost (u) in traveling from a given point on the image plane to every other point on the image plane. According to [30, 33], the accumulated cost u can be found by solving the Eikonal equation,

$$\|\nabla u\| = \tau. \tag{4}$$

Authors in [2, 82] developed a single-pass algorithm to find the accumulated cost u over the entire image domain, and they named the algorithm, “Fast Marching” technique. The calculated accumulated cost u, depends only on the location of a given point, and not on the geometry of the curve at that point. Thus, transforming the problem of generating open geodesics into a minimal path problem.

Cohen and Kimmel [30] used the minimal path technique to develop an interactive edge-based segmentation algorithm, which can be initialized by a single point on the object boundary, to generate closed geodesics. Later, Appleton and Talbot [11] used minimal paths to develop a segmentation algorithm, which can be initialized by a single user-given point inside the object of interest.

In contrast, region-based curve evolution techniques, developed later, define energy functionals based on region statistics rather than local image gradients. The Mumford-Shah model [67], which approximates the image with smoothly varying functions, forms the basis for various region-based segmentation algorithms. Other region-based curve evolution approaches have been developed by authors in [73, 93, 104]. Chan and Vese [25] developed a mean-curvature-flow-based level set implementation of a specific case of the Mumford-Shah energy functional, where the mean intensity of the pixels, inside and outside the curve, are used as piece-wise constant, smooth approximations for the image. In general, these region-based active contour models are less sensitive to noise and initialization in comparison to edge-based active contour models.

Therefore, while region-based segmentation algorithms are less susceptible to local minima, edge-based segmentation algorithms have a better chance of detecting edges along the object boundary. Majority of the segmentation algorithms use either edge- or region-based energy minimization. In [23, 73], authors use a linear combination of a region-based model with the classical edge-based model to exploit the benefits of both approaches. But, these schemes employ linear combination of edge- and region-based terms which may yield new classes of local minima that represent unsatisfactory *compromises* of these two criteria. In this work, we develop an *active geodesic* model that minimizes region-based energy with an in-built edge-based constraint, to exploit the benefits of both approaches.

1.2 Segmentation Based on Shape Priors

Segmentation purely based on image information becomes difficult when the image has inherent noise, low contrast, and missing/diffused edges. Such difficult scenarios are very common in medical images. In such cases, it is desirable to integrate prior shape knowledge in the active contour model, especially when we have data to train for the shape of the object of interest in the given environment.

The concept of using prior shape knowledge for image segmentation was introduced by Cootes *et al.* [34] in their seminal paper *"Active Shape Models - their training and application"*. They used an explicit parametrized curve representation framework for training, as well as curve fitting. Subsequently, research has been conducted actively in the last two decades to incorporate prior shape knowledge in image segmentation. Similar explicit parameterization based shape prior segmentation models were developed in [27,41]. Level set based implicit shape prior segmentation schemes were also developed later [37,38,61,78,79,92].

These traditional prior based segmentation models treat each training shape as a single global entity. Treating shapes as single entities reduces flexibility of the segmentation model in dealing with shapes that have severely uncorrelated shape variations in different parts (segments) of the global shape. This leads to inefficient utilization of training data, leading to undesirable segmentation results.

Instead, in our work, we introduce localized shape prior based segmentation. We cluster parts of the global shape that exhibit highly-correlated local shape variations. These local variations may be independent of the variations in the distant parts of the global shape. To cluster correlated local shape variations, we divide our image domain into various target regions prior to the training phase. We then use localized PCA [85] to train on local variations. Thus, we focus the learning from PCA to these target regions, and maximize the utility of each principal component (shape prior).

These local shape priors are used to constrain the evolving curve locally within the target regions. We can either use these local shape priors in an explicit curve representation framework, as in the case with the Active Shape models, or we can use them in an implicit level set based shape prior approach. We generate locally accurate segmentation results using these localized shape priors to constrain the evolution in each target region. Finally, we combine various locally accurate segmentation curves, using a variational framework, to generate a single accurate global segmentation.

1.3 Motion Segmentation

A variational method for optical flow estimation was introduced by Horn and Schunck [48]. Around the same time, Lucas and Kanade [64] also developed an algorithm for optical flow estimation. Later, various motion-based segmentation algorithms [13, 20, 35, 36, 49, 72, 86, 88] were developed based on the basic principles discussed in [48] and [64].

Optical flow and motion estimation algorithms assume that the brightness (gray value) of a moving pixel does not change with displacement in time. We can define this brightness constancy constraint (BCC) as

$$I_1(x, y) = I_2(x + u, y + v), \tag{5}$$

where $I_1 : \Omega \to \mathbb{R}$ and $I_2 : \Omega \to \mathbb{R}$ are images defined over a domain $\Omega \subset \mathbb{R}^2$ at two different time instants t_1 and t_2, and $\vec{d} = (u, v)$ defines the displacement vector of any given point (x, y). If we linearize Equation (5) in a differential framework, we get the famous optical-flow constraint [48] or the differential brightness constancy constraint

$$I_x \cdot u + I_y \cdot v + I_t = 0, \tag{6}$$

where I_x, I_y and I_t are partial derivatives of the image intensity and $\vec{V} = (u, v)$ is the instantaneous velocity at a given point.

Equations (5) and (6) are ill-posed because they have two unknowns. Thus, we need additional constraints to solve these equations. Horn and Schunck [48] introduced a global smoothness constraint, whereas Lucas and Kanade [64] used a local smoothness assumption to solve this problem.

The goal of motion segmentation is to identify the boundary of objects moving independently in an image sequence. Because the objects are moving independently, the assumption of smoothness in flow field is strongly violated at the object boundaries. It would be ideal if we can apply the smoothness assumption locally in the areas corresponding to the different moving objects, allowing for discontinuities in the flow field at the object boundaries. But to do so, we would have to identify the object boundaries first. Thus, motion segmentation comprises of two main tasks, motion estimation and object boundary detection.

Motion segmentation algorithms use active contour models for boundary optimization and they tend to stumble upon the local minima problem. In this work, we develop a relaxed motion segmentation algorithm, which uses global image information to model occluded regions. This makes our motion-based segmentation algorithm robust against certain local minima.

1.4 Contributions of this Thesis

The central theme of this thesis is to incorporate global information in certain traditional active contour models to make them robust against a large class of local minima. We consider different kinds of active contour models in this thesis and depending on the segmentation cue used, we propose novel ways to avoid local minima by incorporating global image considerations in energy minimization. Below, we summarize the major contributions of this thesis:

- In Chapter 2, we develop two techniques to remove the established problem of directional bias in the numerical implementation of the traditional fast marching schemes. The first approach, which we call *interpolated fast marching method*, interpolates the local traveling cost along the front to make the numerical implementation devoid of any directional bias. The second approach, *upsampled fast marching method*, computes the accumulated cost on an upsampled grid to remove the directional bias caused by overlapping influence areas of the traveling cost at adjacent grid locations.

- In Chapter 3, we present a novel *active geodesic* model, which constrains an evolving active contour to continually be a geodesic with respect to an edge-based metric throughout the evolution process. This model generates segmentation results that exhibit both local and global behaviors rather than being compromises achieved by weighted combination of region- and edge-based energies. This is achieved by reducing the search space in a region-based segmentation model with the edge constraint and making the optimization problem a finite dimension problem. It also ensures global edge optimality during the active evolution phase, which makes the model robust against unwanted local minima. Using this *active geodesic model*, we also develop an intuitive user interaction based segmentation algorithm.

- In Chapter 4, we develop a variational framework employing localized shape priors to learn and fit curves that are effective in capturing local shape variations. We segment various regions in an image separately and then combine the locally accurate segmentation curves to obtain a single globally accurate segmentation. We concentrate the efforts of learning within certain target regions by using weighted PCA based localized shape priors to enhance the utility of PCA as a tool to capture shape variations.

- In Chapter 5, we propose a novel relaxed motion segmentation model. This model treats an image sequence as overlapping foreground layer over a background layer, and estimates these layers with a smoothness constraint using available image information. These estimates generate a model for the background in the occluded regions, thereby addressing the long-standing occlusion problems that severely hamper existing motion segmentation models.

CHAPTER II

NUMERICALLY ISOTROPIC FAST MARCHING SCHEMES

Fast marching method was developed to solve the Eikonal equation (4) using a continuous (first order) model to estimate the accumulated cost at a given grid point, but uses a discontinuous (zero order) model to estimate the traveling cost. As a result the estimate of the accumulated cost at a given point will vary based on the direction of the arriving front, thereby introducing an anisotropy into the discrete algorithm, even though the continuous PDE is itself isotropic. In this chapter, we develop two very different numerically isotropic schemes to remove this anisotropy. In the first scheme, we interpolate the traveling cost along the direction of the propagating front to remove the directional bias, and in the second scheme, we upsample the traveling cost on a higher resolution grid to overcome the directional bias.

2.1 Introduction

Eikonal equation (4) is a first-order non-linear partial differential equation (PDE), which describes propagation of wavefronts in a medium. Solution to the Eikonal equation is useful in a large number of computer vision applications. Fast marching method was developed as a computationally efficient, single-pass algorithm to solve the Eikonal equation on Cartesian grids, independently, by Sethian [2,82] and Tsitsiklis [95]. Cohen and Kimmel [30,33] noticed that the minimal cost problem satisfies Eikonal equation, and developed the minimal path approach based on the fast marching method.

The popularity of fast marching in computer vision algorithms has increased since the introduction of the minimal path approach. Kimmel and Sethian used fast marching for shape from shading and optimal path planning in [57], and recently Petres *et* *al.* [74] used fast marching for path planning in autonomous underwater vehicles. Minimal path approaches have also been used in various medical image segmentation applications [6, 8, 11, 14]. Kimmel and Sethian extended the use of fast marching to compute minimal paths on manifolds [56]. Deschamps and Cohen [42] extended minimal path approach to 3D path extraction and developed a method to extract the path of centered tubular structures. Later, Li and Yezzi [62] developed fast marching-based minimal 4D paths to extract 3D tubular surfaces. Kimmel *et al.* [58] used distance maps for skeletonization and later Telea *et al.* [91] developed a skeletonization algorithm based on the fast marching technique. Fast marching techniques have also been used for image inpainting [90] and 3D surface extraction [12]. Recently, minimal path approach was used by authors in [52,94] for autonomous crack detection on asphalt pavement and concrete structure images. Lin [63] proposed deliberately-anisotropic fast marching, which was later used by Benmansour *et al.* [15] for segmentation of tubular structures like vessels.

Typically, Eikonal equation is solved numerically on discrete grids. Thus it is impossible to find the exact solution. Modifications have been suggested in [28, 40, 47] to improve the accuracy of the Fast marching method. Authors in [40, 55, 75, 83, 95] also suggest using an 8-connected neighbor scheme to improve accuracy. All the above-mentioned techniques use a locally continuous model to estimate the accumulated cost, but assume the traveling cost to be constant (discretized) around each grid point. In [83], Sethian introduced a second order fast marching scheme, which uses a second order approximation for the accumulated cost, but still uses a zero-order approximation (discretized) for the traveling cost. This scheme is very effective in improving the accuracy for fast marching when the traveling cost function is smooth. But in most applications, the desirable image features are not smooth. Only [40] interpolates τ by shifting it to the center of the grid with a nearest neighbor interpolation, but it still assumes a discretized shifted grid for τ. We present two numerically-isotropic fast marching schemes to remove the established problem of anisotropy in the numerical implementation of the Eikonal equation. In [7], we presented some preliminary results of this work.

Various approaches have also been developed to improve the computational efficiency of the traditional fast marching schemes [7,55,97,98,101,102]. The fast marching schemes we propose in this chapter, differ from traditional schemes only the treatment of the traveling cost at each grid point. Hence all these computationally efficient techniques can be readily extended to our approach.

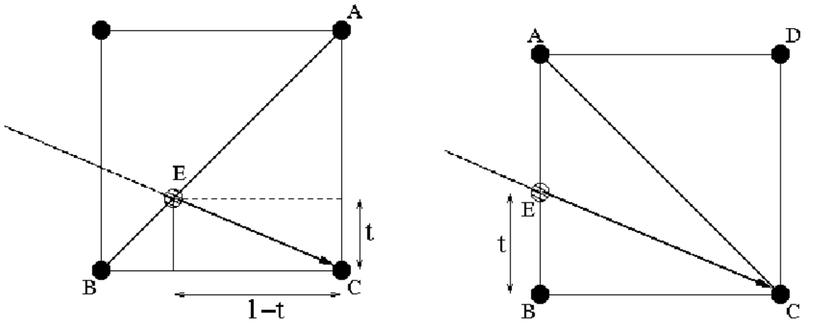
2.2 Conventional Fast Marching Schemes

Ruoy and Tourin [80] proposed a numerical scheme to solve Eikonal equation (4) on discrete 2-D Cartesian grids. For a given location (i, j) with a the traveling cost $\tau_{i,j}$, the accumulated cost $u_{i,j}$ is given by

$$\begin{aligned} &max(u_{i,j} - u_{i-1,j}, u_{i,j} - u_{i+1,j}, 0)^2 + \\ &max(u_{i,j} - u_{i,j-1}, u_{i,j} - u_{i,j+1}, 0)^2 = \tau_{i,j}^2 \end{aligned} \tag{7}$$

A single-pass algorithm to solve this equation was proposed by Adalsteinsson and Sethian [2, 82], and Tsitsiklis [95], simultaneously. Both the proposed algorithms accomplish this task by introducing an order in the selection of grid points. Adalsteinsson-Sethian named this single-pass algorithm, *Fast Marching Method*.

Although Adalsteinsson-Sethian and Tsitsiklis proposed solutions to the same problem, Adalsteinsson-Sethian used a 4-neighbor scheme, whereas Tsitsiklis used an 8-neighbor scheme. Tsitsiklis posed the problem of solving Eikonal equation as an optimal trajectory problem, whereas Adalsteinsson-Sethian used an upwind-difference scheme to solve the Eikonal equation. In this chapter, we deduce solutions to both 4- and 8-neighbor schemes using optimal trajectory approach. We also discuss the drawbacks of these traditional schemes. We then propose certain modifications to these fast marching schemes to overcome their short comings and deduce the modified solutions.



(a) 4-Connected Neighbors Scheme (Adalsteinsson-Sethian Scheme). (b) 8-Connected Neighbors Scheme (Tsitsiklis Scheme).

Figure 1: Triangulation of grid cells for the traditional fast marching schemes.

2.2.1 4-Connected Neighbor Adalsteinsson-Sethian Scheme

Consider the geometry shown in Figure 1(a), where the front is arriving at the grid point C from quadrant AB, and intersecting $\overline{AB}$ at E. The accumulated cost at E can be estimated by linear interpolation,

$$u_E = u_B(1-t) + u_A t,$$

where $0 \leq t \leq 1$. Now, the accumulated cost at C will be the sum of the accumulated cost at E and the cost accumulated in traveling along $\overrightarrow{EC}$ with a constant local traveling cost τ_C.

$$u_C = \min_{0 \leq t \leq 1} \left\{ u_B(1-t) + u_A t + \tau_C \sqrt{t^2 + (1-t)^2} \right\}. \tag{8}$$

The minimization in Equation (8) can be solved either analytically or iteratively. We can obtain the analytic solution by solving the optimality condition to minimize u_C ($\frac{du_C}{dt} = 0$). Taking the derivative of (8) with respect to t yields

$$u_A - u_B + \tau_C \Big(\frac{2t-1}{\sqrt{t^2 + (1-t)^2}} \Big) = 0. \tag{9}$$

Solving this quadratic equation leads to the closed form solution

$$u_C = \begin{cases} \frac{1}{2}\Big(u_A + u_B & \text{if } u_C > u_A, u_C > u_B \\ +\sqrt{2\tau_C^2 - (u_A - u_B)^2}\Big) & \\ min(u_A, u_B) + \tau_C & \text{otherwise.} \end{cases} \tag{10}$$

This is the exact same solution one would obtain by solving the quadratic equation in (7). Similarly, accumulated cost can be calculated for the remaining three quadrants. The minimum value of u from the four quadrants is the accumulated cost at C (u_C). Accumulated cost on the entire grid can be calculated using a min-heap structure [2]. In all future references, we call this basic 4-neighbor Adalsteinsson-Sethian scheme, the fast marching method (FMM).

2.2.2 8-Connected Neighbor Tsitsiklis Scheme

In a similar manner, one can deduce the solution for accumulated cost in the 8-neighbor scheme. Consider the geometry shown in Figure 1(b), where the front is arriving at the grid point C from the octant AB, and intersecting $\overline{AB}$ at E. For $0 \leq t \leq 1$, the accumulated cost at C is given by

$$u_C = \min_{0 \leq t \leq 1} \Big\{ u_B(1-t) + u_A t + \tau_C \sqrt{1+t^2} \Big\}. \tag{11}$$

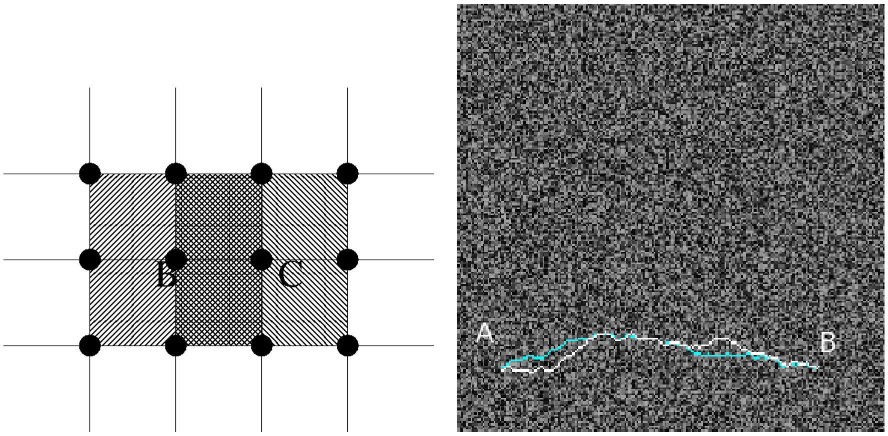
Solving $\frac{du_C}{dt} = 0$, we can derive the optimality condition to minimize u_C,

$$u_A - u_B + \tau_C \Big(\frac{t}{\sqrt{1+t^2}} \Big) = 0. \tag{12}$$

The closed form solution of this quadratic equation can be written as

$$u_C = \begin{cases} u_B + \tau_C & \text{if } u_B \leq u_A, \\ u_A + \sqrt{2}\tau_C & \text{if } \tau_C \leq \sqrt{2}(u_B - u_A), \\ u_B + \sqrt{\tau_C^2 - (u_B - u_A)^2} & \text{otherwise.} \end{cases} \tag{13}$$

Similarly, accumulated cost for fronts arriving from all the eight octants around C are calculated, and the minimum value is assigned to u_C.



(a) Influence areas of τ.   (b) Comparison of minimal paths.

Figure 2: (a) Overlap in influence areas of τ_B and τ_C. (b) Comparison of minimal paths on a random noise image.

2.2.3 Lack of Isotropy in Conventional Schemes

Consider the geometry shown in Figure 2(a). Both conventional schemes described previously use linear approximation to compute the accumulated cost at point C, but use a constant local traveling cost τ_C. Thus, the influence area of the traveling cost given at a grid point includes all the four quadrants/eight octants around it. The resulting overlap in the areas of influence for grid points B and C are shown in Figure 2(a). This suggests that the value of u_C will vary with the direction of the arriving front. Ideally, for isotropic fast marching, the accumulated cost should be independent of the direction of the arriving front. For the image shown in Figure 2(b), we use the traveling cost $\tau(x) = I(x)$, where I(x) is the intensity at each pixel. The accumulated cost in traveling from A to B should be equal to the accumulated cost in traveling from B to A. But, there is a difference in the accumulated costs because of the dependence on marching direction. Figure 2(b) compares the minimal path obtained using back propagation from B to A with the minimal path obtained by reversing the direction of front propagation. The difference in the two paths highlights the error caused by the directional dependence of the fast marching method.

We propose two numerically-isotropic fast marching schemes to overcome the above-mentioned shortcomings. The first method uses a linear/bilinear model (locally) to estimate τ along the direction of the propagating front, within each grid cell, taking the direction of arrival into consideration. We also discuss how this scheme can be made truly isotropic by removing the bias caused by the marching direction. We call this method the *Interpolated Fast Marching Method*. In the second method, we calculate u on an upsampled grid. In upsampling the grid, τ becomes constant in the neighborhood of each grid point, which eliminates the need to estimate τ with a continuous model. We call this method the *Upsampled Fast Marching Method*.

2.3 Interpolated Fast Marching Method

For interpolated fast marching scheme, we assume τ to be continuous around each grid point, and use linear/bilinear interpolation to estimate the value of local traveling cost in each grid cell. In this section, we derive the equations for linear- and bilinear-interpolated fast marching schemes. To estimate the traveling cost in a grid cell, the bilinear scheme uses the value of τ from all the grid points associated with the given quadrant. We cannot accommodate bilinear interpolation in a 4-connected neighbor scheme because a maximum of two neighbors can be used in each quadrant to calculate u. Thus, we only discuss the 8-connected neighbor scheme with bilinear interpolation.

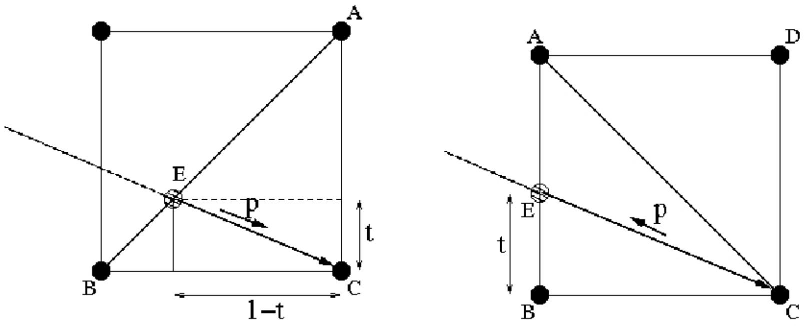
2.3.1 4-Connected Neighbors Linear Interpolation Scheme

Consider a front arriving at the grid point C from the quadrant AB, and intersecting $\overline{AB}$ at E, as shown in Figure 3(a). We use linear interpolation of local traveling cost along the path $\vec{EC}$ to compute u_C. The accumulated cost at C is

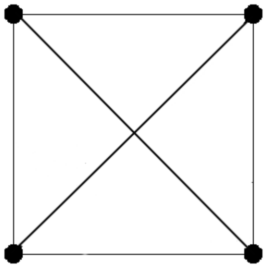
$$u_C = \min_{0 \le t \le 1} \left\{ u_B(1-t) + u_A t + \int_0^1 \tau(p) \sqrt{t^2 + (1-t)^2} dp \right\}. \tag{14}$$

The linear interpolation of the local traveling cost along $\vec{EC}$ is

$$\tau(p) = \tau_C + (\tau_A - \tau_C)p(1-t) + (\tau_B - \tau_C)pt,$$



(a) 4-Connected Neighbor Scheme. (b) 8-Connected Neighbor Scheme.



(c) Isotropic triangulation.

Figure 3: Triangulation of grid cells for interpolated fast marching schemes.

where $0 \leq p \leq 1$. Substituting $\tau(p)$ in (14) yields

$$\begin{aligned} u_C = \min_{0 \leq t \leq 1} \Big\{ & u_B(1-t) + u_A t \\ & + \sqrt{t^2 + (1-t)^2} \left(\frac{\tau_A + \tau_C}{2} + \frac{\tau_B - \tau_A}{2} t \right) \Big\}. \end{aligned} \tag{15}$$

To get the necessary optimality condition for the minimum of u_C, we solve $\frac{du_C}{dt} = 0$. Taking the derivative of (15) with respect to t yields

$$\begin{aligned} & u_A - u_B + \sqrt{t^2 + (1-t)^2} \left(\frac{\tau_B - \tau_A}{2} \right) \\ & + \frac{2t-1}{\sqrt{t^2 + (1-t)^2}} \left(\frac{\tau_A + \tau_C}{2} + \frac{\tau_B - \tau_A}{2} t \right) = 0. \end{aligned} \tag{16}$$

Algebraic manipulations on Equation (16) gives us the quartic equation,

$$\begin{aligned} & t^4(16Y^2) + t^3(16YZ - 24Y^2) + t^2(4Z^2 - 20YZ + 17Y^2 - 2X^2) \\ & + t(10YZ - 4Z^2 - 6Y^2 + 2X^2) + (Z^2 - 2YZ + Y^2 - X^2) = 0, \end{aligned} \tag{17}$$

where $X = (u_A - u_B)$, $Y = \frac{\tau_B - \tau_A}{2}$ and $Z = \frac{\tau_A + \tau_C}{2}$.

This quartic equation can be solved using Ferrari's solution [60], which is derived in Appendix A. If $Y = 0$ *i.e.*$\tau_B - \tau_A = 0$, then (16) reduces to

$$u_A - u_B + \frac{2t-1}{\sqrt{t^2 + (1-t)^2}} \left(\frac{\tau_A + \tau_C}{2} \right) = 0, \tag{18}$$

which can be solved using the closed form solution given in (10).

2.3.1.1 Newton's Method

We can also solve (16) using Newton's method. Let us consider a function,

$$\begin{aligned} f(t) = & u_A - u_B + \sqrt{t^2 + (1-t)^2} \left(\frac{\tau_B - \tau_A}{2} \right) \\ & + \frac{2t-1}{\sqrt{t^2 + (1-t)^2}} \left(\frac{\tau_A + \tau_C}{2} + \frac{\tau_B - \tau_A}{2} t \right). \end{aligned} \tag{19}$$

The solution to $f(t) = 0$, t_*, will give us the locations of minima/maxima of (16). The final accumulated cost at grid point C is the minimum of the values of u_C calculated at t_* and the boundaries $t = 0, t = 1$.

The first order derivative of $f(t)$ is given by

$$f'(t) = \frac{(2t-1)(u_A - u_B)}{\sqrt{t^2 + (1-t)^2}} + (\tau_A + \tau_C) + (8t-3)\frac{\tau_B - \tau_A}{2}. \tag{20}$$

We note that for $t = 0.5$, $f'(0.5) = \tau_C + \frac{\tau_B + \tau_A}{2}$. Thus, $f'(0.5) \neq 0$ for non-zero traveling cost functions. Thus, we can initialize the Newton's scheme with $t_{init} = 0.5$, and iteratively solve for t using the update equation

$$t_{n+1} = t_n - \frac{f(t)}{f'(t)}, \tag{21}$$

given $f'(t) \neq 0$. Typically, a single minimum/maximum of (16) lies in the range $t \in [0, 1]$. If the function is monotonic within the range $t \in [0, 1]$, $(u_C)_{min}$ will lie on the boundary. There are certain corner cases, including the possibility of multiple minima for $t \in [0, 1]$, which we discuss in detail in Appendix B. A very small number of pixels ($< \%2$ in each of our test images) fall in this category.

2.3.2 8-Connected Neighbors Linear Interpolation Scheme

Consider the triangulation of a grid cell shown in Figure 3(b) for 8-connected neighbor scheme. Using linear interpolation to estimate the local traveling cost along $\overrightarrow{EC}$, the accumulated cost at C becomes

$$u_C = \min_{0 \le t \le 1} \Big\{ u_B(1-t) + u_A t + \int_0^1 \tau(p)\sqrt{1+t^2}dp \Big\}. \tag{22}$$

Substituting $\tau(p) = \tau_C + (\tau_B - \tau_C)p + (\tau_A - \tau_B)pt$, where $0 \le p \le 1$, in (22) yields

$$\begin{aligned} u_C = \min_{0 \le t \le 1} \Big\{ & u_A t + u_B(1-t) \\ & + \sqrt{1+t^2}\left(\frac{\tau_B + \tau_C}{2} + \frac{\tau_A - \tau_B}{2}t\right) \Big\}. \end{aligned} \tag{23}$$

Again the minimizer of u_C can be obtained by solving $\frac{du_C}{dt} = 0$. Thus, we have

$$\begin{aligned} & u_A - u_B + \sqrt{1+t^2}\left(\frac{\tau_A - \tau_B}{2}\right) \\ & + \frac{t}{\sqrt{1+t^2}}\left(\frac{\tau_B + \tau_C}{2} + \frac{\tau_A - \tau_B}{2}t\right) = 0. \end{aligned} \tag{24}$$

Equation (24) can be factorized to obtain the quartic equation

$$\begin{aligned} &t^4(4Y^2) + t^3(4YZ) + t^2(4Y^2 + Z^2 - X^2) \\ &+ t(2YZ) + (Y^2 - X^2) = 0, \end{aligned} \tag{25}$$

where $X = (u_A - u_B)$, $Y = \frac{\tau_A - \tau_B}{2}$ and $Z = \frac{\tau_B + \tau_C}{2}$. This quartic equation can be solved using Ferrari's solution [60] (Appendix A). If $Y = 0$ *i.e.*$\tau_A - \tau_B = 0$, then (24) reduces to

$$u_A - u_B + \frac{t}{\sqrt{1+t^2}} \left(\frac{\tau_B + \tau_C}{2} \right) = 0, \tag{26}$$

which can be solved using the closed form solution given in (13).

2.3.2.1 Newton's method

To obtain the solution using Newton's method, we define

$$f(t) = \sqrt{1+t^2}(u_A - u_B) + (1 + 2t^2)\frac{\tau_A - \tau_B}{2} + \frac{\tau_B + \tau_C}{2}. \tag{27}$$

Taking the derivative of $f(t)$ we get,

$$f'(t) = \frac{t(u_A - u_B)}{\sqrt{1+t^2}} + 4t(\frac{\tau_A - \tau_B}{2}) + \frac{\tau_B + \tau_C}{2}. \tag{28}$$

For $t = 0$, $f'(0) = \frac{\tau_B + \tau_C}{2}$. Thus $f'(0) \neq 0$ for non-zero traveling cost functions, and we can initialize Newton's method with $t_{init} = 0$ and iteratively solve for t using the update equation (21). The corner cases with multiple minima are discussed in Appendix C.

2.3.3 Isotropic Linear Interpolation Scheme

Figures 3(a) and 3(b) show the triangulation of a grid cell for 4-neighbor and 8-neighbor schemes, respectively. Depending on the front direction, one of the quadrants/octants is chosen to estimate the accumulated cost. This induces a directional bias. To overcome this directional bias, we have to consider all possible triangulations shown in Figure 3(c). Since we are solving a minimization problem, the accumulated cost across a grid cell must be the minimum of the solutions obtained using the 4-neighbor and 8-neighbor schemes. This makes the marching scheme completely unbiased to direction. We call this scheme - the *Isolinear* scheme.

2.3.4 8-Connected Neighbors Bilinear Interpolation Schemes

The bilinear estimate of the local traveling cost along $\vec{EC}$ for triangulation of the grid cell shown in Figure 3(b) is

$$\begin{aligned}\tau(p) =&\tau_A(p)(pt) + \tau_B(p)(1-pt)\\ &+ \tau_C(1-p)(1-pt) + \tau_D(1-p)(pt),\end{aligned}$$

where $0 \le p \le 1$. The bilinear interpolation scheme is inherently independent of any directional bias within a grid cell. Substituting $\tau(p)$ in (22) yields

$$\begin{aligned}u_C = \min_{0\le t\le 1} \Big\{ &u_A t + u_B(1-t)\\ &+ \sqrt{1+t^2}\left(\frac{\tau_B+\tau_C}{2} + \frac{\tau_A-\tau_B}{3}t + \frac{\tau_D-\tau_C}{6}t\right)\Big\}.\end{aligned} \tag{29}$$

Solving $\frac{du_C}{dt} = 0$ to minimize u_C, yields

$$\begin{aligned}&u_A - u_B + \sqrt{1+t^2}\left(\frac{\tau_A-\tau_B}{3} + \frac{\tau_D-\tau_C}{6}\right)\\ &+ \frac{t}{\sqrt{1+t^2}}\left(\frac{\tau_B+\tau_C}{2} + \frac{\tau_A-\tau_B}{3}t + \frac{\tau_D-\tau_C}{6}t\right) = 0.\end{aligned} \tag{30}$$

Equation (30) can be factorized to obtain the quartic equation

$$\begin{aligned}&t^4(4(Y+Z)^2) + t^3(4W(Y+Z)) + t^2(4(Y+Z)^2 + W^2 - X^2)\\ &+ t(2W(Y+Z)) + ((Y+Z)^2 - X^2) = 0,\end{aligned} \tag{31}$$

where $X = (u_A - u_B)$, $Y = \frac{\tau_A-\tau_B}{3}$, $Z = \frac{\tau_D-\tau_C}{6}$ and $W = \frac{\tau_B+\tau_C}{2}$. This quartic equation can be solved using Ferrari's solution [60] (Appendix A). If $Y = Z = 0$ *i.e.*$\tau_A - \tau_B = \tau_D - \tau_C = 0$, then (30) reduces to

$$u_A - u_B + \frac{t}{\sqrt{1+t^2}}\left(\frac{\tau_B+\tau_C}{2}\right) = 0, \tag{32}$$

which can be solved using the closed form solution given in (13).

2.3.4.1 Newton's method

To obtain the solution using Newton's method, we define

$$\begin{aligned}f(t) =&\sqrt{1+t^2}(u_A - u_B) + (1+t^2)(\frac{\tau_A-\tau_B}{3} + \frac{\tau_D-\tau_C}{6})\\ &+ t(\frac{\tau_B+\tau_C}{2} + \frac{\tau_A-\tau_B}{3}t + \frac{\tau_D-\tau_C}{6}t).\end{aligned} \tag{33}$$

Taking the derivative of $f(t)$ we get,

$$f'(t) = \frac{t(u_A - u_B)}{\sqrt{1+t^2}} + 4t(\frac{\tau_A - \tau_B}{3} + \frac{\tau_D - \tau_C}{6}) + \frac{\tau_B + \tau_C}{2}. \tag{34}$$

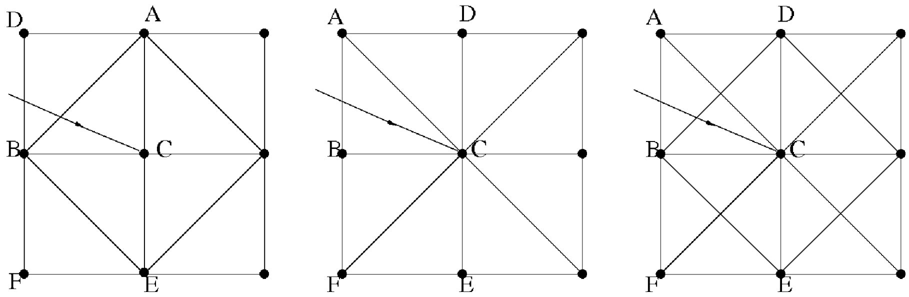
We see that for $t = 0$, $f'(0) = \frac{\tau_B + \tau_C}{2}$. Thus, $f'(0) \neq 0$ for non-zero traveling cost functions and we can initialize Newton's method with $t_{init} = 0$ and iteratively solve for t using the update equation (21). The corner cases with multiple minima are discussed in Appendix C.

2.3.4.2 Comparison of Computation Time

Since Newton's scheme exhibits quadratic convergence for all interpolated fast marching schemes (Appendix B & C), three iterations were sufficient for convergence. Fixing the number of iterations in each update step also ensures that we have the same computation complexity in each update step; making it suitable for hardware implementation. Newton's scheme has fewer (logical and mathematical) operations in comparison to the Ferrari (analytic) scheme; hence, using Newton's scheme is computationally efficient. We compare the computation time on a 500x500 random noise image for the Ferrari's scheme and Newton's scheme. The computation time for analytic and Newton's schemes are listed in Table 1. We call the 4-connected and 8-connected neighbor linear-interpolated fast marching schemes, Linear-4 and Linear-8 schemes, respectively, and the 8-connected neighbor bilinear-interpolated fast marching scheme, Bilinear-8. The computation time was measured on a laptop with a 1.73 GHz processor.

Table 1: Comparison of computation time.

	Linear-4	Linear-8	Bilinear-8
Analytic (Ferrari)	1.51s	2.83s	3.23s
Newton's scheme	0.51s	0.52s	0.65s



(a) 4-Connected Neighbors Scheme. (b) 8-Connected Neighbors Scheme. (c) Isotropic Fast Marching Scheme.

Figure 4: B is the newly $accepted$ grid point and u_C is to be computed.

2.3.5 Marching Forward Loop

We differ from traditional marching techniques only in the calculation of accumulated cost and not in the marching order. Thus, we still follow the main loop of the basic fast marching scheme [82]. When a *trial* point is *accepted* in the min-heap structure, we compute the value of u from both quadrants/octants that include the newly *accepted* point, and replace the newly calculated u with the minimum of the two solutions and the existing value of u (if the point is marked as *trial*).

Consider the example shown in Figure 4(a), where B is the newly *accepted* point, and the accumulated cost at neighbor C is to be computed. As opposed to the basic fast marching technique, u_C does not solely depend on u_A,u_B,u_E and the local traveling cost τ_C, but it also depends on the accumulated and traveling costs at all the other 8-connected neighbors. Thus, using the quadrant containing the minimum of u_A and u_E will not necessarily guarantee the minimum solution to (15). We have to consider both the quadrants that contain B. If the front also arrives at C from the other two quadrants, they will be considered when the corresponding neighbors become *accepted*. The same argument can be extended to the 8-connected neighbor case shown in Figure 4(b). We need to calculate u_C from the two octants containing $\overline{AB}$ and $\overline{FB}$, once point B is *accepted*. In the isolinear fast marching scheme, we consider the possibility of front arriving from all quadrants and octants (Figure

4(c)).

Since we depart from the traditional fast marching method only in the update procedure for the accumulated cost, we can use the various computationally efficient techniques developed for the traditional fast marching scheme. Fast sweeping methods [101], group marching method [55], untidy priority queue-based marching [98] and algorithms which exploit parallel processing, such as [97], can also be readily extended to our approach for hardware implementation.

2.4 Upsampled Fast Marching Method

In the previous section, we discussed how interpolating traveling cost along the arriving front can remove the directional bias. In this section, we discuss a second approach to overcome the problem posed by the overlap in influence areas of τ (Figure 2). Consider the upsampled grid shown in Figure 5, where the solid circles correspond to grid points on the original grid. Now, we see that there is no overlap in the influence areas of τ on the upsampled grid. Since the traveling cost is unique in each grid cell, there is no directional bias in the calculation of u. We first compute u on the upsampled grid, and then downsample the output on the original grid.

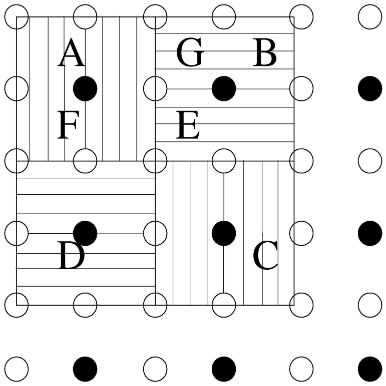


Figure 5: No overlap in influence areas of τ_A, τ_B, τ_C, τ_D.

2.4.1 4-Connected Neighbors Upsampled Scheme

In the upsampled grid, τ is unique in each quadrant around a grid point. The value of τ chosen to calculate u is unique for a given direction of the arriving front. For example, if the front arrives at E from the north-west, then we use τ_A (Figure 5). At the point G, we use τ_A for a front arriving from the west and τ_B for a front arriving from the east. We use τ_A to calculate u_A, irrespective of the direction of the arriving front. Since the value of τ is constant along the direction of the front at a sub-pixel level, it is not necessary to assume a locally continuous model to interpolate τ. Thus, the accumulated cost at E for a front arriving from the north-west is

$$u_E = \min_{0 \le t \le 0.5} \left\{ u_F t + u_G(0.5 - t) + \tau_A \sqrt{t^2 + (0.5 - t)^2} \right\}. \tag{35}$$

This minimization leads to the closed form solution

$$u_E = \begin{cases} \frac{\left(u_F + u_G + \sqrt{\delta}\right)}{2} & \text{if } \delta \ge 0, \\ min(u_F, u_G) + \frac{\tau_A}{2} & \text{otherwise,} \end{cases}$$

where $\delta = \frac{\tau_A^2}{2} - (u_F - u_G)^2$.

2.4.2 8-Connected Neighbors Upsampled Scheme

As in the case with 4-connected neighbors scheme, τ is unique in each octant around a grid point in the upsampled grid. We note that exactly one grid point in each associated quadrant corresponds to a grid point in the original grid; we use the corresponding value of τ to compute u.

By following the procedure described in Section 2.3.5, we calculate u from the two octants that contain the newly *accepted* point. If F is the newly *accepted* point, we calculate u_E in the octants containing $\overline{FA}$ and $\overline{FD}$ (Figure 5). The solution is the minimum of the two values obtained. Thus, for a front arriving from north-west, the accumulated cost at E is

$$u_E = \min_{0 \le t \le 0.5} \left\{ u_A t + u_F(0.5 - t) + \tau_A \sqrt{0.5 + t^2} \right\}, \tag{36}$$

which yields the closed form solution

$$u_E = \begin{cases} u_F + \frac{\tau_A}{2} & \text{if } u_F \leq u_A, \\ u_A + \sqrt{2}\frac{\tau_A}{2} & \text{if } \tau_A \leq 2\sqrt{2}(u_F - u_A), \\ u_F + \frac{\sqrt{\tau_A^2 - 4(u_F - u_A)^2}}{2} & \text{otherwise.} \end{cases}$$

2.5 Numerical Experiments

In this section, we present a few experiments to compare the proposed methods to the basic fast marching method (FMM) [82], Tsitsiklis scheme [95], shifted-grid fast marching (SGFM) [40] and multi-stencil fast marching (MSFM) [47]. We also compare the 4-connected neighbor upsampled (Up4) and 8-connected neighbor upsampled (Up8) schemes with the upsampled version of the SGFM scheme (upSG). All the minimal paths in this section are generated using a sub-pixel level back propagation scheme, to minimize potential errors that can be introduced by the back propagation procedure.

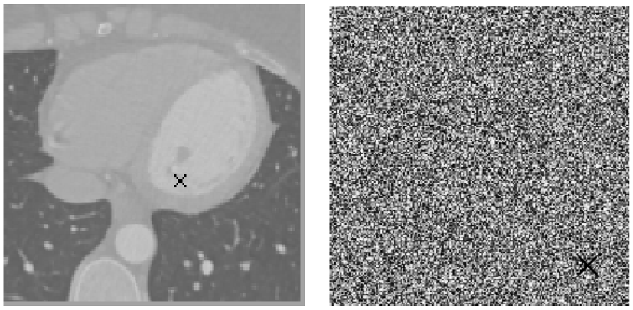
2.5.1 Back Propagation Error and Accuracy

For the two images shown in Figure 6, we calculate u over the entire domain from a randomly chosen point (marked by 'X'). Then we compare the value of u at each point with the total cost in propagating a front from that point to the point marked 'X'. The difference in the two values is the back-propagation error. The average error over the entire image domain gives us average back-propagation error (ABPE). The numerical values of ABPE for the various schemes are listed in the Table 2. For the cardiac image, we use the cost function $\tau(x) = \frac{1}{1+|\nabla I|^2}$, and for the random noise image, we use $\tau(x) = I(x)$.

In the next example, we compare the accuracy of the various techniques for the following cost functions:

$$\tau_1(x, y) = 1/20\sqrt{(sin\tfrac{x}{20}cos\tfrac{y}{20})^2 + (cos\tfrac{x}{20}sin\tfrac{y}{20})^2},$$

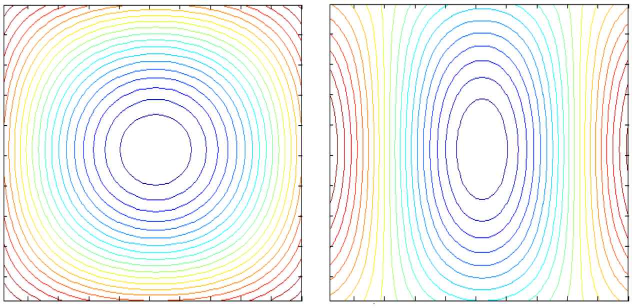
$$\tau_2(x, y) = 1/10\sqrt{(sin\tfrac{x}{20}cos\tfrac{y}{10})^2 + (cos\tfrac{x}{20}sin\tfrac{y}{10})^2}.$$



(a) Cardiac Data (b) Random noise

Figure 6: Test images used to compute Average Back Propagation Error (ABPE).

The isocontours of analytic accumulated cost, $u_{analytic}$, on a 50x50 grid for τ_1 and τ_2 are shown in Figure 7. The geodesics to the center $(26, 26)$ of the grid will be straight lines for τ_1, and curved lines for τ_2. Since, we have the analytic solution for these cost functions, we compare the L_1, L_2 and L_∞ norms for each scheme.



(a) Isocontour of u_1. (b) Isocontour of u_2.

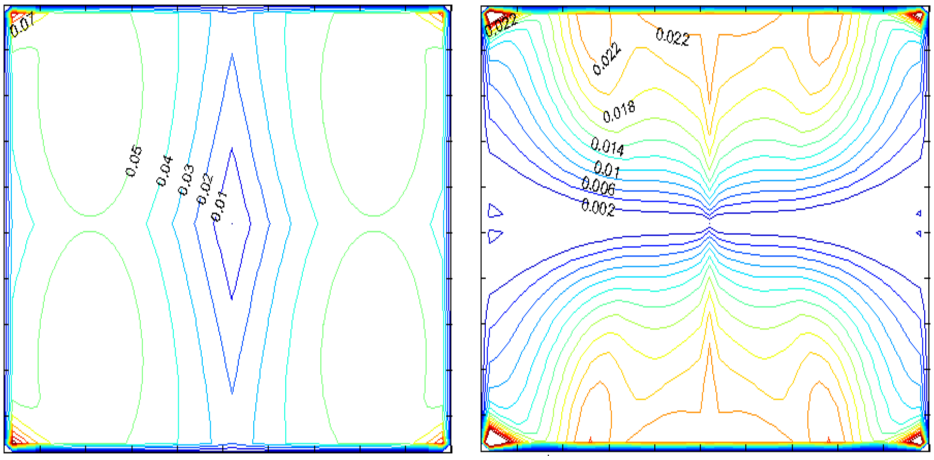
Figure 7: Isocontours of analytic accumulated cost functions.

$$L_1 = \text{mean}(|u - u_{analytic}|),$$

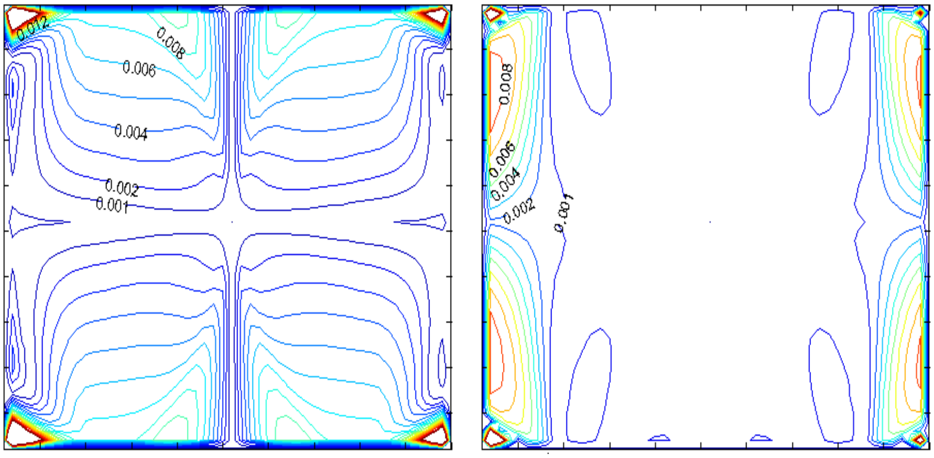
$$L_2 = \text{mean}(|u - u_{analytic}|^2),$$

$$L_\infty = \max(|u - u_{analytic}|).$$

The numerical errors are listed in Table 2. Notice that the error norms show significant improvement for the proposed methods, especially in the case with curved geodesics (τ_2). The isocontours of error for τ_2 using FMM, SGFM, isolinear and up8 schemes are shown in Figure 8.



(a) FMM (b) SGFM



(c) Isolinear (d) Upsampled-8

Figure 8: Isocontours of errors for cost function τ_2.

We enlist computation time for each scheme in the last column of Table 2. We measured computation time on a $500\text{x}500$ random noise image using a laptop with a 1.73 GHz processor.

2.5.2 Segmentation and Minimal Path Results

In Figure 9, we present left ventricle segmentation results on a 2D cardiac image. To segment the image we pick a point on the boundary of the object and detect all saddle points (described in detail in Section 3.2.2 of Chapter 3). From each saddle point, we obtain

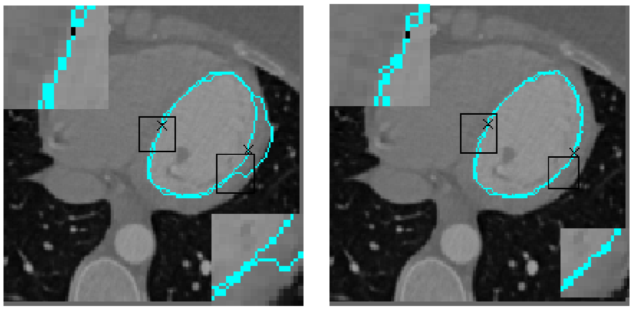
Table 2: Error norms for τ_1 and τ_2, ABPE, and computation time.

	τ_1			τ_2			ABPE		Time (s)
	L_1	L_2	L_∞	L_1	L_2	L_∞	I_1	I_2	
FMM	$2.5\text{x}10^{-2}$	$6.7\text{x}10^{-4}$	0.038	$4.4\text{x}10^{-2}$	$2.1\text{x}10^{-3}$	0.106	0.073	0.390	0.27
Tsitsiklis	$2.1\text{x}10^{-2}$	$4.9\text{x}10^{-4}$	0.028	$3.8\text{x}10^{-2}$	$1.6\text{x}10^{-3}$	0.083	0.101	0.435	0.26
MSFM	$2.4\text{x}10^{-2}$	$6.1\text{x}10^{-4}$	0.035	$4.2\text{x}10^{-2}$	$1.9\text{x}10^{-3}$	0.101	0.083	0.357	0.29
SGFM	$2.3\text{x}10^{-3}$	$6.3\text{x}10^{-6}$	0.005	$1.3\text{x}10^{-2}$	$2.1\text{x}10^{-4}$	0.058	0.002	0.028	0.33
Linear4	$1.1\text{x}10^{-2}$	$1.7\text{x}10^{-4}$	0.029	$1.7\text{x}10^{-2}$	$4.0\text{x}10^{-4}$	0.088	0.012	0.104	0.51
Linear8	$2.3\text{x}10^{-3}$	$6.8\text{x}10^{-6}$	0.005	$4.4\text{x}10^{-3}$	$3.4\text{x}10^{-5}$	0.060	0.003	0.036	0.52
IsoLinear	$2.3\text{x}10^{-3}$	$6.8\text{x}10^{-6}$	0.005	$4.0\text{x}10^{-3}$	$3.1\text{x}10^{-5}$	0.060	0.011	0.091	0.91
Bilinear8	$2.7\text{x}10^{-3}$	$9.4\text{x}10^{-6}$	0.005	$5.0\text{x}10^{-3}$	$4.1\text{x}10^{-5}$	0.061	0.003	0.010	0.65
Up4	$1.8\text{x}10^{-3}$	$7.6\text{x}10^{-6}$	0.010	$3.1\text{x}10^{-3}$	$2.9\text{x}10^{-5}$	0.066	0.045	0.19	1.37
Up8	$3.0\text{x}10^{-4}$	$2.0\text{x}10^{-7}$	0.001	$1.5\text{x}10^{-3}$	$7.8\text{x}10^{-6}$	0.029	0.001	0.02	1.42
UpSG	$2.0\text{x}10^{-3}$	$4.1\text{x}10^{-6}$	0.004	$1.2\text{x}10^{-2}$	$1.9\text{x}10^{-4}$	0.057	0.002	0.01	1.42

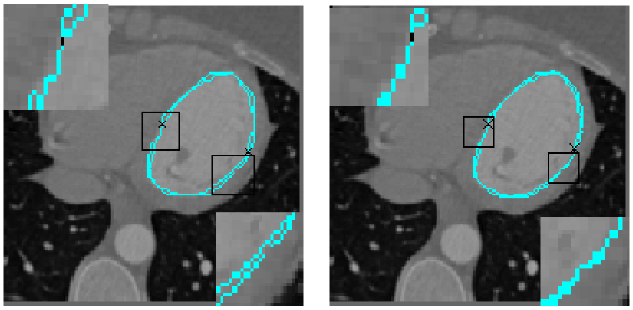
two minimal paths back to the initial point; each set of two paths gives us a closed segmentation curves. We choose the saddle point whose corresponding closed segmentation curve has the minimum Chan-Vese [24] energy. Images in Figure 9 show the overlay of segmentation curves initialized with two different user given points on the boundary. We see that the segmentation curves are not consistent and they depend on initialization. This inconsistency is due to the difference in the marching direction, which is exaggerated in the presence of weak image features. We highlight certain regions in these images to compare the segmentation results obtained using the different marching schemes.

In Figure 10(a), we present a concrete structure image with a crack. Figure 10(b) shows the corresponding crack detected manually. For semi-automated crack detection, we provide two endpoints on the image to detect the crack (using $\tau(x) = I(x)$). We compare the results generated by various schemes with the manually detected crack in Figure 10.

Next, we compare minimal paths obtained by reversing the marching direction for the same set of end points (Figure 11). We see that the proposed isolinear fast marching scheme consistently detects the crack in both directions. The inherent anisotropy in traditional schemes reduce the consistency in crack detection. Thus, accuracy in crack detection varies with the marching direction.



(a) FMM (b) Tsitsiklis



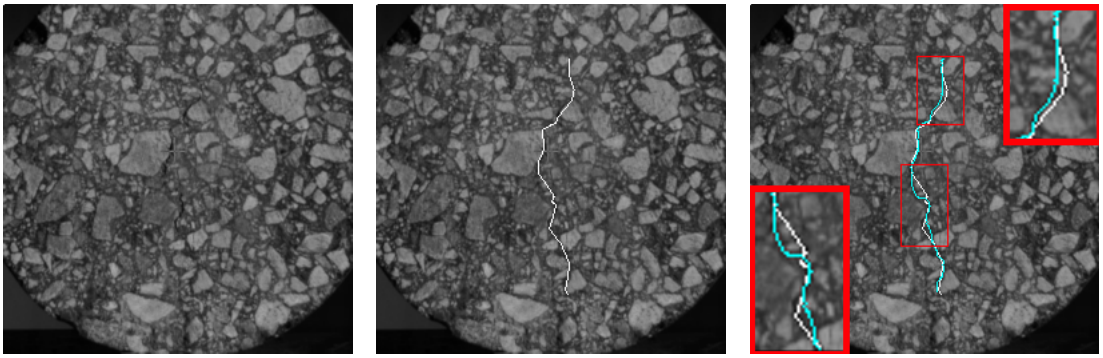
(c) SGFM (d) Isolinear

Figure 9: A comparison of Left Ventricle segmentation with two different initializations.

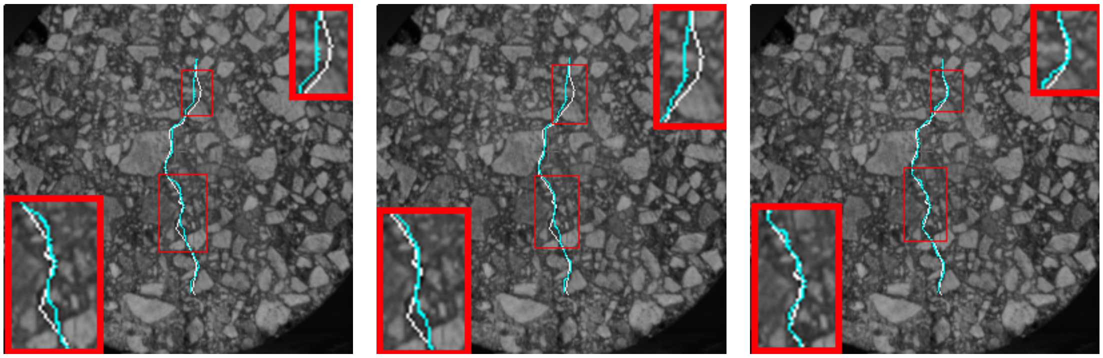
In Figure 12, we present an asphalt road image with a crack. These images were captured by an automatic crack sealing machine (ACMS). The ACMS has a robotic arm to apply sealant to the detected crack, without human intervention [94]. Thus, it is very important to detect the crack very accurately, irrespective of the moving direction of the vehicle housing the ACMS. The low-contrast in the captured image makes this task very challenging. Figures 12(b-f) compare the consistency in the detected path among the various fast marching schemes.

In Figure 13, we compare the manually detected crack with the minimal path detected using the traditional fast marching scheme, isolinear scheme and upsampled fast marching scheme. As we can see, the accuracy of the traditional schemes vary with the marching direction.

Finally, we compare the minimal paths obtained on a random 200x200 noise image in

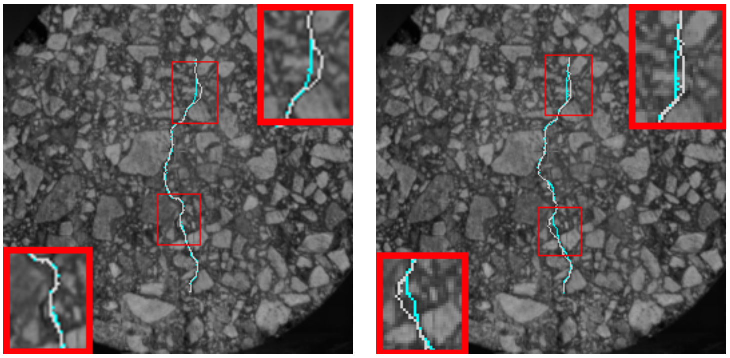


(a) Concrete Structure (b) Manually detected crack (c) FMM

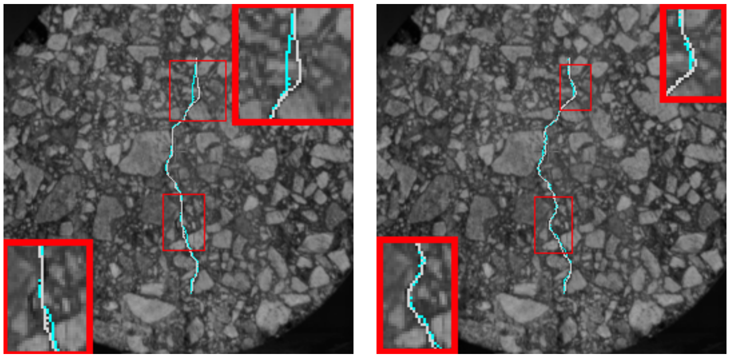


(d) MSFM (e) SGFM (f) Isolinear

Figure 10: Comparison of crack detection in concrete structures: (a) Initial stage of crack development in a concrete structure. (b) Manually detected crack. (c-f) Comparison of minimal paths for crack detection on concrete images with the manually detected crack.

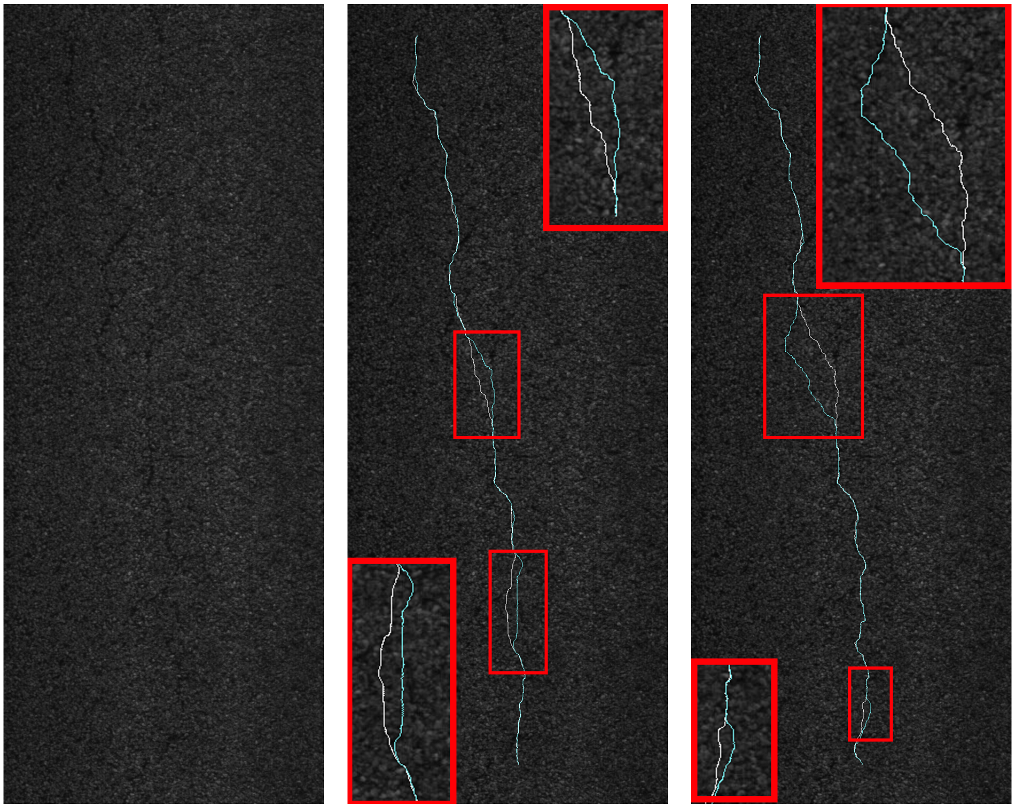


(a) FMM (b) MSFM



(c) SGFM (d) Isolinear

Figure 11: Comparison of minimal paths for crack detection on concrete images generated by changing marching direction.


(a) Asphalt Road Image (b) FMM (c) MSFM

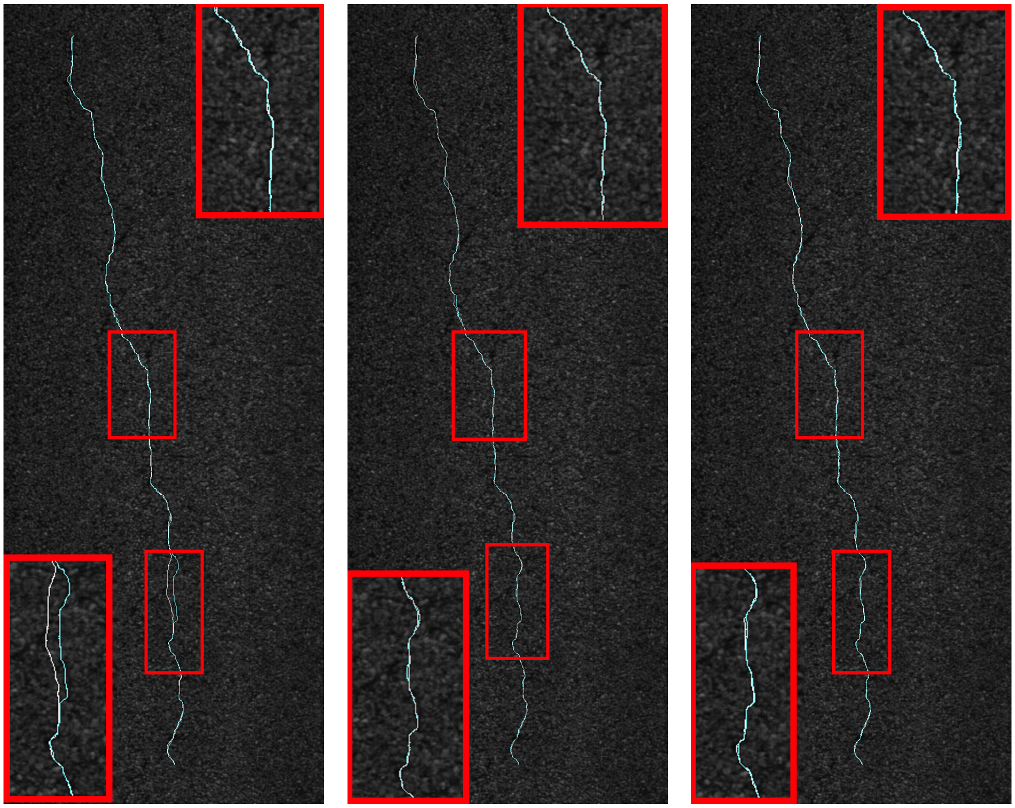
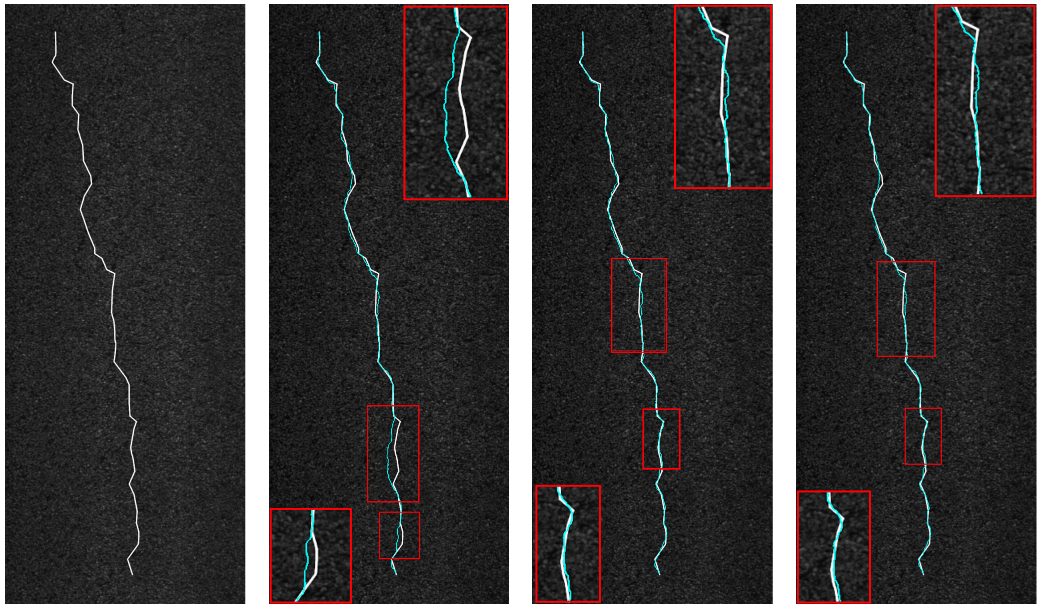

(d) SGFM (e) Isolinear (f) Upsampled

Figure 12: Comparison of minimal paths for crack detection on asphalt road images generated by changing marching direction.



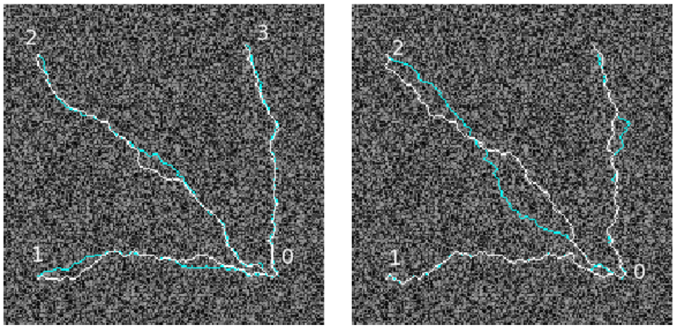
(a) Manually detected crack (b) FMM. (c) Isolinear (d) Upsampled

Figure 13: Comparison of crack detection results with manually detected crack map: We compare the manually detected crack (White Path) with the generated minimal path (Cyan Path).

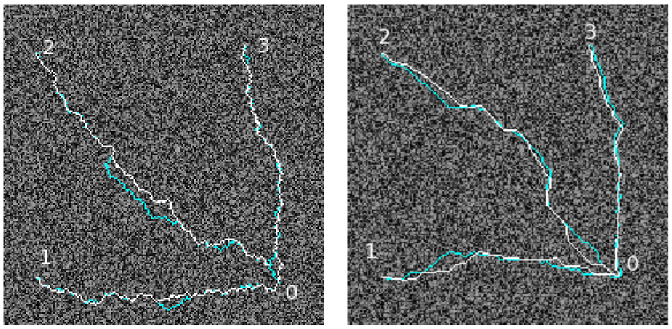
traveling from point ‘0’ to points ‘1’,‘2’ and ‘3’, with the corresponding minimal paths obtained by reversing the marching direction (Figure 14). We see that the proposed schemes give consistent paths even in the absence of strong image features. The results also validate the ABPE values listed in Table 2. The ABPE for the Tsitsiklis scheme is the highest and accordingly, the paths obtained with the Tsitsiklis scheme show a lot of variation. Although the average error for SGFM is low, the minimal paths obtained using SGFM show significant variations. Interpolation of the cost function in SGFM artificially smooths the traveling cost, which reduces the significance of the image features.

2.6 Conclusion

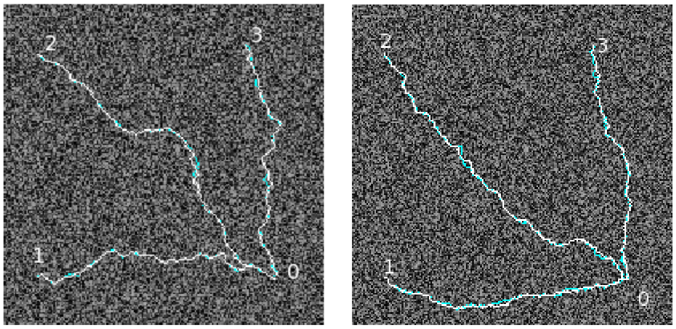
We have presented two techniques to remove the established problem of directional bias in the numerical implementation of fast marching and improve the accuracy of the fast marching method. One approach interpolates the local traveling cost along the front and the other computes accumulated cost on an upsampled grid. We showed that combining the 8 and 4-connected neighbor schemes further reduces the inaccuracy by considering all possible directions of the arriving front. We have compared both our approaches to the existing fast marching techniques and show significant improvements. Although both our approaches have higher computation time, they can be implemented efficiently on hardware and they are practical solutions to eliminate the inaccuracies of existing fast marching techniques.



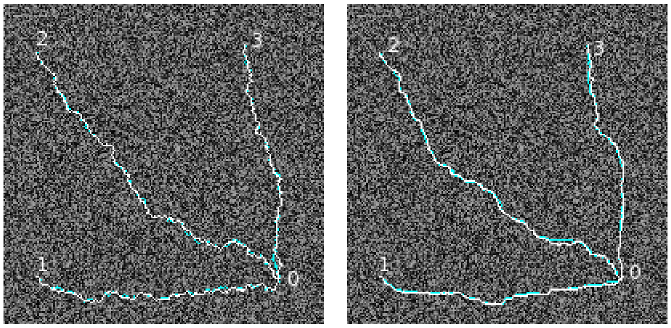
(a) FMM (b) Tsitsiklis



(c) MSFM (d) SGFM



(e) Isolinear (f) Bilinear



(g) Upsampled (h) Upsampled shifted grid

Figure 14: Minimal Paths on a random noise image: Comparison of minimal paths obtained by reversing the marching direction.

CHAPTER III

ACTIVE GEODESICS: A REGION-BASED ACTIVE CONTOUR MODEL WITH A GLOBAL EDGE-BASED CONSTRAINT

In this chapter, we develop an *active geodesic* model, which constrains any evolving region-based active contour to continually be a geodesic with respect to an edge-based metric through its entire "active" evolution rather than just at its final state (as in the traditional *geodesic active* contour models). Since the contour is always a geodesic throughout the evolution (during its "active" state), we automatically get a degree of global optimality with respect to the edge-fitting criterion. This enables us to construct a purely region-based energy minimization model without having to devise arbitrary trade-offs in the combination of our energy function to balance edge-based terms with the region-based terms. This approach of integrating edge information as a *geodesic constraint* while optimizing a purely region-based energy yields a new class of active contours that exhibit both local and global behaviors. Additionally, these *active geodesics* are naturally responsive to intuitive types of user interaction.

3.1 Introduction

The need to find solutions to active contour models that are less susceptible to local minima has led to the development of various modifications to traditional edge- and region-based energy minimization models. In the past, various papers have developed complicated schemes and modifications, either to exploit edge information with less sensitivity to local minima [11, 18, 30, 31, 69, 73], or to utilize region-based energy models

[25,76,81,93,99,104], which can reduce sensitivity to local minima. Although, the region-based models exhibit far less sensitivity to local minima, they demand contrast and homogeneity levels that are often unrealistic for a broad class of imagery. The *active geodesic* model developed in this chapter tries to solve the problem by minimizing one form of energy (region-based) with a global constraint of a second form of energy (edge-based).

The earliest approach that converted the purely edge-based local optimization problem, introduced in [51], into a global optimization problem was developed by Cohen and Kimmel in [30]. They converted the energy minimization problem into a cost accumulation problem, which can be solved with a single-pass Fast Marching algorithm [2]. This accumulated cost depended only on the location of a given point, and not on the geometry of the curve at that point. Open geodesics could now be computed as minimal paths between two specified end points. Using this minimal path approach Cohen and Deschamps [32] developed an interactive edge-based segmentation algorithm, which is initialized by the user with a few points on the object boundary. Sun *et al.* [87] introduced circular shortest paths, which used the minimal path approach to detect closed curves with coinciding end points. Later, Appleton *et al.* [10, 11] introduced the globally optimal geodesic active contour (GOGAC) model, which initializes segmentation with a single user-given point inside the object of interest.

In contrast to these edge-based active contour models, the region-based curve evolution techniques, which were developed later, define energy functionals based on region statistics rather than local image gradients [25, 67, 76, 81, 93, 99, 104]. In general, these region-based segmentation models are less sensitive to noise and initialization when compared to the edge-based models. Since these region-based models make strong assumptions about homogeneity of the image, they fail to capture the relevant edges in certain cases.

Therefore, while region-based segmentation algorithms are less susceptible to local minima, edge-based segmentation algorithms have a better chance of detecting edges along the object boundary. Paragios *et al.* [73] used a linear combination of a probability based active region model with the classical edge-based model to exploit the benefits of both approaches. The final segmentation curve had certain desirable properties of both models. But a fixed weight for the linear combination may not be suitable for all kinds of images. Chakraborty *et al.* [23] introduced a game-theory based approach to combine region- and edge-based models. In addition to sensitivity to the choice of linear weighting factors, such energy-based schemes that employ combination of edge- and region-based terms may yield new classes of local minima that represent unsatisfactory *compromises* of these two criteria.

All the segmentation models discussed so far treat the image as a continuous function; the models are defined in the continuum and they converge to either a local or global minimum for a given energy. A second class of image segmentation algorithms view images as a graph or a collection of discrete points [17, 45, 66, 77, 84]. The intelligent scissors algorithm, described in [66], requires the user to specify several points on the object boundary to generate a segmentation curve. Algorithms in [17, 45, 89] require the user to specify a few pixels belonging to the interior and the exterior of the object of interest. Grab cuts [77], developed later, asks the user to specify a bounding box around the object of interest along with the labelled pixels. These interactive algorithms can generate arbitrary segmentation results based on user interactions.

Here, we develop an *active geodesic* model that constrains the evolving active contour to continually be a geodesic with respect to an edge-based metric through its entire "active" evolution rather than just at its final state. Thus, we call our model the *active geodesic* model as opposed to the traditional *geodesic active contour* model. With the closed geodesic constraint in effect during the "active" evolution state we devise a purely region-based energy minimization model. We use the Chan-Vese [25] (region-based energy) model in all examples presented here, but our model is generic to all other forms of statistical region-based energy functions. Since the contour is a closed geodesic during the entire evolution process, we are guaranteed to have a global form of optimality with respect to an edge-based metric. Thus, the *active geodesic* exhibits edge optimality not only in the final converged state, but throughout the curve evolution process.

Maintaining global edge optimality throughout the region-based evolution process has the following powerful advantages over other existing techniques:

1. Edge optimality reduces the search space for region-based optimization by reducing the number of degrees of freedom (infinite dimensional problem to a finite dimensional problem) to significantly improve the robustness of our algorithm.
2. All intermediate results are geodesics which are *globally* affected by image edges rather than arbitrary contours which can respond locally to spurious structure in the image (including noise). This enables *active geodesics* to avoid a larger class of local minima when compared to the traditional *geodesic active contour* evolution models that are globally edge optimal only in their final converged state.
3. The final segmentation will be a region-based minimum with the edge constraint, and not a compromise achieved by an energy model with a weighted combination of region- and edge-based energies.

In Section 3.4, we also develop an interactive segmentation algorithm based on this *active geodesic* model. The user initializes the algorithm by identifying a single point in the interior of the object of interest. Subsequently the user can interact with the algorithm, if necessary, by placing additional *repellers* and *attractors* to repel or attract the *active geodesic* towards desired edges. These *repellers* and *attractors* vary the edge-based metric locally, which in turn has a global effect on the *active geodesic*. This lends a very intuitive response and interpretation to the user interaction in our approach.

3.2 Coupling Region- and Edge-based Segmentation

In this section, we describe the coupling of region- and edge-based segmentation using the minimal path approach [30]. We first discuss an extension of the minimal path approach to detect closed geodesics and then incorporate region-based energy to find a globally accurate segmentation.

3.2.1 Edge-based Segmentation using Minimal Paths

For edge-based minimal paths, we use the Eikonal equation from (4) with an edge-based travelling cost function of the form

$$\tau(x) = g\left(\frac{1}{1 + \|\nabla I\|}\right) + \epsilon, \tag{37}$$

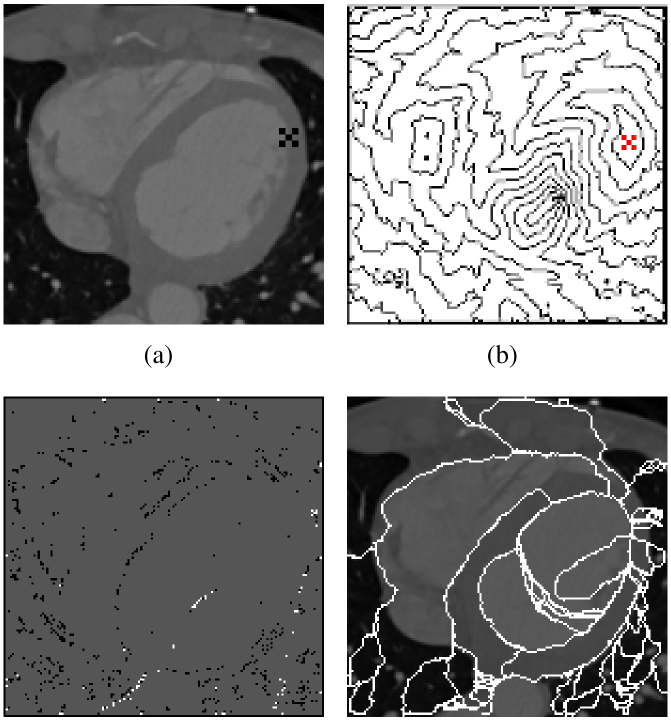
where $g(\cdot)$ is a monotonically increasing function, $\epsilon > 0$ is a regularizer and ∇I denotes the gradient of the image at a given location. In all the examples presented in this chapter we use the Interpolated Fast Marching scheme (Refer Section 2.3) with a monotonic function of the form $g(x) = x^m$, where $m \geq 1$. To extract the shortest path between two points, we calculate u by propagating wavefronts from one of the two points (source point) to the other (end point). Then, by following the gradient descent in the vector field $\vec{\nabla} u$, we trace our path back to the source point. The path obtained is the globally optimal open geodesic (shortest path) between the two points.

3.2.1.1 Detecting Closed Curves using Minimal Paths

If two different global minimal paths exist between two points on the image, the two open geodesics complete a closed contour. Now, consider a single source point given in the image domain. To detect closed curves, we have to find points on the domain from which two global minimal paths (back to the single source) exist, *i.e*, the two paths have the same accumulated cost. These special points are called saddles of u [29] and they can be interpreted as the points where the propagating fronts collide. The colliding fronts also indicate that the two open geodesics forming the closed curve have the same tangent at the saddle point. Note that the two open geodesics may or may not have the same tangent at the source point. Thus the closed geodesic formed by combining two open geodesics is guaranteed to be truly geodesic at all the points except at the source point.

Since two minimal paths form a single closed curve, there will be a unique saddle point associated with each closed curve. We can associate two descents and two ascents in u to each saddle point [29]. Thus, we can detect saddle points by calculating the number of level crossings of u around each point in the domain. Most of the points on the grid have two level crossings whereas the saddle points will have four level crossings.

Consider the cardiac image shown in Figure 15(a) where the source point (marked 'X') is placed on the boundary of the object of interest. Figure 15(b) shows the level set representation of the propagating wavefronts from the source. Figure 15(c) and Figure 15(d) show the various detected saddle points and closed curves associated with each detected saddle point, respectively.


(a) (b)

(c) (d)

Figure 15: (a) Cardiac image with source point marked by an 'X'. (b) Level set representation of the propagating wavefronts. (c) Saddle point map: The white pixels in the image indicate the saddle points. The white pixels indicate four level crossings around them, gray pixels indicate two level crossings and the black pixels indicate zero level crossings. (d) Closed geodesics associated with each detected saddle point.

3.2.2 Shock Curves

As we can see in Figure 15(c), several saddle points are obtained for a single source point. So, we need to find a way to weed out the unwanted saddle points. Authors in [30] present a technique to weed out the unwanted saddle points by filtering them based on the edge information in the image. They make an assumption that a saddle point closer to an edge is relevant. Then they present the reduced set of saddle points to the user, who chooses the appropriate saddle point to form the closed curve. Instead, we propose an elegant approach to detect the closed curve by finding the appropriate saddle point without any user interaction.

Let us first formalize the notion of relevance in the context of saddle points. Consider a small circle of infinitesimal radius around the source point. The closed geodesic contour segmenting the object will contain two minimal paths which descend (against the gradient of u) towards the source, within the circle, from two different directions. Thus, we can deem a saddle point as relevant if the minimal paths obtained by descent arrive at the source from two different directions. The closed curve formed by this saddle point is truly a closed geodesic at all the points except the source point (The two open geodesics may not have the same tangent at the source).

One way to detect relevant saddle points (closed geodesics) is the brute-force approach. For numerical implementation of this approach, let us approximate the infinitesimal circle around the source with the 8 pixel neighborhood around the source point. Now, we generate minimal paths back to the source from each saddle point and find the saddle points for which the minimal paths arrive (at the source) from two different neighboring pixels. This exhaustive approach is computationally expensive.

Another way to find these relevant saddle points (closed geodesics) is by tracking the fronts as they propagate in different directions from the source. For numerical implementation, we again approximate the infinitesimal circle with the 8 pixels neighborhood around the source. We now assign a different label to each of the 8 neighboring pixels. As we calculate the accumulated cost u by propagating the front in the upwind direction, we also propagate the label in the direction of the front.

Figure 16(a) shows an illustration of the fronts emanating out of the source in different directions. Each gray level indicates a different neighbor of the source point from which the front at any given location propagated. We also observe that the fronts arriving from two different directions form shock curves when they meet. By definition, the locations on these shock curves where the fronts arrive from two exactly opposite directions (collide) are the saddle points. The minimal paths from saddle points lying on the shock curves arrive at the source from two different directions, forming closed geodesics. Thus, saddle points lying on the shock curve are the relevant saddle points. In Figure 16(b) we can see the shock curves; the saddle points lying on these curves are highlighted in red.

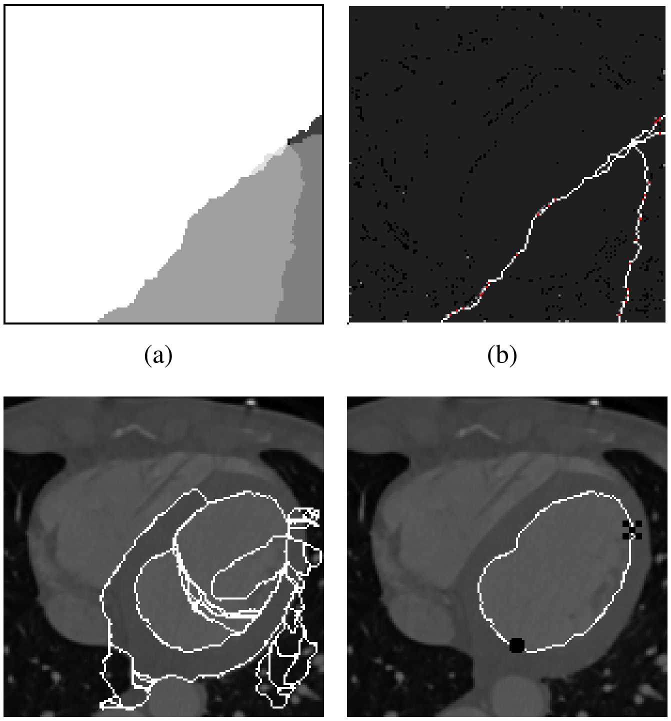
The minimal paths from the other points on the shock curve also arrive at the source from two different directions, but the two paths do not have the same accumulated cost. Thus, they form a set of two open geodesics with different tangents at both the end-points. Saddle points are isolated points on a shock curve which form curves that are closed geodesic at the shock curve.

3.2.3 Incorporating Region-based Energy

We reduce the number of saddle points under consideration with the aid of shock curves. Now, the next step is to choose the appropriate saddle point, which segment the object. We begin with the assumption that the object surrounded by the edge also exhibits certain region-based properties (homogeneity). We now compare the region-based energy for all closed geodesics formed by the saddle points. Let us consider the Chan-Vese [25] energy function of the form

$$E = (I - \mu)^2 + (I - \nu)^2, \tag{38}$$

where μ is the mean of pixel intensities inside the curve and ν is the mean of pixel intensities outside the curve, respectively. Among all detected closed curves, the curve with



(a) (b)

(c) (d)

Figure 16: (a) Each gray level indicates a different neighbor of the source point from which the front at the given location propagated. The intersection of these labelled regions form the shock curves. (b) Shock curve: Saddle points lying on the shock curve are marked in red. (c) Closed geodesics associated with the relevant saddle points. (d) Closed geodesic with minimum region-based energy. The source point is marked by an ‘X’ and the saddle point is marked by the dot.

the minimum region-based energy segments the object. From the various closed geodesics shown in Figure 16(c), the closed geodesic with the minimum region-based energy segments the left ventricle as shown in Figure 16(d).

Until this point, we assumed that our source point was on the boundary of the object of interest. This meant that the saddle point associated with the closed geodesic, exhibiting least region-based energy, also fell on the object boundary. Consider the source point shown in Figure 17(a), which is placed away from the object boundary. The segmentation obtained using the approach described previously does not segment the left ventricle. Thus, for a given source point, we see that the minimal path approach guarantees global edge optimality, but it does not guarantee that the closed curve also corresponds to a minimum of the region-based energy.

3.3 Active Geodesic Contour Model: Region-based Active Contours Model with Global Edge-based Constraints

In this section, we describe the region-based active contour evolution approach with a global edge-based constraint. Conventional curve evolution models treat energy minimization as an infinite dimension problem, where each point on the curve can evolve independently. Instead, in our *active geodesic* model, we perturb the saddle point based on the region-based energy of the curve. Perturbing the saddle point causes an indirect evolution of the curve and this curve evolution has only two degrees of freedom, the two co-ordinates of the saddle point. We treat the new location of the saddle point as our source and detect closed curves as described in the previous section. We continue evolution until we reach a minimum for the region-based energy. The minimal path approach ensures global edge optimality. Thus, our segmentation evolves purely based on the region-based energy, with an in-built global edge optimality. We show some examples in Section 3.5, which indicate that we can converge to the final segmentation in a few iterations with our approach. This is a result of dimensionality reduction in the optimization problem.

3.3.1 Level set Representation

We seek to represent the curve (geodesic) C obtained with the minimal path approach as a level set function. For given u, the minimal geodesic is described as the gradient descent path [11,30]

$$\frac{\partial C}{\partial s} = -\frac{\nabla u}{|\nabla u|}, \tag{39}$$

where s is the arc length parameter.

Consider a level set function ψ that will form the embedding level set function for the geodesic C. The geodesic C now becomes a level set of this function. By definition, the value along any level set (geodesic) in ψ is constant. Thus the derivative along a given level set is '0',

$$\frac{\partial \psi(C)}{\partial s} = 0. \tag{40}$$

Representing the geodesic in two dimensional co-ordinate system, $C = (x_1, x_2)$, and using chain rule, we get

$$\begin{aligned} \frac{\partial \psi}{\partial x_1} \cdot \frac{\partial x_1}{\partial s} + \frac{\partial \psi}{\partial x_2} \cdot \frac{\partial x_2}{\partial s} &= 0. \\ \Rightarrow \nabla \psi \cdot \frac{\partial C}{\partial s} &= 0. \end{aligned} \tag{41}$$

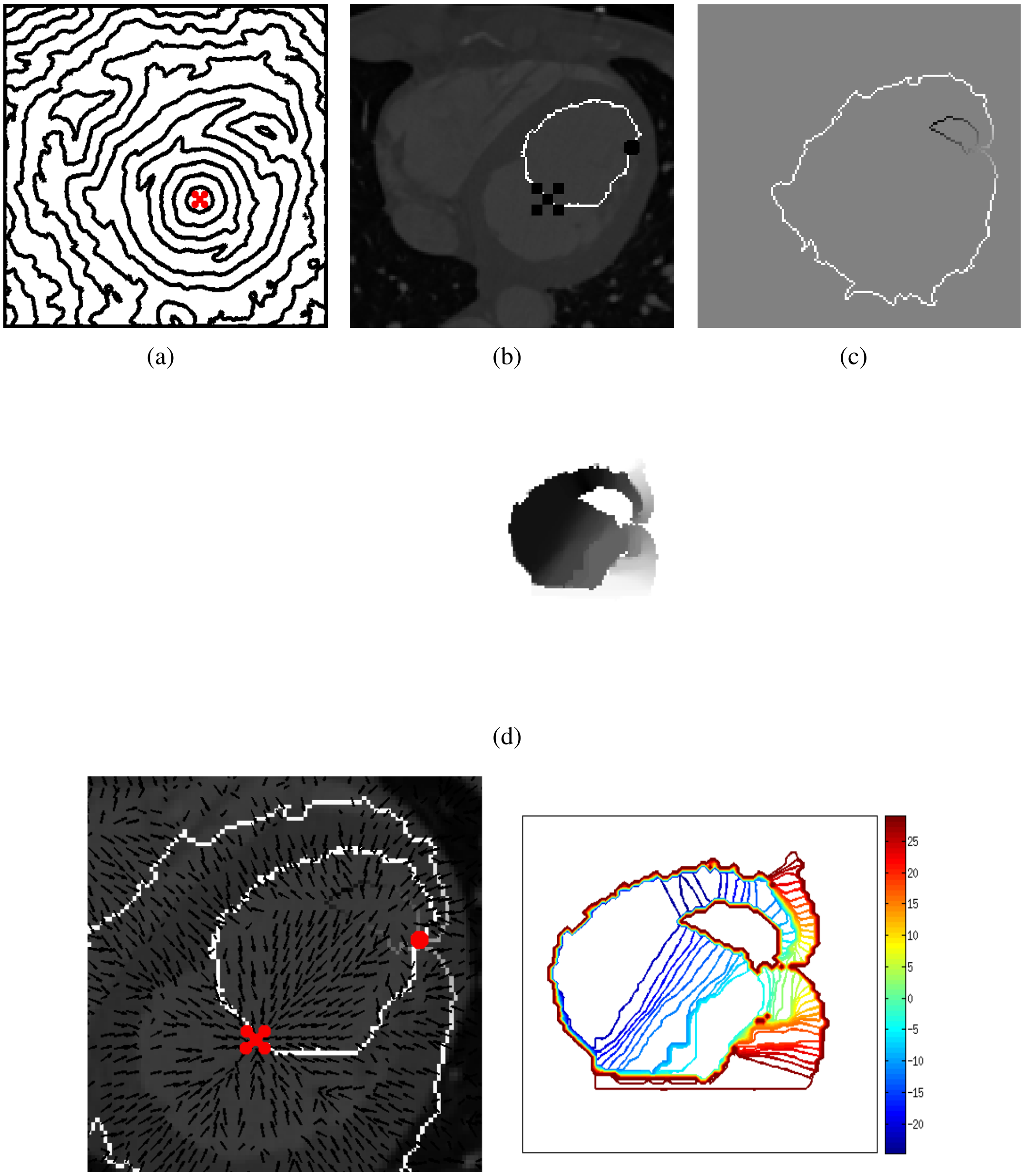
Substituting (39) in (41) we get,

$$\begin{aligned} \nabla \psi \cdot \Big(-\frac{\nabla u}{|\nabla u|} \Big) &= 0. \\ \Rightarrow \nabla \psi \cdot (-\nabla u) &= 0. \end{aligned} \tag{42}$$

Let S denote the saddle point on the given curve C. The level set corresponding to the value of accumulated cost at the saddle point, $u(S)$, forms a boundary, ∂R_S, to the vector field $-(\vec{\nabla u})$, where

$$R_S = \{(x, y) \in \mathbb{R}^2 : u(x, y) \leq u(S)\}. \tag{43}$$

We parametrize the boundary, ∂R_S, with a linear function $\psi(x) = p$, as shown in Figure 17 (c). The value of ψ varies linearly as we move away from the saddle point. Solving the



(a) (b) (c)

(d)

(e) (f)

Figure 17: (a) Level set representation of the propagating wavefronts (u). (b) Closed curve with minimum region-based energy and the associated saddle point. (c) Boundary ∂R_S associated with saddle point S parametrized with a linear function $\psi(x) = p$. (d) Generated level set function, ψ. (e) Vector field $-(\nabla u)$. (f) Level set function ψ.

PDE given in (42), we can propagate ψ in the direction of the characteristics of $-(\vec{\nabla u})$, to form a level set function in the region R_S. The level set function ψ forms an implicit representation of the curve C.

The PDE in (42) can be solved using the standard Fast-sweeping Method [100, 101], which requires cyclically alternating the order of traversal using the ordering of the rectangular grid for a few iterations, to update the value of ψ. The number of iterations required depends on how convoluted the trajectories corresponding to the vector field $-(\vec{\nabla u})$ are.

A second approach, the Ordered Traversal approach [100], can be used to converge to the solution with a single sweep across the grid points within region R_S. To decide the order of traversal, we define a length function L, where $L(x)$ is the arc length of the trajectory, corresponding to the vector field $-(\vec{\nabla u})$, between ∂R_S and given location x. Such a length function must satisfy the following first-order linear PDE [100]:

$$\nabla L \cdot (-\nabla u) = \ 1, \quad \text{with } L(\partial R_S) = 0. \tag{44}$$

We can solve the PDE in (44) with a min-heap data structure similar to the one used in Fast Marching schemes [2, 7, 40]. As we update $L(x)$ at a grid location, we simultaneously update the value of $\psi(x)$ using (42) at that grid location. Since, the values of L and ψ are calculated in the order in which the characteristic curves flow from the boundary ∂R_S, only one sweep across all the grid points in region R_S is required.

If we choose a function such that $\psi(S) = 0$, then the closed geodesic C will be embedded as the zero level set of ψ. Figures 17 (d) and (f) show the level set function ψ generated by solving the PDE in (42) for the gradient vector field $-(\vec{\nabla u})$ shown in Figure 17 (e) with the boundary values in Figure 17 (c).

3.3.2 Energy Minimization

Our goal is to minimize the region-based energy for the edge-optimal geodesic C. Let us begin by considering a general class of region-based energy, E, of the form

$$E(C) = \int\limits_{\Omega} f(x)dA. \tag{45}$$

We can represent the gradient of $E(C)$ w.r.t the parameter p as the line integral

$$\frac{\partial E}{\partial p} = \int\limits_{C} f \cdot \frac{\partial C}{\partial p} \cdot \vec{N} ds, \tag{46}$$

where s is the arc length parameter of the curve and $\vec{N}$ is the outward normal to the curve C. Since the level set function ψ is a function of the parameter p, as well as the curve C, we have

$$\psi(C, p) = p. \tag{47}$$

Representing the geodesic as $C = (x_1, x_2)$, and taking the gradient of (47), we get

$$\begin{aligned}
&\frac{\partial \psi}{\partial x_1} \cdot \frac{\partial x_1}{\partial p} + \frac{\partial \psi}{\partial x_2} \cdot \frac{\partial x_2}{\partial p} = 1. \\
&\Rightarrow \nabla \psi \cdot \frac{\partial C}{\partial p} = 1. \\
&\Rightarrow \frac{\nabla \psi}{\|\nabla \psi\|} \cdot \frac{\partial C}{\partial p} = \frac{1}{\|\nabla \psi\|}. \\
&\Rightarrow \frac{\partial C}{\partial p} \cdot \vec{N} = \frac{1}{\|\nabla \psi\|}.
\end{aligned} \tag{48}$$

Since C is embedded in level set function ψ, both C and ψ have the same outward normal, $\vec{N} = \frac{\nabla \psi}{\|\nabla \psi\|}$. Thus, substituting (48) in (46) we get

$$\frac{\partial E}{\partial p} = \int\limits_{C} \frac{f}{\|\nabla \psi\|} ds. \tag{49}$$

The factor of $\|\nabla \psi\|$ in the denominator of the line integral varies the contribution of each point on the curve. Since, the curve evolution is solely based on saddle point evolution, the points on the curve closer to the saddle point have higher significance. As we see from

Figures 17 (d) and (f), the value of $\|\nabla\psi\|$ increases as we move away from the saddle point on the zero level set (geodesic). Thus, the points on C closer to the saddle point have a higher contribution to the integral than the points further away.

In the examples discussed here, we use the Chan-Vese [25] energy model (38). For the Chan-Vese model, f takes the form

$$f = 2(\mu - \nu)\{I - \frac{\mu + \nu}{2}\}. \tag{50}$$

We now perturb the saddle point S against the gradient $\frac{\partial E}{\partial p}$, in the direction normal to the curve. The value of the line integral in (49) governs how we perturb the saddle point. By definition saddle points are isolated points on the shock curve. Hence, the two open geodesics back to the source from the new location of the saddle point will not be truly geodesic at the new location (*i.e.* tangents will not agree). Thus, we make the saddle point at the new location our source point. We now recompute u from this new source point. This generates several saddle points. We weed out the unwanted saddle points using the procedure discussed in 3.2.2. From the remaining saddle points we pick a saddle point that satisfies the following two conditions:

1. The associated closed geodesic has a lower region-based energy when compared to the region-based energy of the closed geodesic obtained prior to perturbing the saddle point.

2. It lies closest to the previous source point.

The accumulated cost u from the previous iteration is used as the metric to measure distance from the previous source point and not the Euclidean distance. Picking a saddle point, which satisfies both conditions ensures that the region-based energy for the closed geodesic decreases gradually with each iteration. We follow this procedure until we converge to a minimum *i.e.* when none of the saddle points satisfy the two conditions. This final curve will be a minima for the region-based energy under the global edge-based constraint, respectively.

3.4 Interactive Segmentation Algorithm

Using the *active geodesic* evolution model described in the previous section, we present an interactive segmentation algorithm. The user initializes segmentation by placing a single point within the desired object of interest. If the segmentation result generated by *active geodesic* model is not the desired result, the user can interact with the algorithm by placing *repellers* and *attractors* to repel or attract the *active geodesic* towards a desired segmentation.

3.4.1 Attractors and Repellers

For each *repeller* (P) and *attractor* (Z) the user places, we update the travelling cost function locally by placing a pole and zero, respectively. The updated travelling cost functions are given by

$$\tau'(x) = \tau(x) \circ h_1\Big(\frac{1}{\text{distance}(x, P)}\Big), \tag{51}$$

$$\tau'(x) = \tau(x) \circ h_2\Big(\text{distance}(x, Z)\Big), \tag{52}$$

where h_1 and h_2 are monotonically increasing functions, the '$\circ$' operator represents Hadamard product[1], *distance*$(\cdot, \cdot)$ is the Euclidean distance and $h_1(x) = h_2(x) = x^2$. The regularizer ϵ in (37) ensures that a *repeller* has the desired influence even when placed near strong edges. By placing these *attractors* and *repellers* the user is locally modifying the travelling cost (edge-based metric). This local variation has a global effect on the *active geodesic* evolution as we will see in the examples in Section 3.5.

3.4.2 Algorithm Details

We initialize the algorithm by asking the user to place a single point within the desired object of interest. We then place a *repeller* at this location. This artificially placed *repeller* serves the following two purposes:

---

[1] Hadamard product is the entry-wise product of two matrices. For two given matrices A_{mxn} and B_{mxn}, $(A \circ B)_{i,j} = (A)_{i,j} \cdot (B)_{i,j}$.

1. It identifies the object of interest.
2. It ensures that the propagating wavefronts wrap around the pole to guarantee that at least one closed geodesic exists.

Now, we randomly pick a point in the vicinity of the *repeller* (different from the *repeller*), as the source point, and follow the procedure described in Section 3.3. In the very first iteration we do not have a reference for source from previous iteration, hence we choose the saddle point closest to the *repeller* placed by the user. We move the saddle point against the gradient $\nabla_p E$ to minimize the region-based energy of the *active geodesic*. This saddle point becomes the source for the second iteration. We continue the evolution described in Section 3.3 until we converge to a minimum. Figure 18 shows the evolving *active geodesic* and the final segmentation of the left ventricle for the cardiac image.

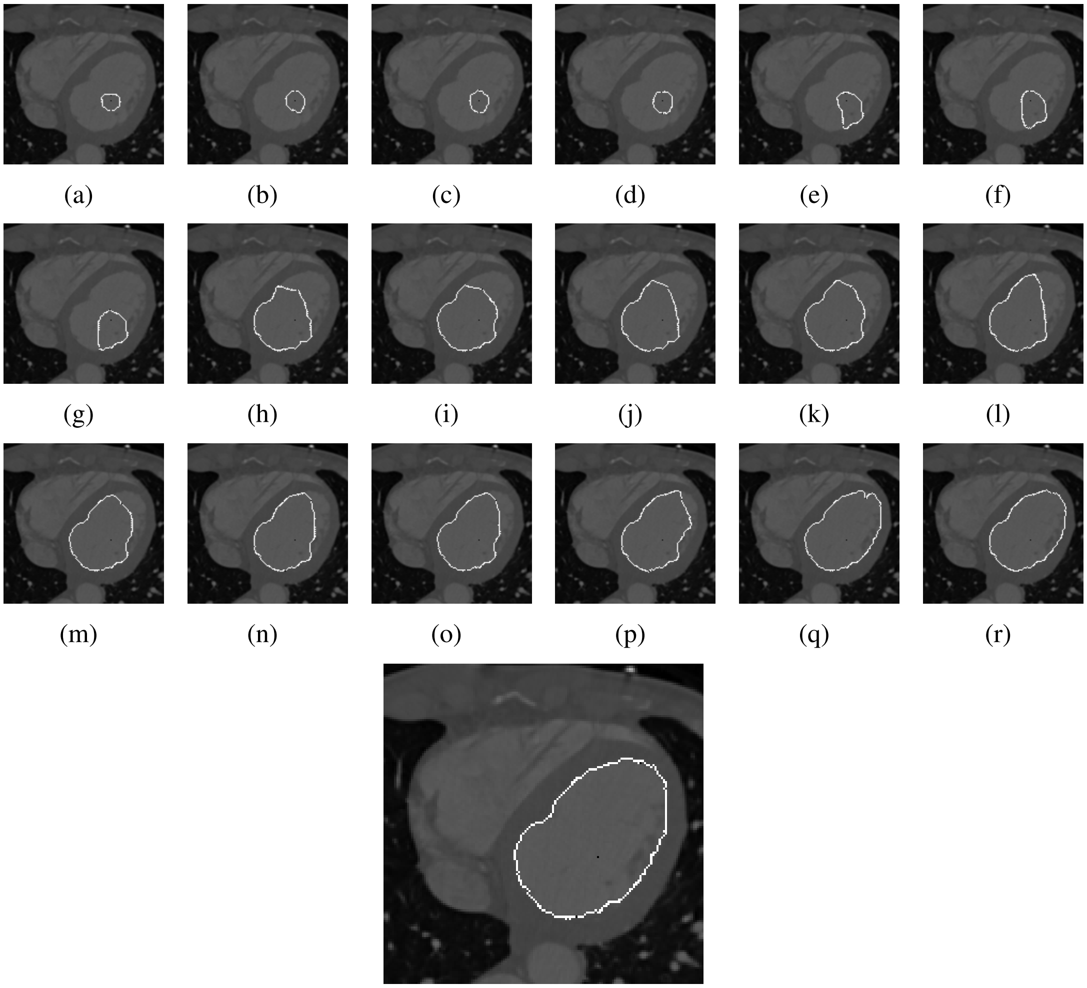
Once we converge to a minimum, we present the user with the resultant closed geodesic. If the user is not satisfied with the segmentation result he can add an *attractor* or a *repeller* to drive the *active geodesic* towards the desired edges. Consider the segmentation result shown in Figure 19(a). Placing another *repeller* inside the closed contour further evolves the *active geodesic* away from the new *repeller* as shown in Figure 19(b).

The *repellers* placed by the user are classified as interior or exterior *repellers* based on their location with respect to the current state of the *active geodesic*. We choose only those saddle points that form closed geodesics that separate all the interior *repellers* from the exterior *repellers*. This ensures that a *repeller* placed inside the closed curve lies inside the final segmentation and a *repeller* placed outside the closed curve stays outside the final segmentation. No such constraint is placed on the *active geodesic* based on the location of *attractors* added by the user.

Placing a few more *attractors* and *repellers*, the user can converge to the final desired segmentation of the right ventricle as shown in Figure 19(b). A brief outline of the algorithm is given in Table 3.

Table 3: Pseudo-code for the interactive segmentation algorithm.

```
do
  if First iteration = TRUE
    Initialize algorithm with a repeller inside the object of interest.
    Update the travelling cost function.
    Choose a random point other than the repeller as the initial source point.
  else
    Add a repeller or an attractor and update the travelling cost function.
  end
  do
    if First iteration = FALSE
      Perturb the saddle point based on region-based energy and make it the new
      source point.
    end
    Propagate wavefronts from the source point.
    Detect shock curves and the associated saddle points.
      if First iteration = TRUE
        Find the saddle point closest to the repeller.
      else
        Find the saddle point lying closest to the source in the previous iteration,
        which also minimizes region-based energy and separates the interior
        and exterior repellers.
      end
  while Convergence = FALSE
  Complete the closed contour using the current source and saddle point to obtain the
  converged segmentation.
while Desired Segmentation = FALSE
```



(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

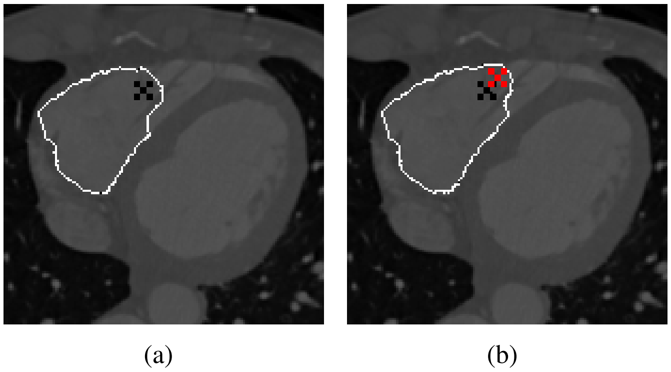
(m) (n) (o) (p) (q) (r)

(s)

Figure 18: Left Ventricle segmentation with proposed *active geodesic* model: (a) Segmentation after the first iteration. (b-r) Evolution of the closed curve to minimize the region-based energy. (s) Final converged segmentation after 19 iterations.

3.4.3 Computational Complexity

The three major tasks associated with the numerical implementation of the outlined segmentation algorithm are propagating the wavefronts from the source point to generate u, detecting the saddles of u and generating the level set function ψ. Since the source point changes in each iteration, we need to recompute u, saddles of u and ψ in each iteration. The computational complexity of generating u and detecting the saddles of u for an image with grid size N are $O(NlogN)$ and $O(N)$, respectively [30]. The level set ψ exists only



(a) (b)

Figure 19: The initial *repeller* is marked by a black 'X'. (a) Converged segmentation of the right ventricle with a single *repeller* inside the ventricle. (b) Converged segmentation after the user places a second *repeller*.

in the region R_S given in (43). The number of pixels in region R_S, which covers only a fraction of the total grid size, can be denoted by $(k \cdot N)$, where $k < 1$. The computational complexity of calculating ψ with the min-heap data structure described in Section 3.3.1 is $O((k \cdot N)log(k \cdot N))$. Thus, the overall computational complexity of our algorithm is of the order $O(NlogN)$.

3.5 Experimental Results

The *active geodesic* model combines a global edge-based constraint with a region-based energy minimization model in a framework that enables intuitive interactions to generate non-arbitrary segmentation results with a sense of global optimality. Since our model minimizes region-based energy with a global edge constraint in an interactive segmentation algorithm, we compare the results of our method with the following existing segmentation algorithms:

1. **Active Contours Without edges (Chan-Vese Segmentation) model** [24]

   As opposed to the proposed global edge-constrained *active geodesic* model, the Chan-Vese model is an unconstrained region-based energy minimization model. The algorithm is initialized with a user specified initial contour. By minimizing the energy given in (38), we converge to the final segmentation result. Figure 20 shows segmentation results for the left and right ventricles in the cardiac image shown in Figure 15 (a). We can see that this purely region-based model fails to segment the ventricles because it does not take any edge information into consideration.

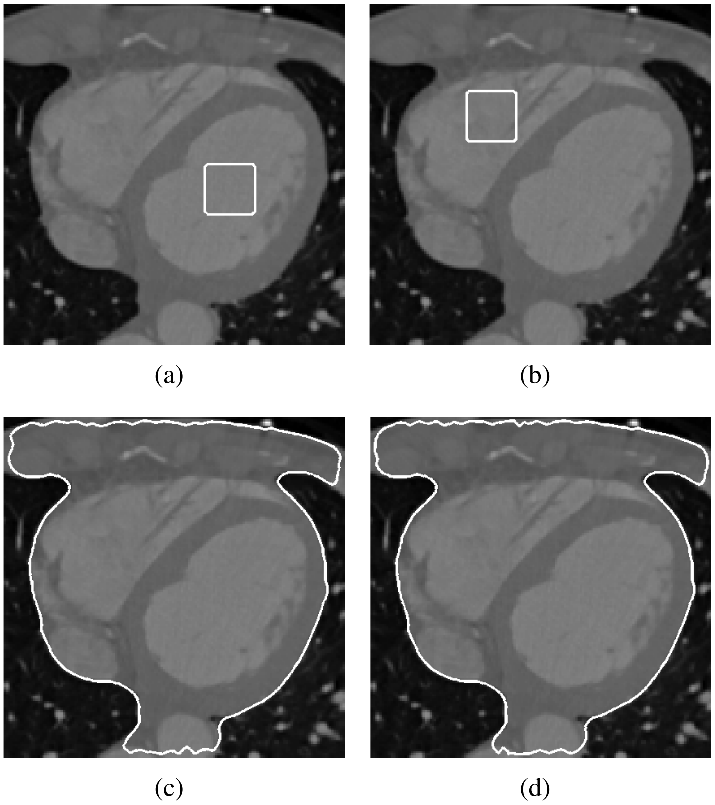


Figure 20: Segmentation of ventricles with the purely region-based Chan-Vese model: (a,b) Initialization for the left and right ventricle segmentation. (c,d) Final region-based, Chan-Vese segmentation.

2. **Globally Optimal Geodesic Active Contour (GOGAC) model** [11]

GOGAC is a purely edge-based segmentation model, which generates globally optimal edge-based segmentation. The algorithm is initialized with a single user-given point inside the object of interest. The edge-based travelling cost in (37) is then modified to

$$\tau'(x) = \frac{1}{r}\left(g\left(\frac{1}{1+\|\nabla I\|}\right)+\epsilon\right), \tag{53}$$

where r is the Euclidean distance from the user-specified point. A globally optimal edge-based segmentation is obtained using a minimal path approach by inducing an artificial cut in the image domain. The procedure followed by the algorithm does not allow further user interactions to improve segmentation results.

In Figure 21, the GOGAC-model-based left and right ventricle segmentations are initialized by the user with points marked by black 'X's. We obtain the globally optimal closed geodesic with the vertical cut induced in the image plane as shown in the Figures 21 (a) and (b), respectively. Although, the final segmentation curves are global minima with respect to the edge-based travelling cost in (53), they fail to segment the ventricles due to the presence of strong, misleading edges inside the ventricles.

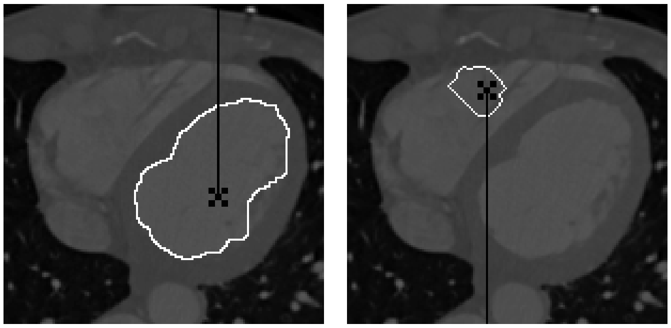

(a) (b)

Figure 21: GOGAC based ventricle segmentation: (a,b) Left and Right Ventricle segmentation using the purely-edge-based-GOGAC model initialized with the points marked by 'X' within the ventricles.

3. **Linear Combination of Edge- and Region-based Energies** [73]

In this model, the active contour minimizes a linear combination of edge- and region-based energies,

$$E = \alpha \cdot E_{reg} + (1-\alpha) \cdot E_{edge}, \tag{54}$$

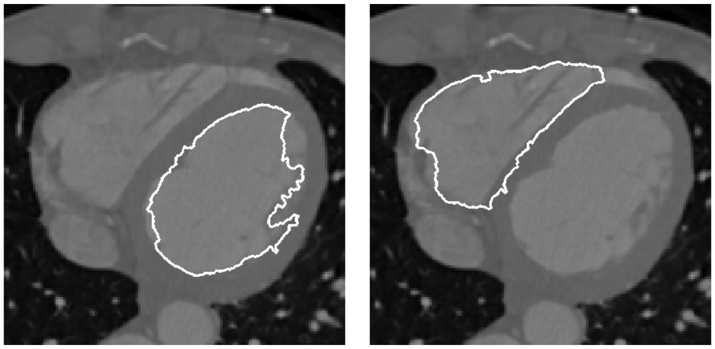
where $\alpha \in (0,1)$. We again use the Chan-Vese energy (38) as our region-based energy in this linear combination. The edge-based energy is of the form

$$E_{edge} = \int_C \phi \cdot ds, \tag{55}$$

where,

$$\phi = \frac{1}{(1 + ||\nabla I||^2)}. \tag{56}$$

Figure 22 shows the left and right ventricle segmentation initialized with contours shown in Figures 20 (a) and (b), respectively. As the contour converges to the edges of the ventricles, it gets stuck in certain local minima within the ventricles. Further, it may become necessary to heuristically change the weight of the linear combination for different scenarios (Results presented here were generated with $\alpha = 0.75$). In addition to sensitivity to the choice of α, such linear combination yield new classes of local minima that represent unsatisfactory *compromises* of edge- and region-based models.



(a) (b)

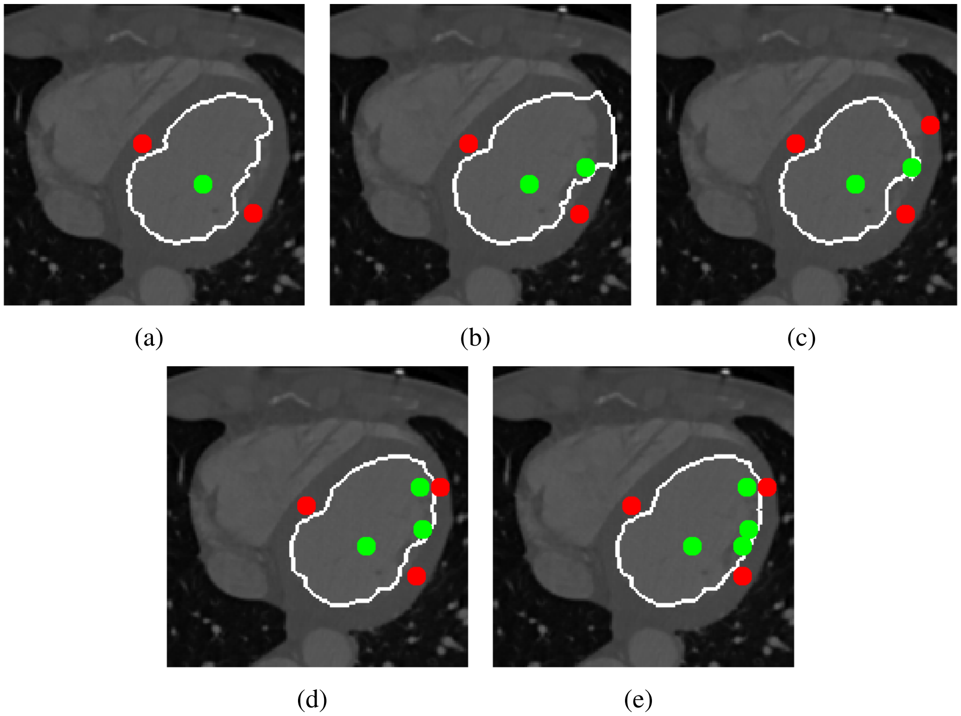
Figure 22: Segmentation of ventricles with linear combination of region- and edge-based energies: (a,b) Final segmentation of left and right ventricles obtained by minimizing the energy in (54), when initialized with curve given in Figures 20 (a) and (b), respectively.

4. **Random Walker Segmentation** [45]

This is an interactive segmentation algorithm, which is initialized with a few pixels

> on the image with user-defined labels. Based on an intensity-weighted metric, the algorithm determines the probability of a random walker starting from each unlabelled pixel to reach a pre-labelled pixel. Assigning each pixel, with the label corresponding to highest probability generates the segmentation result.
>
> Analogous to the interactive segmentation model described in Section 3.4, we present the user with the segmentation result generated with the initial few labelled pixels. We then ask the user to add one labelled pixel at a time to interact with the segmentation algorithm and improve the segmentation result, if required. Figure 23 shows one such sequence of user interactions, where the user added labelled pixels to segment the left ventricle. The pixels marked by the green dots were marked as points lying inside the ventricle, whereas the points marked with the red dots were marked as points lying outside the ventricle. The random walker generates unconstrained, arbitrary segmentation results based on user interactions. Thus, we need several precisely placed user-labels to generate the final segmentation of the left ventricle as shown in Figures 23 (a-e). In Figure 24, we show the final segmentation of the left and right ventricles using the random walker segmentation algorithm after several user interactions.

Figure 25(a) shows the final segmentation of the left ventricle in the cardiac image shown in Figure 15(a) with the proposed *active geodesic* model. A single *repeller* placed by the user is sufficient to segment out the left ventricle. Since, the final segmentation tries to optimize a region-based energy with an edge-based constraint, it overcomes the minima that hampers the Chan-Vese segmentation model, GOGAC model and the linear combination model. Further, the global edge-constraint ensures that our model generates meaningful segmentation results rather than arbitrary intermediate results generated by other user interactive segmentation algorithms (such as the random walker algorithm). Thus, we can achieve desired segmentation with fewer user interactions (with a single *repeller* in the case with left ventricle). Similarly, Figure 25(b) shows the final segmentation of the right


(a) (b) (c)

(d) (e)

Figure 23: Left Ventricle segmentation with random walker algorithm: The segmentation of the left ventricle with user-interactive Random Walker Segmentation Algorithm. Green dots - User-labelled pixels belonging to the interior of the ventricle and Red dots - User-labelled pixels belonging to the interior of the ventricle.

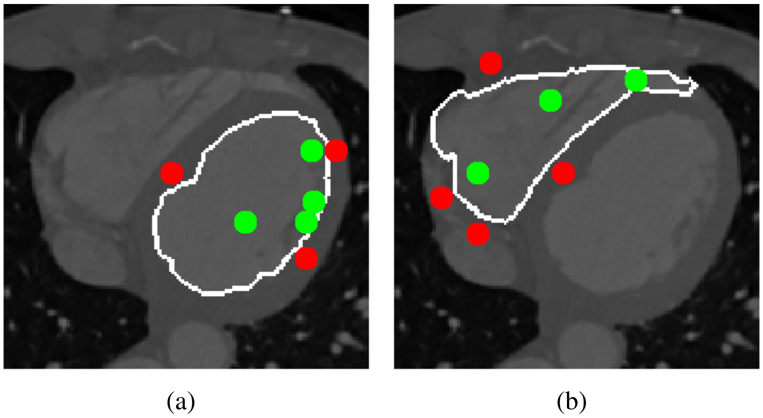
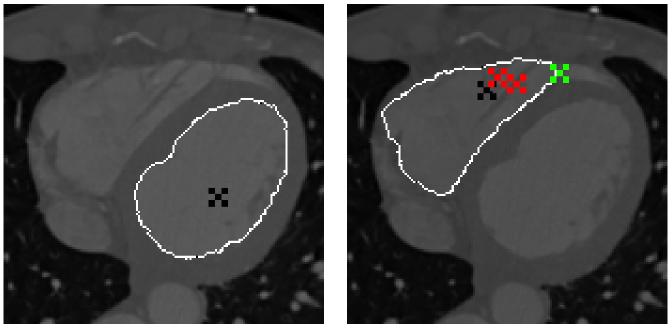

(a) (b)

Figure 24: Random walker based segmentation of both ventricles: Final Left and Right Ventricle Segmentation using random walker algorithm after several user interactions. Green dots - User-labelled pixels belonging to the interior of the ventricle and Red dots - User-labelled pixels belonging to the exterior of the ventricle.

ventricle after modifying the metric by placing a few *attractors* and *repellers*.



(a) (b)

Figure 25: The initial *repeller* is marked by a black 'X'.(a) Left Ventricle segmentation. Desired segmentation was achieved in 19 iterations (b) Right Ventricle segmentation after the user has placed a few *attractors* (marked by green 'X's) and *repellers* (marked by red 'X's). Desired segmentation was achieved after 27 iterations.

In Figure 26, we compare the segmentation of the two nuclei in a cell image with multiple nuclei. Again we see that using the *active geodesic* model segments both nuclei with a few user interactions, where as the other techniques fail to segment the nuclei. The random walker algorithm segments the nuclei with reasonable accuracy after placing several labelled pixels in the interior and exterior regions of the two nuclei. With the edge-based GOGAC approach, the final segmentation depends on the initialization and the orientation of the cut (Figures 27(b,d)). Since, it is not possible to interact with the algorithm to improve segmentation, we fail to segment one of the nuclei.

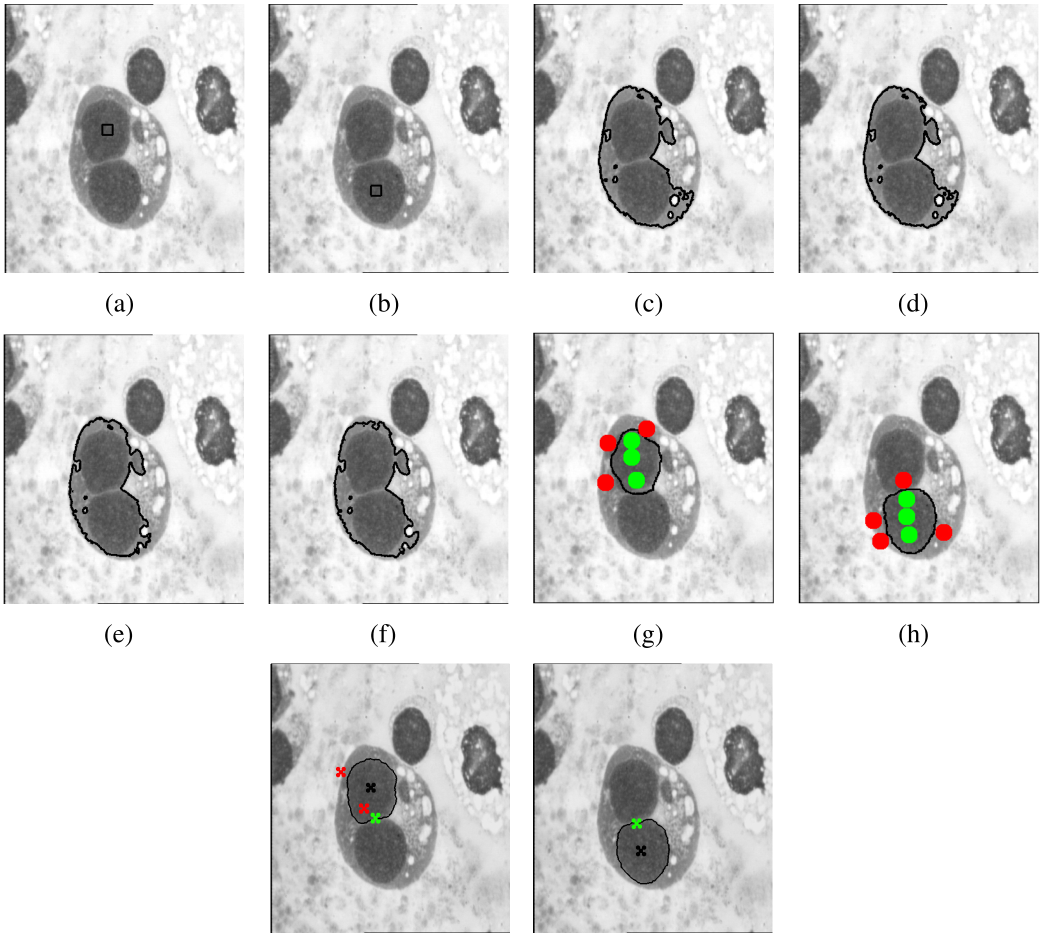
In Figures 27(a,c,e-g), we present cell segmentation results. We can see that the edge-based GOGAC approach fails due to the presence of several strong edges within the cell. The random walker algorithm also fails to capture the cell edges that lie in regions with similar intensity distribution along both sides of the edge. Using our approach, the user can accurately segment the cell by placing a few additional *attractors* and *repellers* (Figures 27(a,c,e)).

Figures 27(a,c,e), also illustrate how we can segment the cell with three different initializations. Note that the initial *repeller* in Figures 27(a,c) match the initial *repeller* for nuclei segmentation in Figures 26 (i,j), respectively.

Finally, in Figures 28, 29 and 30 we compare the segmentation results on hip bone image. Due to weak edges and low-contrast separating the interior regions of the bone from the exterior regions, this image poses an extremely challenging scenario for image segmentation algorithms. We see that the traditional models, which are not interactive fail to segment the bones in this image. Where as, both the random walker and *active geodesic* approaches segment the bones in the image after a few user interactions.

3.6 Conclusion

We have presented a novel *active geodesic* model, which constrains an evolving active contour to continually be a geodesic with respect to an edge-based metric throughout the evolution process. The edge optimality constraint reduced the infinite dimension region-based optimization problem into a finite dimension problem by reducing the search space. Further, using minimal path technique to generate geodesics during the "active" evolution of the contour ensured that the edges captured by the curve corresponds to edges of a global minimizer rather than the unwanted local minima. Minimizing a region-based energy subject to this edge-based constraint yields closed geodesics that exhibit both local and global behaviors rather than being compromises achieved by weighted combination of region- and edge-based energies. We showed that *active geodesics* generated by our model are also naturally responsive to intuitive user interactions. We used this fact to develop an interactive segmentation algorithm. We also showed results illustrating the benefits of our approach over existing segmentation algorithms on various medical images.

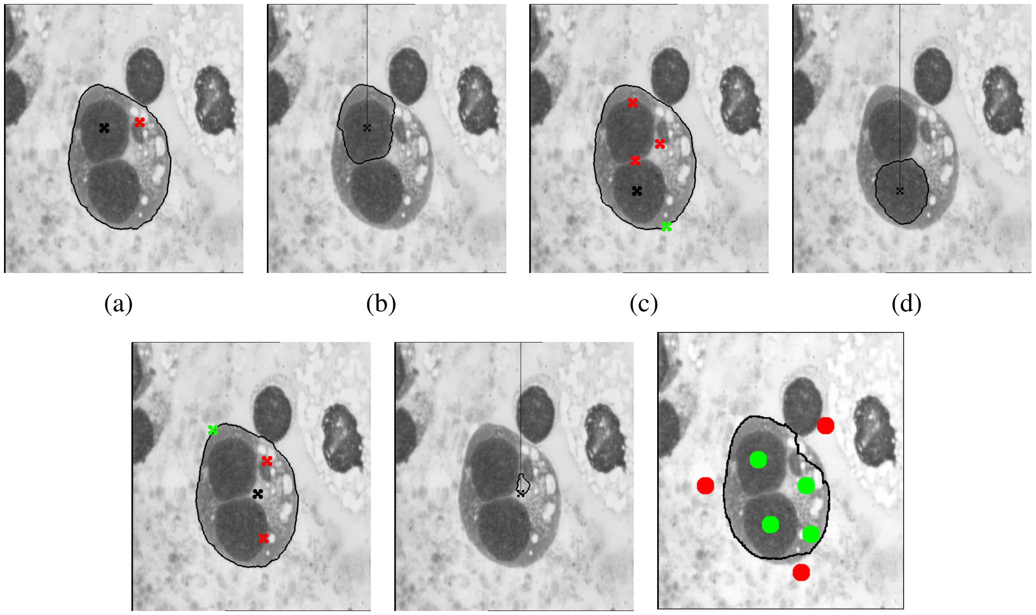


(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j)

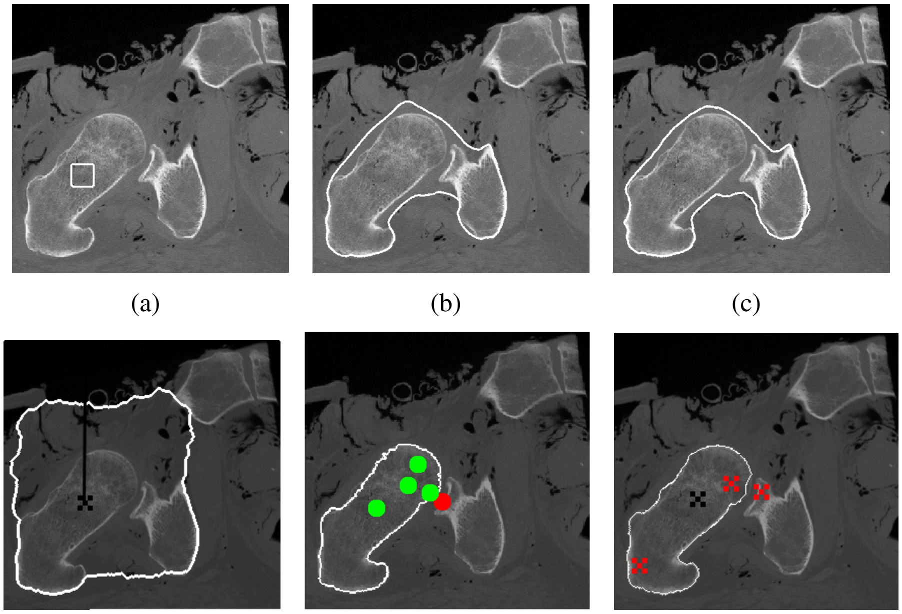
Figure 26: Nuclei segmentation: (a,b) Initialization for segmentation of the two nuclei with Chan-Vese model and linear combination model. (c,d) Final region-based Chan-Vese segmentation. (e,f) Final segmentation optimizing the combination of region- and edge-based energies. (g,h) Segmentation with random walker algorithm. Green dots - User-labelled pixels belonging to the interior of the nuclei and Red dots - User-labelled pixels belonging to the exterior of the nuclei. (i,j) *Active geodesic* based segmentation: *Attractors* - Green 'X's and *Repellers* - Red 'X's. Desired segmentation was achieved after 49 and 21 iterations, respectively



(a) (b) (c) (d)

(e) (f) (g)

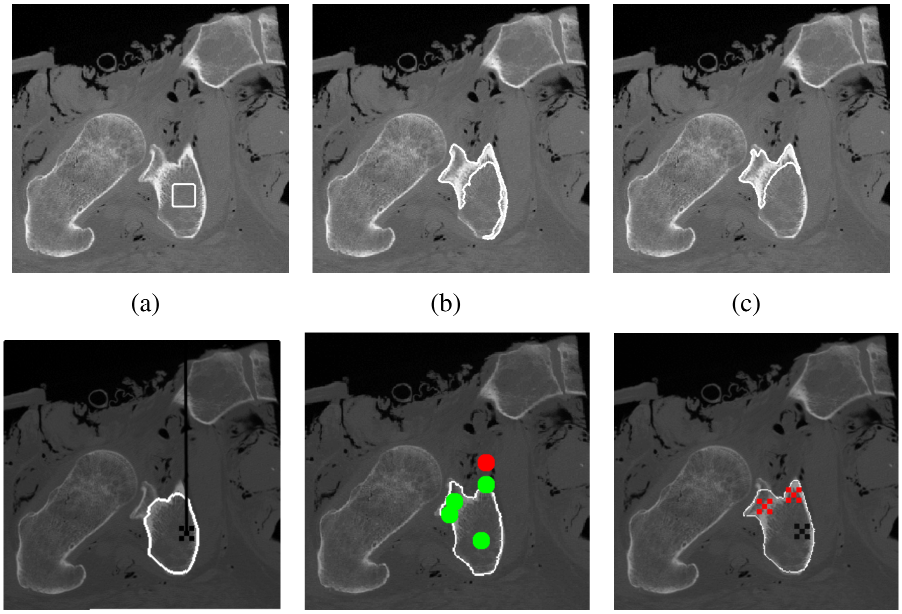
Figure 27: Cell segmentation:(a,c,e) *Active geodesic* based segmentation: The initial *repeller* is marked by a black 'X'. Subsequent *attractors* are marked by green 'X's and *repellers* are marked by red 'X's. Desired segmentation was achieved in 22, 34 and 40 iterations respectively. (b,d,f) Cell segmentation results using edge-based GOGAC approach. (g) Segmentation with random walker algorithm. Green dots - User-labelled pixels belonging to the interior of the cell and Red dots - User-labelled pixels belonging to the exterior of the cell.



(a) (b) (c)

(d) (e) (f)

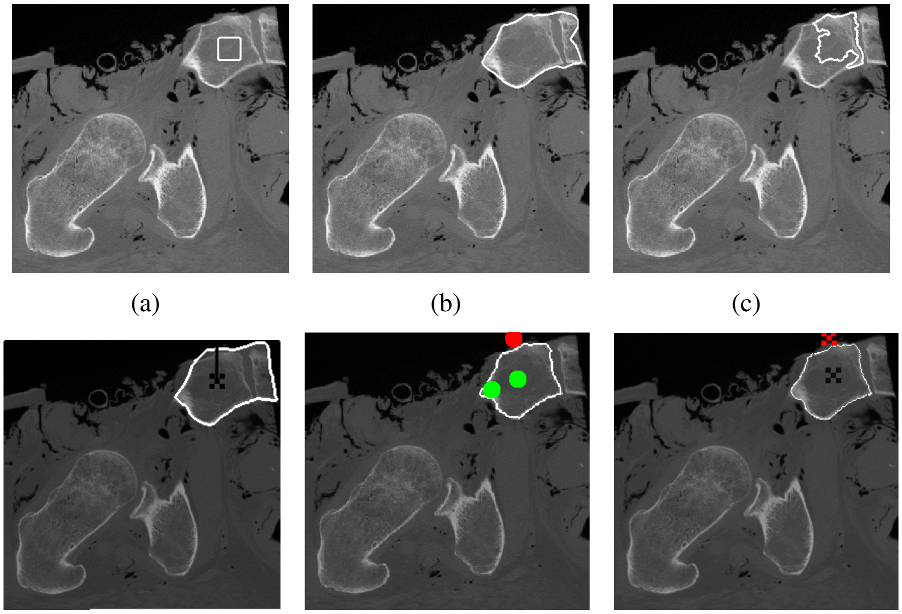
Figure 28: Hip bone segmentation: (a) Initialization. (b) Final region-based Chan-Vese segmentation. (c) Final segmentation optimizing the linear combination of region- and edge-based energies. (d) GOGAC-based segmentation. (e) Random walker algorithm based segmentation. Green dots - User-labelled pixels belonging to the interior of the bone and Red dots - User-labelled pixels belonging to the exterior of the bone. (f) *Active geodesic* based segmentation. *Attractors* - Green 'X's and *Repellers* - Red 'X's. Desired segmentation was achieved after 41 iterations.



(a) (b) (c)

(d) (e) (f)

Figure 29: Hip bone segmentation: (a) Initialization. (b) Final region-based Chan-Vese segmentation. (c) Final segmentation optimizing the linear combination of region- and edge-based energies. (d) GOGAC-based segmentation. (e) Random walker algorithm based segmentation. Green dots - User-labelled pixels belonging to the interior of the bone and Red dots - User-labelled pixels belonging to the exterior of the bone. (f) *Active geodesic* based segmentation. *Attractors* - Green 'X's and *Repellers* - Red 'X's. Desired segmentation was achieved after 17 iterations.



(a) (b) (c)

(d) (e) (f)

Figure 30: Hip bone segmentation: (a) Initialization. (b) Final region-based Chan-Vese segmentation. (c) Final segmentation optimizing the linear combination of region- and edge-based energies. (d) GOGAC-based segmentation. (e) Random walker algorithm based segmentation. Green dots - User-labelled pixels belonging to the interior of the bone and Red dots - User-labelled pixels belonging to the exterior of the bone. (f) *Active geodesic* based segmentation. *Attractors* - Green 'X's and *Repellers* - Red 'X's. Desired segmentation was achieved after 7 iterations.

CHAPTER IV

LOCALIZED PCA-BASED SEGMENTATION

In this chapter, we focus our attention on principal component analysis (PCA) based shape prior implementation for image segmentation. We use a parametric model to implicitly represent the segmenting curve as a combination of shape priors obtained by representing each training shape as a signed distance function. We define an objective region-based energy functional on the curve to optimize the parameters representing the curve. In the second part of this chapter, we introduce localized shape priors, which target (local) regions within the global shape to selectively focus the PCA efforts in certain region. This maximizes the utility of each principal shape prior. The training data for this approach consists of auxiliary (target) masks along with training shapes. The masks indicate various regions of the shape exhibiting highly correlated local variations, which may be rather independent of the variations in the distant parts of the global shape. Thus, in a sense, we are clustering the variations exhibited in the training data set. Our model fits segmenting curves locally within various target regions (divisions) in an image and then combines these locally accurate segmentation curves to obtain a single hybrid (global) segmentation. The resulting algorithm thus provides a globally accurate solution, which retain local variations in shape.

4.1 Introduction

Shape priors are widely used for automatic image segmentation, especially in cases where some prior knowledge about object of interest in the given environment is known. Shape priors are also very useful in scenarios where images have inherent noise, low contrast, missing or diffused edges. Such images are very common in medical imaging applications.

Prior shape knowledge was first used in an explicit parametrized active contour model for image segmentation by Cootes *et al.* in [34]. They called the segmentation model

"Active Shape Model". Subsequently, Chen *et al.* [27] used an average shape model to incorporate shape information in geometric active contours. Leventon *et al.* [61] used the level set framework to restrict the flow of active contours using shape information. Level set based shape priors and segmentation was later developed in [78, 92]. These models form the basis for various shape based variational segmentation models.

Methods to improve prior based segmentation by selectively preferring certain shape priors/objects over others have been discussed by Cremers *et al.* [37] and Rousson *et al.* [79]. The dynamic labeling approach developed by Cremers *et al.* [37], uses dynamic labeling to enforce known shapes to minimize the Mumford-Shah [68] energy functional. This technique is effective in segmenting objects severely corrupted by occlusion when a similar object is present in the training dataset. Recently, authors in [38, 39] have used kernel-based methods to utilize PCA effectively for image segmentation. Davatzikos *et al.* [41] showed that using wavelets in a Hierarchical Active Shape Model framework can capture certain local variations. Recently authors in [103] developed an explicit ASM-based scheme that generates independent partitions and uses PCA strictly local to these partitions and authors in [4] use global PCA with weighted local fitting. All the above-mentioned approaches treat the entire shape as a single global entity.

We begin this chapter with a detailed discussion of the traditional shape prior based curve evolution model in a level set framework. Before we learn shape variations from a given training set, we align the training images with respect to scale and orientation (Section 4.2), so that we use our resources in capturing only the shape variations and not the pose variations. In Section 4.3.1, we describe the process of representing these training shapes as level sets, and describe the process of learning the shape variations using PCA. We also discuss the parameter optimization required for fitting (segmentation) in Section 4.3.2. Later, in section 4.4, we develop localized PCA based shape priors in a level set framework and discuss localized parameter optimization to generate locally accurate segmentation curves. We describe the level set based framework used to combine these locally accurate segmentation curves in Section 4.5. Finally, in Section 4.6, we present an application of our approach to Myocardial segmentation in Cardiac images. The work in this chapter has been presented in a conference paper [5].

4.2 Binary Shape Alignment

The various images present in the training dataset vary in shape, size and orientation (Figure 31). If we use PCA on such training data, the variations in size and orientation will mask the shape variations; thereby preventing PCA from capturing shape variations effectively. To concentrate the efforts of PCA in capturing just the shape variations, we first align images in our training dataset with respect to scale and orientation. This ensures that the shape priors capture only the shape variations and not the pose variations.

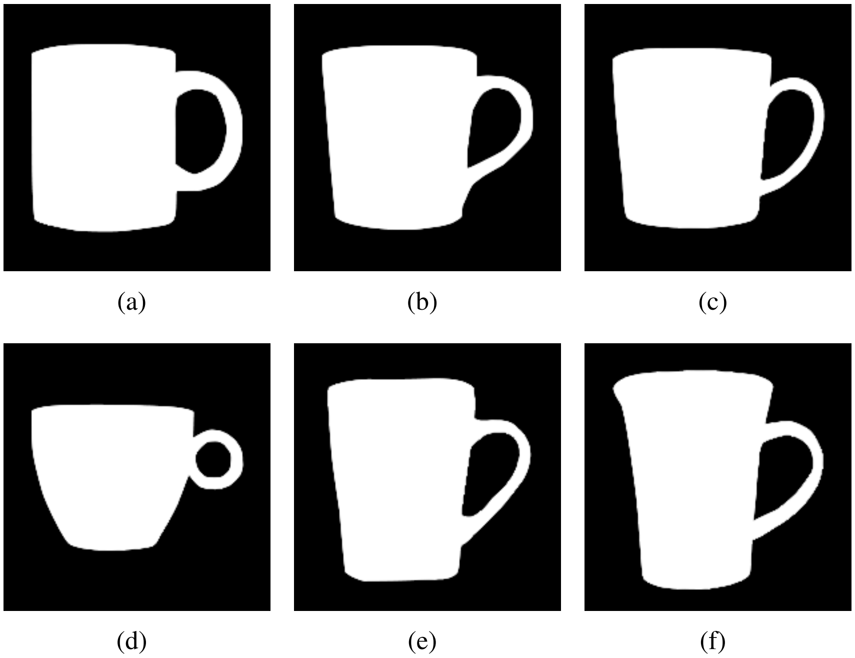


Figure 31: Training images.

Consider training set with n binary images $\{I_1, I_2, ..., I_n\}$ with pose parameters $\{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$. For 2D images we define a pose parameter vector $\mathbf{p}$, such that $\mathbf{p} = [a, b, h, \theta]$, where a, b, h and θ correspond to x-, y- translation, scale and rotation, respectively. The transformed image of I, denoted by $\tilde{I}$, for a given pose parameter ($\mathbf{p}$) is obtained using a transformation matrix $T(\mathbf{p})$. This transformation matrix $T(\mathbf{p})$ maps co-ordinates $(x, y) \epsilon \mathbb{R}^2$ to $(\tilde{x}, \tilde{y}) \epsilon \mathbb{R}^2$. The transformation matrix is a product of three matrices: a transition matrix $M(a, b)$, a scaling matrix $H(h)$ and an in-plane rotation matrix $R(\theta)$.

$$\tilde{I} = T(\mathbf{p})I,$$

where,

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} = \mathrm{T}(\mathbf{p}) \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}.$$

Transformation matrix $T(\mathbf{p}) = MHR$, where,

$$M = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} h & 0 & 0 \\ 0 & h & 0 \\ 0 & 0 & 1 \end{bmatrix}, R(\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

As suggested by Tsai $et\ al.$ [92], we can jointly align n images in our training set by minimizing the following energy functional with respect to the pose parameters:

$$E_{align} = \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \left\{ \frac{\int\int_{\Omega} (\widetilde{I}^i - \widetilde{I}^j)^2 dA}{\int\int_{\Omega} (\widetilde{I}^i + \widetilde{I}^j)^2 dA} \right\}, \tag{57}$$

where Ω is the image domain and dA is unit area.

An example of the overlay of training images before and after alignment are shown in Figures 32(a) and 32(b), respectively.

(a) (b)

Figure 32: Binary Shape Alignment: (a) Shape overlay before alignment (b) Shape overlay after alignment.

4.3 Shape Priors

In this section, we describe the traditional approach for image segmentation using global shape priors in a level set framework. This approach treats the entire training shape as a single global entity and uses PCA to learn the shape variations in the given training data set.

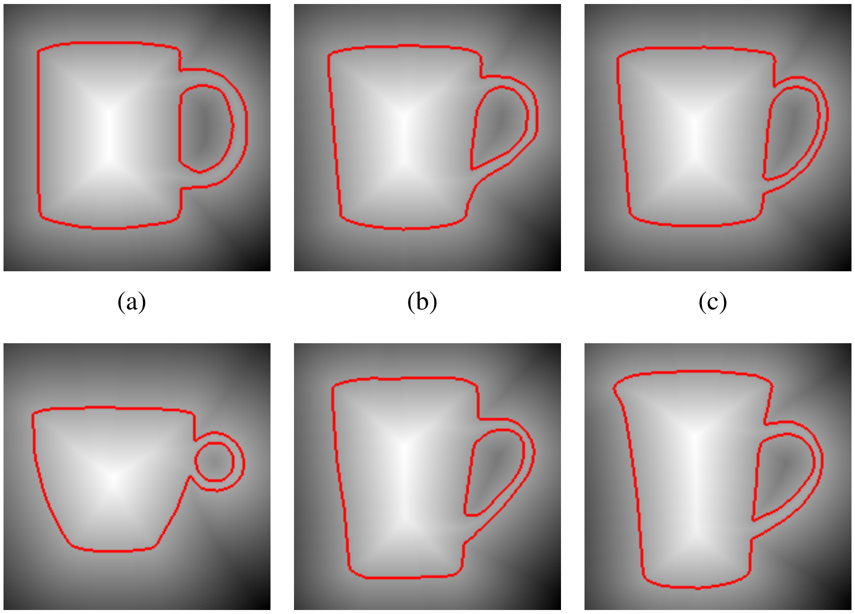
4.3.1 Global PCA

We use the level set approach introduced by Osher and Sethian [71] to represent training shapes. We represent training shapes using signed distance functions $\{\psi_1, \psi_2, ...\psi_n\}$, where the shape boundary is embedded as the zero level set. Negative distances are assigned to regions inside the boundary, and positive distances to regions outside the boundary. Figure 33 shows the level set representation of the training shapes given in Figure 31. Taking the average of the n signed distance functions we get the mean level set (Figure 34) for the training images,

$$\overline{\psi} = \frac{1}{n}\sum_{i=1}^{n}\psi_i.$$

We extract the shape variability in each training image by subtracting the mean signed distance function $\overline{\psi}$ from each signed distance function, forming n mean offset functions $\{\hat{\psi}_1, \hat{\psi}_2, ...\hat{\psi}_n\}$.

$$\hat{\psi}_i = \overline{\psi} - \psi_i.$$



(a) (b) (c)

(d) (e) (f)

Figure 33: Training shapes represented as signed distance functions.

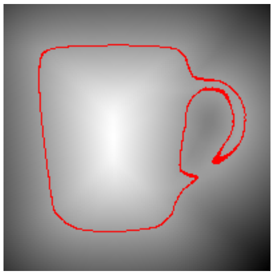


Figure 34: Mean shape generated using all training images.

We then define a shape variability matrix S, given by

$$S = [\hat{\Psi}_1 \quad \hat{\Psi}_2 \quad ... \quad \hat{\Psi}_n], \tag{58}$$

where the samples of $\hat{\psi}_i$ are stacked lexicographically to form each column $\hat{\Psi}_i$.

As suggested in [1, 92], we take the eigenvalue decomposition of $(1/n)S^T S$, which gives us the eigenshapes (principal modes), Ψ_i's. For $S = \{\hat{\psi}_1, \hat{\psi}_2, ...\hat{\psi}_n\}$, we get a maximum of n different eigenshapes $\{\Psi_1, \Psi_2, ...\Psi_n\}$. A new level set function can be expressed as a linear combination of these eigenshapes. For $k \leq n$,

$$\tilde{\psi}(\boldsymbol{\alpha}) = \overline{\psi} + \sum_{i=1}^{k} \alpha_i \Psi_i, \tag{59}$$

where $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, ..., \alpha_k\}$ are the weights associated with the k eigenshapes. We again use the zero level set of $\tilde{\psi}$ as the representation of our shape. Thus, by varying $\boldsymbol{\alpha}$, we vary $\tilde{\psi}$, which indirectly changes the shape. The value of k must be chosen large enough to capture the prominent shape variations in the given training dataset. But if the value of k is too large, the model will capture some intricate details that are specific to a particular training shape in the dataset. In all examples presented in this chapter, we choose the value of k empirically.

The segmentation also needs to accommodate variations in the pose parameters along with the variations in shape. Thus we include the pose parameters in the representation of the level set function $\tilde{\psi}$ in (59).

$$\tilde{\psi}(\boldsymbol{\alpha}, \mathbf{p}) = \overline{\psi}(\mathbf{p}) + \sum_{i=1}^{k} \alpha_i \Psi_i(\mathbf{p}). \tag{60}$$

4.3.2 Parameter Optimization via Gradient Descent

We need to choose a geometric active contour energy model to segment this region. We use an active contour model, which minimizes a region-based energy. Other forms of energy may also be used effectively with our model. Consider a general class of region-based energy

$$E(\tilde{\psi}) = \int_{R^{in}} f_{in}(x,y) dA + \int_{R^{out}} f_{out}(x,y) dA, \tag{61}$$

where $R^{in} = \{(x,y) \in \mathbb{R}^2 : \tilde{\psi}(x,y) < 0\}$ and $R^{out} = \{(x,y) \in \mathbb{R}^2 : \tilde{\psi}(x,y) \geq 0\}$.

We employ gradient descent on $E(\tilde{\psi})$ with respect to the pose parameters $\mathbf{p}$ and shape parameters $\boldsymbol{\alpha}$ for the evolution of $\tilde{\psi}$. For concise notations, we denote Θ as a collective representation of the pose and shape parameters. We denote the gradient of $E(\tilde{\psi})$, with respect to a given parameter Θ, by $\nabla_\Theta E$. The update equation for parameter Θ is given by

$$\Theta^{i+1} = \Theta^i - \nabla_\Theta E \, dt.$$

We now denote the zero level set of $\tilde{\psi}$ by $C_{\tilde{\psi}}$. The evolution of parameters Θ results in the evolution of $\tilde{\psi}$ and causes an implicit evolution of $C_{\tilde{\psi}}$. To compute $\nabla_\Theta E$, we need a line integral on curve C, which is formed by the shared boundary of regions R^{in} and R^{out}. We can represent $\nabla_\Theta E$ as the line integral

$$\nabla_\Theta E = \int\limits_C f_{in} \nabla_\Theta C \, \vec{N}_{\tilde{\psi}} ds + \int\limits_C f_{out} \nabla_\Theta C(-\vec{N}_{\tilde{\psi}}) ds, \tag{62}$$

where s is the arc length parameter of the curve and $\vec{N}_{\tilde{\psi}}$ is the outward normal of the zero level sets of $\tilde{\psi}$.

Since the zero level set of $\tilde{\psi}$ is a function of C and Θ, we have

$$\tilde{\psi}\,(C(\Theta, s), \Theta) = 0. \tag{63}$$

Taking gradient of (63) w.r.t. Θ, we get

$$\begin{aligned}
&\nabla_\Theta \tilde{\psi} + \nabla \tilde{\psi} \cdot \nabla_\Theta C = 0 \\
&\Rightarrow \nabla_\Theta \tilde{\psi} + ||\nabla \tilde{\psi}|| \cdot \vec{N}_{\tilde{\psi}} \cdot \nabla_\Theta C = 0 \\
&\Rightarrow \nabla_\Theta C \cdot \vec{N}_{\tilde{\psi}} = \frac{-\nabla_\Theta \tilde{\psi}}{||\nabla \tilde{\psi}||}.
\end{aligned} \tag{64}$$

Substituting (64) in Equation (62), we get

$$\nabla_\Theta E = \int\limits_C \left(\frac{-\nabla_\Theta \tilde{\psi}}{||\nabla \tilde{\psi}||}\right)(f_{in} - f_{out}) ds, \tag{65}$$

where $\vec{N}_{\tilde{\psi}} = \frac{\nabla \tilde{\psi}}{||\nabla \tilde{\psi}||}$.

In all examples presented here, we use the region-based energy functional proposed by Chan and Vese [25]. The Chan-Vese energy functional is suited to segment images with bi-modal distribution. We use the following choices for functions f_{in} and f_{out}:

$$f_{in} = (I - \mu)^2, \quad f_{out} = (I - \nu)^2,$$

where μ and ν denote the mean intensity values inside the regions R_{in} and R_{out}, respectively.

4.4 Localized Shape Prior based Segmentation

In order to use localized PCA for shape priors, we first need to divide our image domain into various target regions, which cluster parts of the global shape exhibiting highly-correlated local shape variations. The shape variations in these regions may be independent of the shape variations in other parts of the global shape.

Figure 35 shows a sample set of training images and associated auxiliary masks. These training images are manually traced training shapes for the object we intend to segment, mugs in this example. The masks shown in Figure 35(h-m), separate the variations in the shape of the handle from the shape of the body of the mug. With these target regions, we use a spatially weighted (localized) PCA [85] to learn the shape variation in each target region. Thus, we focus the learning from PCA to these regions and maximize the utility of each principal component (shape prior). Since we use a spatially weighted learning and fitting procedure, we do not require very precise target masks.

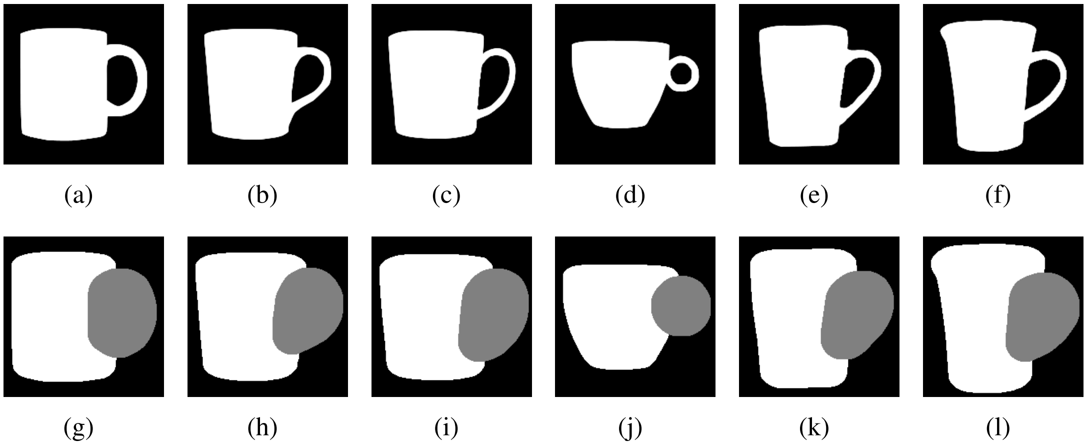


Figure 35: Training images and associated target masks: (a-f) Training images, (g-l) Automatically Generated Target Masks: The white regions mark Target Mask '1' - capturing variations in the shape of the body. Gray regions mark Target Mask '2' - capturing variations in the shape of the handle.

We represent these training shapes and target masks using the level set formulation [71]. We apply weighted PCA [85] on level sets used to represent training shapes and associated masks, to obtain a set of localized eigenshapes (shape priors) and eigenmasks (mask priors) corresponding to each target mask. We represent the segmenting curve and the mask for each (target) local segmentation as a linear combination of these localized eigenshapes and eigenmasks. Evolving these curves locally within each mask we obtain a set of locally accurate segmentation curves. We combine these curves to obtain a hybrid global segmentation curve. Finally, we evolve this hybrid curve to obtain a globally accurate segmentation. Although the local parameters affect the global shape of the segmenting curve, they evolve based on information local to the target mask.

4.4.1 Localized PCA

Since target masks correspond to regions in the original training shapes, we do not align these target masks separately. Instead, we transform the masks with the same set of pose parameters obtained in aligning the training shapes.

We use signed distance functions to represent training shapes and associated target masks [61,78,92], as described in Section 4.3.1. The zero level set depicts the shape/mask boundary, with positive distances indicating the regions inside the boundary and negative distances indicating the regions outside the boundary. Let $\{\psi_1, \psi_2, ...\psi_n\}$ denote the signed distance functions for the n shapes and $\{\phi_{T1}, \phi_{T2}, ...\phi_{Tn}\}$ denote the signed distance functions for the corresponding binary masks for a specific target T. We define the mean level set for the shapes as $\overline{\psi} = (1/n)\sum_{i=1}^{n} \psi_i$ and the mean level set for the masks of a given target T as $\overline{\phi}_T = (1/n)\sum_{i=1}^{n} \phi_{Ti}$. Now, we define an extended shape variability matrix for each target mask as explained in [1,9].

$$S_T = \Big(\begin{bmatrix} \psi_1 - \overline{\psi} \\ \phi_{T1} - \overline{\phi}_T \end{bmatrix} \begin{bmatrix} \psi_2 - \overline{\psi} \\ \phi_{T2} - \overline{\phi}_T \end{bmatrix} \cdots \begin{bmatrix} \psi_n - \overline{\psi} \\ \phi_{Tn} - \overline{\phi}_T \end{bmatrix} \Big). \tag{66}$$

In addition to the shape variability matrix, we also define a weighting matrix M_T for each target T,

$$M_T = \Big(\begin{bmatrix} g(\phi_{T1}) \\ \mathbf{1} \end{bmatrix} \begin{bmatrix} g(\phi_{T2}) \\ \mathbf{1} \end{bmatrix} \cdots \begin{bmatrix} g(\phi_{Tn}) \\ \mathbf{1} \end{bmatrix} \Big).$$

Here $g(\cdot)$ is a non-linear function, which has unit weight for the elements within the mask and for the regions outside the mask it monotonically decreases to zero as we move away from the boundary of the mask ($g(\tilde{\phi} \geq 0) = 1, 0 < g(\tilde{\phi} < 0) < 1$). $\mathbf{1}$ denotes unit weight for elements corresponding to regions inside the mask in M_T. Now, we use the spatially weighted EM framework described in [85] to estimate k principal components for shape and target masks ($k < n$). The matrix M_T gives higher emphasis to the regions within the mask, hence the shape priors will capture the local shape variations better. We denote these localized principal components for the shapes and masks as $\{\hat{\psi}_{T1}, \hat{\psi}_{T2}, ...\hat{\psi}_{Tk}\}$ and $\{\hat{\phi}_{T1}, \hat{\phi}_{T2}, ...\hat{\phi}_{Tk}\}$, respectively.

Using these k priors we formulate a new level set function $\tilde{\psi}_T$ as a linear combination of the mean level set for the shape $\overline{\psi}$ and the k principal modes (localized shape priors). Accommodating pose variations in the framework in addition to these shape variations for each given target mask T, we define a new level set $\tilde{\psi}_T$ as

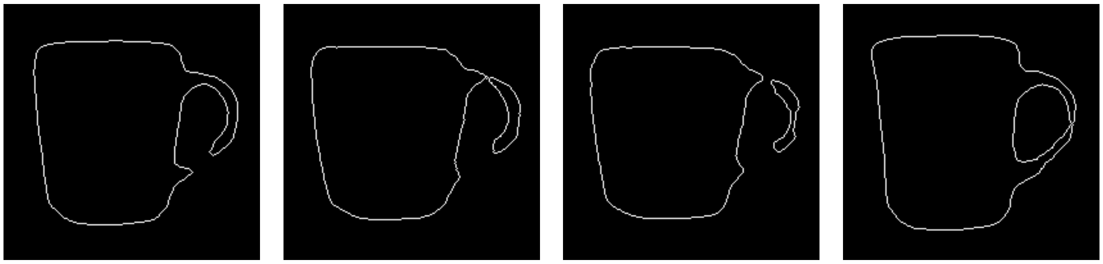
$$\tilde{\psi}_T(\boldsymbol{\alpha}, \mathbf{p_T}) = \overline{\psi}(\mathbf{p_T}) + \sum_{l=1}^{k} \alpha_{Tl}\psi_{Tl}(\mathbf{p_T}). \tag{67}$$

Similarly, we define a level set function for the mask as

$$\tilde{\phi}_T(\boldsymbol{\alpha}, \mathbf{p_T}) = \overline{\phi}(\mathbf{p_T}) + \sum_{l=1}^{k} \alpha_{Tl}\phi_{Tl}(\mathbf{p_T}).$$

Here $\{\alpha_{T1}, \alpha_{T2}, ..., \alpha_{Tl}\}$ are the weights associated with each principal mode $\{\psi_{T1}, \psi_{T2}, ..., \psi_{Tl}\}$, $\{\phi_{T1}, \phi_{T2}, ..., \phi_{Tl}\}$ and $\mathbf{p_T}$ is the pose vector. The zero level set of $\tilde{\psi}$ represents the shape boundary. By varying weights $\boldsymbol{\alpha}_T$, we can vary the shape of the segmenting curve and associated target mask implicitly. The evolution of the $\tilde{\phi}$ should correspond to evolution of $\tilde{\psi}$. Thus, we use the same set of pose and shape parameters for both these functions.

Figure 36 compares the variation caused by the principal mode of the global PCA and the localized PCA for the two target masks. Notice that the principal component corresponding to target mask 2 (Figure 36(d)) captures the variation in shape of the handle, whereas the global PCA fails to capture this variation.



(a) Mean level set $\overline{\psi}$

(b) Shape variation using principal mode of global PCA

(c) Shape variation using the principal mode for Target Mask 1

(d) Shape variation using the principal mode for Target Mask 2

Figure 36: Shape variability of global PCA and localized PCA: In (b), global PCA fails to capture shape variation of the handle, whereas (c) capture local changes in mug body while (d) captures variation in the mug handle.

4.4.2 Localized Parameter Optimization via Gradient Descent

In this section, we describe the coupling for evolution of level set $\tilde{\psi}$ with the mask $\tilde{\phi}$ for segmenting the region in the image within a given target mask. The domain under consideration will not be the entire image, but only the region within the given mask ($\tilde{\phi} \geq 0$). The definition for the region-based energy defined in (61) is now modified to

$$E(\tilde{\psi}, \tilde{\phi}) = \int_{R^{in}} f_{in}(x, y) dA + \int_{R^{out}} f_{out}(x, y) dA, \qquad (68)$$

where $R^{in} = \{(x, y) \in \mathbb{R}^2 : \tilde{\psi}(x, y) < 0, \tilde{\phi}(x, y) \geq 0\}$ and $R^{out} = \{(x, y) \in \mathbb{R}^2 : \tilde{\psi}(x, y) \geq 0, \tilde{\phi}(x, y) \geq 0\}$.

We again employ gradient descent on $E(\tilde{\psi}, \tilde{\phi})$ with respect to the shape and pose parameters. We use the same set of pose and shape parameters for the $\tilde{\psi}$ ans $\tilde{\phi}$ (Collectively denoted as Θ).

The zero level set of $\tilde{\psi}$ and $\tilde{\phi}$ are denoted by $C_{\tilde{\psi}}$ and $C_{\tilde{\phi}}$, respectively. Figure 37 shows a graphical representation of these zero level sets. The evolution of parameters Θ results in the evolution of $\tilde{\psi}$ and $\tilde{\phi}$, as well as an implicit evolution of $C_{\tilde{\psi}}$ and $C_{\tilde{\phi}}$.

To compute $\nabla_\Theta E$ in (65), we need the line integral on the curve C, which is formed by the shared boundary of regions R^{in} and R^{out}. Since we use the same set of parameters for evolution of the curve and the mask, the domain will change after each update of $\tilde{\phi}$. To stop our evolution based on energy minimization, we must compare the region based energies under the same domain. Thus, we update the mask once every few iterations.

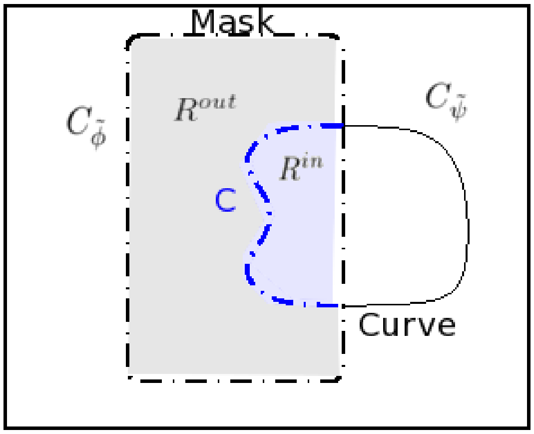


Figure 37: Domain Representation: Domain under consideration is marked by the shaded region. The region inside the mask and the curve form R^{in} and the region outside the curve but inside the mask forms R^{out}.

Figure 38 shows an example of local segmentation using two target masks. We initialize our segmentation with the mean level set shown in Figure 36(a). Since the curve evolves only on the basis of the cues within the mask, we get a reliable segmentation in the regions inside the mask.

4.5 Combined Shape Evolution

4.5.1 Initialization

The target regions are chosen such that they isolate correlated local variations. In reality, the local variations in these target regions are never completely independent from the global shape. Thus, we have to combine the local segmentation curves corresponding to each target mask to obtain a hybrid segmentation. To combine the individual local segmentation curves, we use the same level set framework that was used to generate these local

(a) Segmentation for Target Mask 1 with 2 Principal Modes (b) Segmentation for Target Mask 2 with 2 Principal Modes

Figure 38: Localized PCA based segmentation: Red curves mark the boundary of the target region. We see that we achieve accurate segmentation inside the target mask by ignoring image information from regions outside the mask.

segmentation curves.

Given N target regions, we combine the level sets $\{\tilde{\psi}_1, \tilde{\psi}_2, ...\tilde{\psi}_N\}$ into a single hybrid level set Ψ_{init}.

$$\Psi_{init} = \frac{\sum_{T=1}^{N}\{\lambda_T\ \rho(\tilde{\phi}_T) \cdot \tilde{\psi}_T(\boldsymbol{\alpha}_T, \mathbf{p}_T)\}}{\sum_{T=1}^{N}\{\lambda_T\ \rho(\tilde{\phi}_T)\}}, \tag{69}$$

where λ_T is the scalar weight associated with each target level set and $\rho(\tilde{\phi}_T) \geq 0$ is a non-linear function. We set the value of $\lambda_T = 1/N$ to obtain the initial hybrid level set. The function $\rho(\cdot)$ should be 1 in the regions inside the mask (positive values of $\tilde{\phi}_T$), and for the regions outside the mask it should monotonically decrease to zero as we move away from the mask boundary. In regions where the masks overlap, the hybrid level set will be the average of the overlapping level sets. Thus, the hybrid level set seamlessly combines the various target level sets (with a higher weight given to the regions inside each mask). Figure 39(a) shows the hybrid initialization using the local segmentation curves from Figures 38(a) and 38(b).

4.5.2 Evolution

To represent the hybrid level set Ψ, we use a single set of pose parameters $\mathbf{P}$ along with k principal modes corresponding to each target level set $\tilde{\psi}_k$. Since we have a new set of pose parameters, we update the eigenshapes and eigenmasks for each target with the corresponding pose parameters to obtain $\hat{\psi}_{Tl} = \hat{\psi}_{Tl}(p_T)$, and $\hat{\phi}_{Tl} = \hat{\phi}_{Tl}(p_T)$. Thus, (67) becomes

$$\tilde{\psi}_T(\boldsymbol{A}_T, \mathbf{P}) = \Psi_{init}(\mathbf{P}) + \sum_{l=1}^{k} A_{Tl}\hat{\psi}_{Tl}(\mathbf{P}), \tag{70}$$

where constants represented by $\boldsymbol{A}_T$ are the weights associated with the new eigenshapes. A similar equation can be derived for the update of mask $\tilde{\phi}_T$. Now, we can express the hybrid level set as

$$\Psi = \frac{\sum_{T=1}^{N}\{\lambda_T\, \rho(\tilde{\phi}_T) \cdot \tilde{\psi}_T(\boldsymbol{A}_T, \mathbf{P})\}}{\sum_{T=1}^{N}\{\lambda_T\, \rho(\tilde{\phi}_T)\}}. \tag{71}$$

This hybrid level set is a function of pose parameter set ($\mathbf{P}$), shape parameters corresponding to each target region $\{\boldsymbol{A}_1, \boldsymbol{A}_2, ..., \boldsymbol{A}_N\}$, and scalar weights $\{\lambda_1, \lambda_2, ..., \lambda_N\}$. We converge to the final segmentation by employing gradient descent (Section 4.4.2) on each of these parameters. The parameters λ_T and $\boldsymbol{A}_T$ evolve based on the cues within their respective target regions. Thus, curve evolution retains the local properties within each region, but the pose parameters are affected by the collective region inside all target masks. The pose parameters evolve based on global cues and shape priors evolve based on local cues. Thus, the final segmentation retains the local shape variations in each region, which are combined using global set of pose parameters.

Figure 39(b) shows the final evolved hybrid segmentation using 2 principal modes for each target mask. Figure 39(c) shows segmentation obtained using the conventional global PCA approach using 4 principal modes. Although we use the exact same number of principal modes to segment the image in both cases, the localized PCA approach does a very good job of capturing the shape of the body of the mug, as well as the handle. Since most of the effort from the global PCA is used in learning and fitting the shape of the body, it is unable to segment the handle correctly. Our approach concentrates the efforts of the weighted PCA to segment each target region separately, thus achieving a better global segmentation. In Figure 40, we show results on the same test image with added occlusions, pose variation and additive Gaussian noise. Results in Figure 40 suggest that our approach is robust under

(a) Initial Hybrid segmentation (b) Final (evolved) Hybrid Segmentation (c) Segmentation using standard PCA with 4 modes

Figure 39: Comparison of segmentation: The final hybrid segmentation is able to segment the mug handle and the body of the mug. Where as the global PCA based approach uses all its training resources to capture the variation in the mug body and fails to segment the handle.

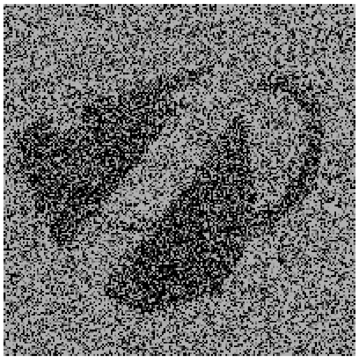
such demanding conditions, which are common to various segmentation tasks.

4.6 Application to Cardiac Image Segmentation

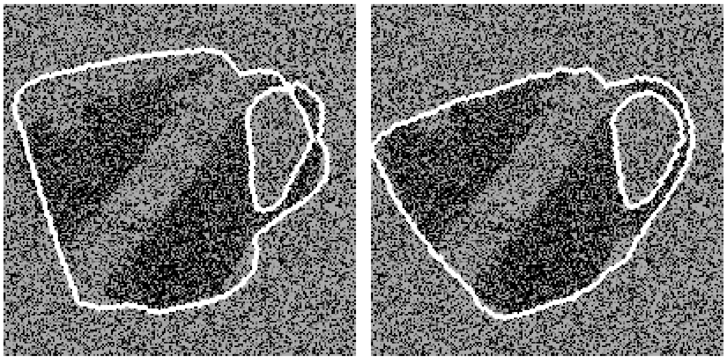
Myocardial segmentation in cardiac images is a very challenging problem, because of the low contrast separating the ventricles from the myocardium and partially missing boundaries along the ventricles. In this section, we present Myocardial Segmentation results on a cardiac dataset using our localized PCA based approach.

We used a data set of 200 2-D images from a 4-D interactive manual segmentation of a single patient's cardiac CT scan for our experiment. We used 100 of these images for training and the other 100 as the test set. Along with the manual tracings, we have a corresponding set of images marking the regions in the data with weak/missing edges, *i.e* the regions with low confidence in image data. Selectively using information from shape priors by neglecting the regions of low confidence improves segmentation . Figure 41(a-c) shows an example of one training image slice with corresponding manual segmentation and the binary mask indicating regions of low confidence in image data.

To generate target masks, we dilate the left and right ventricles obtained from manual



(a) Image in with occlusion and additive Gaussian noise



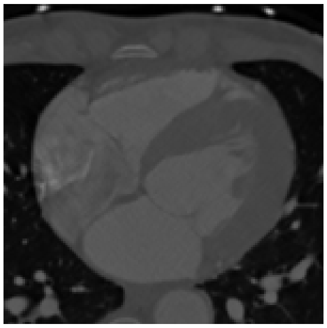
(b) Segmentation using standard PCA with 4 modes

(c) Segmentation using localized PCA with 2 modes in each target

Figure 40: Comparison of global PCA based segmentation with localized PCA based segmentation on mug image with occlusion and noise.

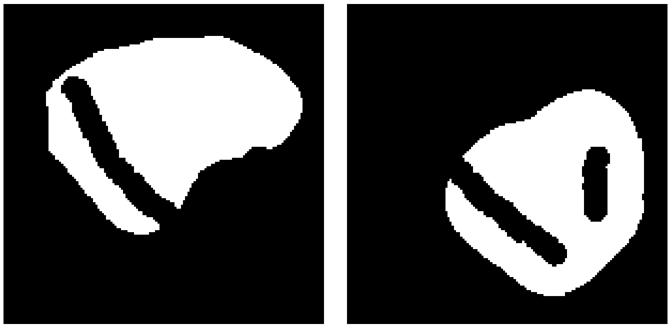
tracings. We dilate the masks enough to include some parts from the exterior regions of the epicardial boundary. From these dilated masks we exclude the regions with weak/missing edges, *i.e.* the regions with low confidence (Figure 41(d,e)). Thus, in the training phase we learn the location of these weak and missing edges along with the shape of the myocardium and the mask. Since we use a non-linear function $g(.)$ in the weighting matrix M_T, the localized PCA captures shape variations even in regions of low confidence. But, during curve fitting (evolution), we neglect all image information from these regions of low confidence.

Figure 42 compares the results of segmentation on a test image. We use 12 principal modes for each target mask, and compare our result with the global PCA approach that uses



(a) Image from training data

(b) Manual Segmentation (c) Regions of Low Confidence



(d) Mask for Right Ventricle (e) Mask for Left Ventricle

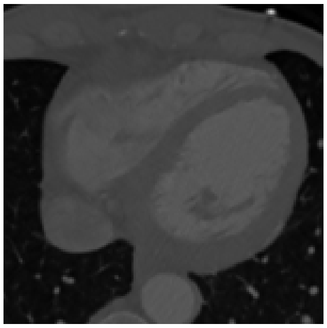
Figure 41: Myocardial segmentation: Training data and Target Masks generated from the manual segmentation.

25 principal modes. The average distance between the manual tracing and the obtained segmentation for the localized PCA based approach is 1.47 mm with a variance of 1.88mm. The average distance for the global PCA based approach is 5.21mm with a variance of 6.41mm. We see that our approach can segment the boundaries along both the ventricles and the epicardium, whereas the global PCA approach tries to fit the curve simultaneously on the endocardial and epicardial boundary, and in the process fails to achieve either. We

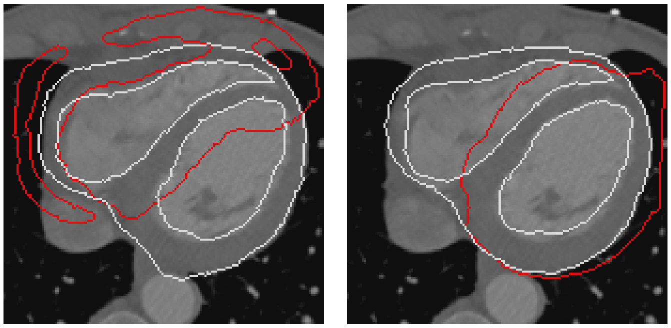
note that the masks were generated automatically from the training data, which suggests that our approach does not rely on custom designed masks. The masks need not be accurately designed for the training data; any mask that can successfully cluster the correlated local shape variations will improve segmentation. Since our approach uses fewer modes for segmentation, it can be very useful in cases where the training data set consists of limited training data.

4.7 Conclusion

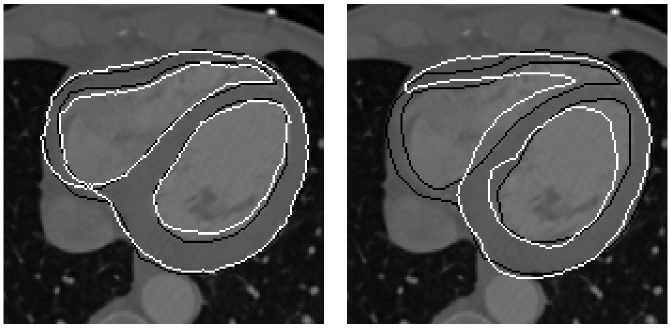
In this chapter, we have presented a variational framework that can employ localized shape priors to segment various regions in an image separately and then combine these locally accurate segmentation curves to obtain a single globally accurate segmentation. The examples presented here show that concentrating efforts of localized shape priors within certain target regions can enhance the utility of PCA as a tool. Although the examples presented here use only 2 target regions on 2-D images, it can be easily extended to accommodate multiple target regions in multi-dimensional data.



(a) Test Image Slice



(b) Segmentation with 12 Modes: Right Ventricle Mask (c) Segmentation with 12 Modes: Left Ventricle Mask



(d) Final global segmentation with 12 Modes (e) Segmentation using global PCA with 24 Modes

Figure 42: Segmentation results on cardiac image: In (b,c) the black curves mark the boundary of target region the white curve mark the obtained semi-local segmentation and in (d,e) the black curves are manual segmentation curves, which we compare with these manual tracings (white curve).

CHAPTER V

MOTION SEGMENTATION

We begin this chapter with a brief description of the traditional approaches used for motion segmentation. We then discuss the pitfalls of the traditional models and develop a new motion segmentation model, which overcomes these issues. The model proposed here treats an image sequence as a combination of a foreground layer overlapping a background layer. We use the images in the sequence to generate the layers under a smoothness assumption. We solve the motion segmentation problem by simultaneously optimizing the boundary separating the foreground and background, estimating a motion model to fit the layers to the image data and updating smooth estimates for the layers. The layering structure is defined such that the background in the occluded region is estimated under the smoothness constraint with available image information, without having to explicitly detect the occluded regions. Secondly, the layers are defined such that they utilize all pixels in the image sequence, with each pixel being considered exactly once, regardless of the image order. This ensures that all occluded regions are considered in the model and occluding regions are not considered more than once. Such modeling of the occluded region is possible because we relax the strict brightness constancy constraint and estimate a motion model by fitting the foreground and background layer to the image sequence. We therefore refer to our approach as *Relaxed Motion Segmentation*.

5.1 Introduction

In image sequences, motion of objects in the sequence becomes a very important cue for segmentation. A wide variety of motion-based computer vision applications like optical flow estimation, motion segmentation and tracking utilize motion based cues for object segmentation. A Variational methods for optical flow estimation were introduced by Horn and Schunck [48]. Lucas and Kanade [64] also introduced an optical flow estimation algorithm based on the optical flow constraint. The Horn-Schunk optical flow estimation used a global smoothness constraint in the flow field, whereas Lucas-Kanade used a local smoothness constraint to estimate the flow field. Later, various motion segmentation algorithms based on the basic principles discussed in these papers [48, 64] were developed.

Wang and Adelson [96] proposed a two-step motion segmentation approach, where they first estimate a local flow field and then cluster regions using the estimated flow field. Black and Anandan [16] used a parametric motion model to estimate the motion vectors, and then used active contours to segment the estimated flow field. Farneback [43] introduced 3-D orientation tensors along with the parametric motion model to improve motion vector estimation. Later, Paragios and Deriche [72] developed a motion-based level set segmentation algorithm.

As opposed to the previously mentioned two-step segmentation approaches, motion segmentation models described in [20, 35] use local smoothness constraint in the flow field to identify the boundary of objects moving independently in the image sequence. Cremers and Soatto developed the motion competition model [36], which couples motion estimation and boundary optimization using the brightness constancy constraint (BCC), introduced by Alvarez *et al.* [3], instead of the differential optical flow constraint [48]. Dense flow fields, with large displacement, can be estimated using the brightness constancy constraint. Motion estimation algorithm with dense flow fields have also been developed in [19, 86] using the brightness constancy constraint.

The above-mentioned algorithms assume absence of occlusion due to moving objects and the motion models are valid only in unoccluded regions. But, in reality handling occlusion is an important factor in motion segmentation algorithms. Certain approaches have been developed to identify regions in the image that are occluded [13, 49, 88] and impose certain constraints in the occluded region, while some approaches, such as [59], enforce motion field discontinuity in sub-domains for occlusion handling.

In Section 5.3, we propose an enhancement to the traditional motion segmentation model in which we treat foreground object and background of the image sequence as overlapping layers. We simultaneously estimate the layers under the smoothness assumption, the motion parameters and the object boundary using available information from two consecutive frames in the image sequence. The concept of treating images as deformable, overlapping layers was introduced in [50], and using smoothness constraint in regions inside and outside the evolving contour was introduced by Mumford and Shah [67]. Under the smoothness constraint, we define the foreground and background layers such that the background in occluded regions are estimated with available image information, without having to explicitly detect the occluded regions. With this proposed approach, we can model the occluded regions and circumvent the need to impose constraints on the evolving contour in these occluded regions.

5.2 Traditional Motion Segmentation Models

There are two major tasks associated with motion segmentation: a) estimating motion vectors, and b) detecting object boundaries. Traditionally, motion segmentation algorithms can be classified into two categories based on how they perform these two tasks. In the first approach, based on the global smoothness assumption, the optical flow field is calculated and then the flow field is segmented using an active-contour-based curve evolution model. In the second approach, an energy functional based on the brightness constancy constraint is defined. This approach relies on the local smoothness assumption to jointly optimize motion estimation and boundary detection.

5.2.1 Optical Flow-Based Motion Segmentation

For the optical flow based two-step approach, we first calculate the optical flow field, and then segment objects in the flow field using a region-based curve evolution model. We can formulate the problem of estimating optical flow as an energy minimization problem by adding a global smoothness constraint on the optical flow velocity. Combining global smoothness constraint with (6), we get the energy functional for optical flow as

$$E_{op} = \int_{\Omega} \Big((I_x \cdot u + I_y \cdot v + I_t)^2 + \alpha (||\nabla u||^2 + ||\nabla v||^2) \Big) dA, \tag{72}$$

where $\alpha > 0$ is a weighting factor. We can now solve for u and v iteratively, using the following equations:

$$\begin{aligned} u^{n+1} &= \overline{u}^n - \frac{I_x [I_x \overline{u}^n + I_y \overline{v}^n + I_t]}{\alpha^2 + {I_x}^2 + {I_y}^2}, \\ v^{n+1} &= \overline{v}^n - \frac{I_y [I_x \overline{u}^n + I_y \overline{v}^n + I_t]}{\alpha^2 + {I_x}^2 + {I_y}^2}. \end{aligned} \tag{73}$$

Here u^{n+1}, v^{n+1} are the updated estimates of u^n and v^n, and $\overline{u}^n$ and $\overline{v}^n$ are the local averages [48]. Solving (73) iteratively, until u and v converge, gives us the optical flow field.

This flow field can be interpreted as a vector-valued image, with two components at each pixel location. We can use the vector-valued curve evolution model described in [26] for curve evolution. Consider an evolving curve C in the given image domain. We represent the curve implicitly using a level set function ϕ [70], such that $C = \{(x, y) \in \mathbb{R}^2 : \phi(x, y) = 0\}$. The region inside the curve is represented by $R^{in} = \{(x, y) \in \mathbb{R}^2 : \phi(x, y) \geq 0\}$ and the region outside the curve is represented by $R^{out} = \{(x, y) \in \mathbb{R}^2 : \phi(x, y) < 0\}$. We can now define a region-based energy functional for the two-channel flow field as

$$\begin{aligned} E_{cv} = \lambda \cdot Length(C) &+ \int_{R^{in}} (u - U_{in})^2 dA + \int_{R^{in}} (v - V_{in})^2 dA \\ &+ \int_{R^{out}} (u - U_{out})^2 dA + \int_{R^{out}} (v - V_{out})^2 dA, \end{aligned} \tag{74}$$

where U_{in} and V_{in} are the average values of u and v inside the evolving curve C, U_{out} and V_{out} are the average values of u and v outside the evolving curve C and $\lambda \geq 0$ is a weighting factor.

Minimizing the energy via gradient descent separates the image domain into regions with the least flow-field variation inside and outside the curve. Evolving ϕ implicitly evolves the segmenting curve C. The gradient descent equation for the level set representing the curve can be written as

$$\frac{\partial \phi}{\partial t} = -\delta(\phi)\Big(2(U_{in}+U_{out})(u-\frac{U_{in}-U_{out}}{2})\cdot\vec{N} + 2(V_{in}+V_{out})(v-\frac{V_{in}-V_{out}}{2})\cdot\vec{N} + \gamma\cdot\vec{N} + \lambda\cdot\kappa\cdot\vec{N}\Big), \tag{75}$$

where $\vec{N}$ is the outward normal of the zero level set of ϕ, κ is the curvature of C and γ is a constant that regulates inflationary/shrinking force. As described in [25], the function δ is implemented by a smooth approximation of the delta function.

Figures 43(a) and 43(b) show two successive frames of an image sequence. In Figure 43(c), we present the optical flow field computed using (72). Because of the global smoothness constraint, the flow field does not change abruptly at object boundaries.

We start our segmentation with the initial contour shown in Figure 44(a). Minimizing the energy in (73) drives the contour to the final segmentation, shown in Figure 44(b). For the given initialization, we choose the constant γ to have a shrinking effect on the contour. This prevents the contour from getting stuck in local minima. We see that the final segmentation does not align with the object boundaries because of the global smoothness constraint.

(a) Image I_1. (b) Image I_2. (c) Optical Flow.

Figure 43: Optical flow: (a,b) Successive images from an image sequence. (c) Optical flow field overlayed on the image.

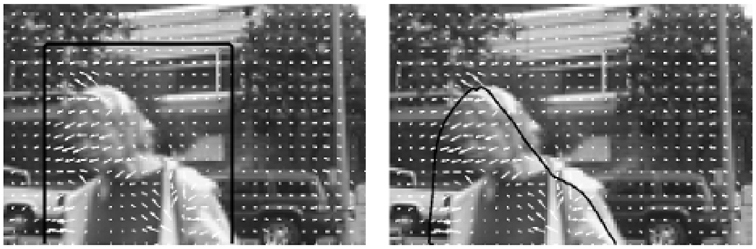

(a) Initialization. (b) Final segmentation.

Figure 44: Vector-valued segmentation on the calculated flow field

5.2.2 Joint Motion Estimation and Boundary Optimization

In this section, we describe the single-step approach for motion segmentation. In this approach, we estimate motion and optimize the object boundary simultaneously, by minimizing a single energy functional based on the brightness constancy constraint (76). We assume that the motion field is smooth (constant) locally, within regions inside and outside the curve, but not across the curve. Thus, when the curve boundary evolves to match the object boundary, we do not smooth the flow field across the discontinuity at this boundary.

This joint optimization approach combines the two tasks of estimating motion vectors and detecting object boundaries. This is achieved by minimizing a single energy function to simultaneously estimate motion vectors and detect object boundary. Optical flow and motion estimation algorithms assume that the brightness (gray value) of a moving pixel does not change with displacement. The brightness constancy constraint (BCC), based on this assumption, can be defined as

$$I_1(x,y) = I_2(x',y'), \tag{76}$$

where $I_1 : \Omega \rightarrow \mathbb{R}$ and $I_2 : \Omega \rightarrow \mathbb{R}$ are images defined over a domain $\Omega \subset \mathbb{R}^2$ at two different time instants t_1 and t_2, and x' and y' are transformed co-ordinates with respect to a given group action model g_i. Group action model g_i is defined such that $[x' \; y']^T = g_i(\; [x \;\; y]^T)$, *i.e.* $X' = g_i(X)$, where $X = [x \;\; y]^T$. The group action model g_i is defined in terms of affine or non-affine parameters p_i. For the examples considered here, a parameter set p_i

consisting of translation along the X- and Y-axis was sufficient for motion segmentation.

Equation (76) is ill-posed because it has two unknowns. Thus, we need additional constraints to solve this equation. As mentioned earlier, Horn and Schunck [48] introduced a global smoothness constraint, whereas Lucas and Kanade [64] used a local smoothness assumption to solve this problem. Motion segmentation models assume that the motion field is locally smooth within the foreground and the background. Based on this local smoothness assumption, the energy functional for a typical motion segmentation model can be defined as

$$E = \lambda \cdot \text{Length}(C) + \int_R [I_1 - I_2 \circ g_{in}]^2 d\Omega + \int_{(\Omega \setminus R) \setminus O} [I_1 - I_2 \circ g_{out}]^2 d\Omega, \tag{77}$$

where R represents the region inside the segmenting curve C, $I_1 : \Omega \to \mathbb{R}$ and $I_2 : \Omega \to \mathbb{R}$ are images defined over a domain $\Omega \subset \mathbb{R}^2$ at two different time instants t_1 and t_2, g_{in} and g_{out} represent the smooth motion field (group actions for each region) in R and $\Omega \setminus R$, respectively, O denotes the occluded regions according to the current motion model and object boundary and $\lambda \geq 0$ is a weighting factor. The symbol '$\circ$' denotes that the given image is composed with the respective motion vector field (group action model). We represent the curve implicitly using a level set function ϕ [70], such that $C = \{(x, y) \in \mathbb{R}^2 : \phi(x, y) = 0\}$. The region inside the curve is now represented by $R = \{(x, y) \in \mathbb{R}^2 : \phi(x, y) \geq 0\}$, and the region outside the curve is $\Omega \setminus R = \{(x, y) \in \mathbb{R}^2 : \phi(x, y) < 0\}$.

Notice that the motion field is smooth (constant) locally, in the regions inside and outside the given segmenting curve, but not across the curve. When the curve boundary evolves to match the object boundary, the flow field maintains a discontinuity at the boundary. Since, the energy functional depends on the motion vectors and the embedding level set function ϕ, we minimize this energy by alternating evolution of the level set and the motion vectors.

5.2.2.1 *Updating motion vectors*

For a fixed ϕ, we can update the motion vectors parameters p_{in} and p_{out} by the following update equation:

$$p_i^{\,n+1} = p_i^{\,n} + \frac{\partial E}{\partial p_i} \cdot dp_i, \tag{78}$$

where $i = \{\text{in}, \text{out}\}$.

$$\begin{aligned} \frac{\partial E}{\partial p_{in}} &= -\int_R [I_1 - I_2 \circ g_{in}] \cdot \nabla I_2 \circ g_{in}, \\ \frac{\partial E}{\partial p_{out}} &= -\int_{(\Omega \setminus R)\setminus O} [I_1 - I_2 \circ g_{out}] \cdot \nabla I_2 \circ g_{out}. \end{aligned} \tag{79}$$

5.2.2.2 *Level set evolution*

Conversely, for fixed motion vectors, the gradient descent equation for the level set ϕ can be written as

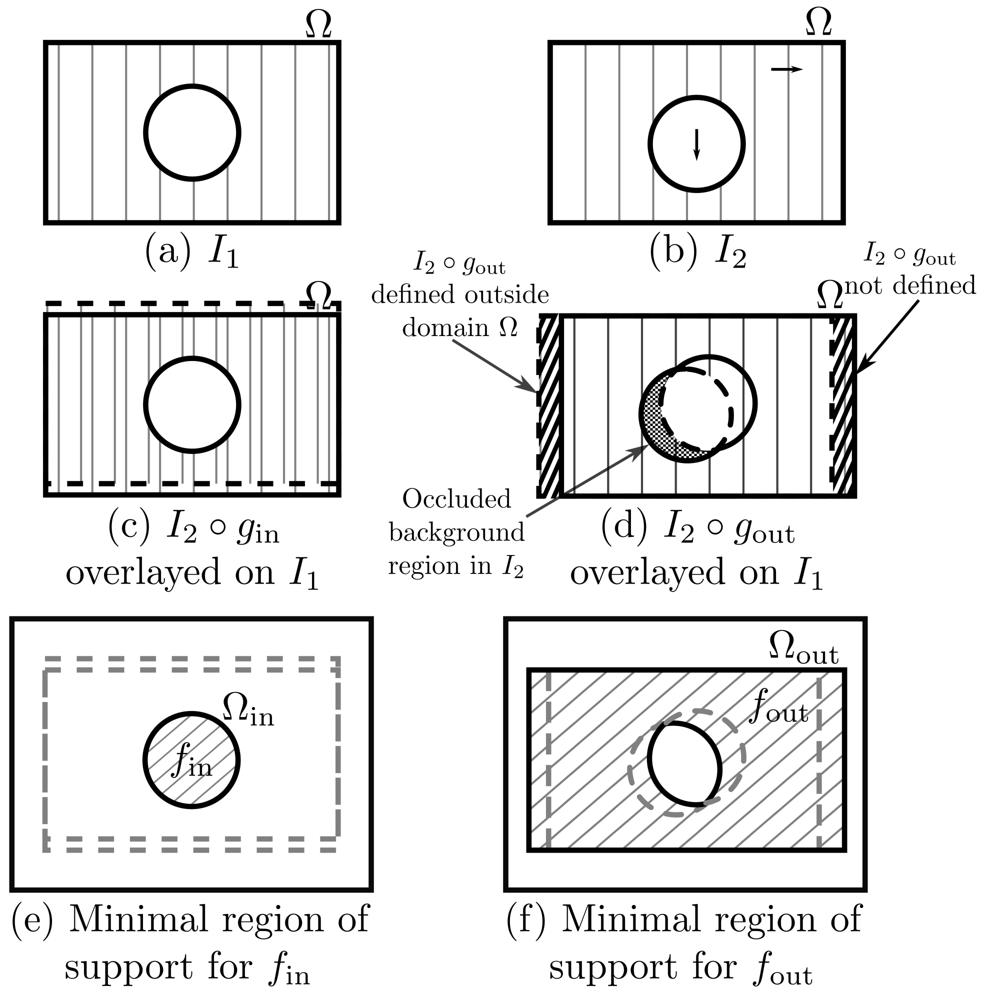
$$\frac{\partial \phi}{\partial t} = -\delta(\phi)\Big([(I_1 - I_2 \circ g_{in})^2 - (I_1 - I_2 \circ g_{out})^2] \cdot \vec{N} + \gamma \cdot \vec{N} + \lambda \cdot \kappa \cdot \vec{N}\Big), \tag{80}$$

where $\vec{N}$ is the outward normal of the zero level set of ϕ, κ is the curvature of C, γ is a constant and $\delta(\phi)$ is smooth approximation of a delta function.

5.2.3 Occlusion Problems in Traditional Motion Segmentation Models

Consider two sample images, I_1 and I_2, shown in the illustration in Figure 45 (a) and Figure 45 (b). Here we represent the foreground object by a circle and it occludes parts of the background, which is represented by vertical lines. The group action for the foreground can be denoted by a translation in the vertical direction and the group action for the background can be denoted by translation in the horizontal direction.

Figure 45 (c) shows the overlay of I_1 and $I_2 \circ g_{in}$ for the integral in region R and Figure 45 (d) shows the overlay of I_1 and $I_2 \circ g_{out}$ for the integral in region $\Omega \setminus R$ in equation (77). The domain for the difference term $I_1 - I_2 \circ g_{in}$ is well defined for region R. But, $I_1 - I_2 \circ g_{out}$ (for the background) is not consistently defined over the entire domain $(\Omega \setminus R)$ because of the following:



(a) I_1 (b) I_2

(c) $I_2 \circ g_{\text{in}}$ overlayed on I_1 (d) $I_2 \circ g_{\text{out}}$ overlayed on I_1

(e) Minimal region of support for f_{in} (f) Minimal region of support for f_{out}

Figure 45: Illustration of the occlusion problem. (a) I_1. (b) I_2. (c) Overlay of I_1 and $I_2 \circ g_{in}$. (d) Overlay of I_1 and $I_2 \circ g_{out}$. (e,f) Minimal region of support for f_{in} and f_{out}.

***a) Occlusion*:**

Since motion in the foreground and the background are different, there are regions in the background of I_2 that are occluded by the foreground, but visible in I_1. The occluded region is highlighted in Figure 45 (d).

***b) Image domain boundary inconsistency*:**

Secondly, there are regions in image I_2 that move outside the domain Ω when operated with the motion model. These pixels are not accounted in the motion segmentation model. Moreover, there are regions in domain Ω, where $I_2 \circ g_{out}$ is not defined. These regions are highlighted in the overlay image in Figure 45 (d).

Since the definition of traditional motion segmentation model does not account for these problems, it may lead to certain undesirable segmentation results in which the contour gets stuck in undesired minima. Figure 46 shows an example in which the contour gets stuck in local minima using traditional motion segmentation model. Figure 46 (a) and Figure 46 (b) show images at two time instants t_1 and t_2 and Figure 46 (c) shows the initial contour. The segmentation obtained by minimizing the energy in (77) is shown in Figure 46(d). We notice that the curve gets stuck in undesired minima due to improper occlusion handling. Since the contour is not on the object boundary, the motion vector estimation is not precise, which further aggravates the occlusion problem causing the contour to get stuck in the minima. Adding a shrinking balloon force or an inflationary balloon force or increasing the weight on the curvature term to overcome the minima may not lead to the correct solution, as shown in Figure 46 (e), Figure 46 (f) and Figure 46 (g), respectively.

Figure 47 shows an example for motion segmentation on the hockey player image sequence. Due to camera motion, there is significant motion along X-axis in the background and the foreground. Thus the background behind the player is severely occluded. We see that segmentation curve gets stuck in a minima due to improper occlusion handling in the traditional motion segmentation model.

5.3 Relaxed Motion Segmentation

In this section, we develop a motion segmentation model, which can overcome image boundary inconsistency and occlusion handling problems described previously.

5.3.1 Relaxed Motion Segmentation Model

Consider images $I_1 : \Omega \rightarrow \mathbb{R}$ and $I_2 : \Omega \rightarrow \mathbb{R}$ defined over a domain $\Omega \subset \mathbb{R}^2$ at two different time instants t_1 and t_2. We define layers $f_{in} : \Omega_{in} \rightarrow \mathbb{R}$ over a domain $\Omega_{in} \subset \mathbb{R}^2$ and $f_{out} : \Omega_{out} \rightarrow \mathbb{R}$ over a domain $\Omega_{out} \subset \mathbb{R}^2$ for the foreground and background, respectively. These layers represent smooth approximates of the regions in the foreground

(a) I_1. (b) I_2. (c) Initialization.

(d) Traditional Motion Segmentation Result. (e) Traditional Motion Segmentation with shrinking force. (f) Traditional Motion Segmentation with inflationary force.

(g) Traditional Motion Segmentation with large weight on curvature term.

Figure 46: Avoiding local minima in motion segmentation: (a) I_1. (b) I_2. (c) Initial contour. (d) Final segmentation contour is stuck in an undesirable minima. (e) Final segmentation with a shrinking force. (f) Final segmentation with an inflationary force. (g) Final segmentation with a large weight on the curvature term.

and background from both images. Based on these smooth foreground and background layers we define a new energy functional for the relaxed motion segmentation model,

$$\begin{aligned} E = \beta\Big\{ & \int_{\Omega\cap R}(I_1 - f_{in})^2 d\Omega + \int_{\Omega\backslash R}(I_1 - f_{out})^2 d\Omega \\ & + \int_{\Omega\cap g_{in}(R)}(I_2 - f_{in}\circ {g_{in}}^{-1})^2 d\Omega + \int_{\Omega\backslash g_{in}(R)}(I_2 - f_{out}\circ {g_{out}}^{-1})^2 d\Omega\Big\} \\ & + (1-\beta)\Big\{\int_{\Omega_{in}} ||\nabla f_{in}||^2 d\Omega_{in} + \int_{\Omega_{out}} ||\nabla f_{out}||^2 d\Omega_{out}\Big\} + \lambda\int_C ds, \end{aligned} \tag{81}$$

(a) I_1. (b) I_2.

(c) Initialization. (d) Traditional Motion Segmentation Result.

Figure 47: Avoiding local minima in motion segmentation: (a) I_1. (b) I_2. (c) Initial contour. (d) Final segmentation contour is stuck in an undesirable minima.

where the domains Ω_{in} and Ω_{out} are defined such that,

$$\Omega_{in} \subset R \cap (\Omega \cup {g_{in}}^{-1}(\Omega)),$$

$$\Omega_{out} \subset (\Omega \setminus R) \cup \{{g_{out}}^{-1}(\Omega) \setminus ({g_{out}}^{-1} \circ g_{in})(R)\}. \tag{82}$$

The first four terms in the energy functional (81) are data fidelity terms, which ensure that the foreground and background layers are consistent with the image data. The terms penalizing $||\nabla f_{in}||^2$ and $||\nabla f_{out}||^2$ are smoothness terms for the foreground and background layers, respectively. Finally, the last term is a penalty on the length of the curve C. By varying the parameter $\beta \in [0, 1]$, we can vary the significance of the data fidelity and the smoothness terms.

In our model, we choose the minimal region of support for f_{in} and f_{out} by taking $\Omega_{in} = R \cap (\Omega \cup {g_{in}}^{-1}(\Omega))$ and $\Omega_{out} = (\Omega \setminus R) \cup \{{g_{out}}^{-1}(\Omega \setminus ({g_{out}}^{-1} \circ g_{in})(R)\}$. The minimal region of support for the foreground and background layers are non-arbitrary domains that are large enough to account for f_{in} and f_{out}, but no larger. The highlighted regions in

Figure 45 (e) and Figure 45 (f) show examples of minimal regions of support for f_{in} and f_{out}, respectively.

In equation (81), with minimal regions of support, the data fidelity terms consider each and every pixel on both images either in the background or the foreground layer. The disjoint domains of integration for the background and foreground layers on each image ensure that each pixel (on both images) is considered exactly once. There are no holes due to occluded regions in the estimated background layer and any portion of the image that moves outside the image domain when operated with the motion model is accounted for in the smooth approximation of the background layer.

Since this energy functional depends on group action models g_{in} and g_{out}, foreground and background layers f_{in} and f_{out}, and embedding level set function ϕ, we minimize the energy by alternating the evolution of the level set function, motion vectors and smooth estimates of the foreground and background layers.

5.3.1.1 Update for Foreground and Background Layers

For fixed ϕ, g_{in} and g_{out}, we update f_{in} and f_{out} by solving the following equations:

$$\Delta f_{in} = (f_{in} - I_1) \cdot \chi_{(R \cap \Omega)} + (f_{in} - I_2 \circ g_{in}) |\frac{\partial g_{in}}{\partial X}| \cdot \chi_{(R \cap {g_{in}}^{-1} \Omega)}, \tag{83}$$

$$\Delta f_{out} = (f_{out} - I_1) \cdot \chi_{(\Omega \backslash R)} + (f_{out} - I_2 \circ g_{out}) \cdot |\frac{\partial g_{out}}{\partial X}| \cdot \chi_{(\Omega \backslash (g_{in} \circ {g_{out}}^{-1}) R)}, \tag{84}$$

where χ_{R_i} is a characteristic function such that,

$$\chi_{R_i}(x) = \begin{cases} 1, & \text{if } x \in R_i, \\ 0, & \text{otherwise.} \end{cases} \tag{85}$$

5.3.1.2 *Updating motion vectors*

For fixed ϕ, f_{in} and f_{out}, we update the motion vectors g_{in} and g_{out}. The partial derivatives $\frac{\partial E}{\partial p_{in}}$ and $\frac{\partial E}{\partial p_{out}}$ are given by

$$\begin{aligned}\frac{\partial E}{\partial p_{in}} =& -2\int_{\Omega\cap R}(I_2\circ g_{in}-f_{in})\cdot(\nabla f_{in})\cdot \text{adj}\left(\frac{\partial g_{in}}{\partial X}\right)\cdot\frac{\partial g_{in}}{\partial p_{in}}d\Omega \\ &+\int_C (I_2\circ g_{in}-f_{in})^2\cdot\frac{\partial g_{in}}{\partial p_{in}}\cdot\text{adj}(\frac{\partial g_{in}}{\partial X})^T\cdot N\cdot ds,\end{aligned}$$

$$\begin{aligned}\frac{\partial E}{\partial p_{out}} =& -2\int_{{g_{out}}^{-1}(\Omega\backslash g_{in}(R))}(I_2\circ g_{out}-f_{out})\cdot(\nabla f_{out})\cdot \text{adj}\left(\frac{\partial g_{out}}{\partial X}\right)\cdot\frac{\partial g_{out}}{\partial p_{out}}d\Omega \\ &+\int_C (I_2\circ g_{in}-f_{out}\circ {g_{out}}^{-1}\circ g_{in})^2\cdot\frac{\partial g_{in}}{\partial p_{out}}\cdot\text{adj}(\frac{\partial g_{in}}{\partial X})^T\cdot N\cdot ds.\end{aligned} \tag{86}$$

Here, $\vec{N}$ is the outward normal to the curve C, and $\frac{\partial g_{in}}{\partial X}$ and $\frac{\partial g_{out}}{\partial X}$ are Jacobian matrices introduced in these equations since we switch the group action operator from the smooth estimates to the images. The complete derivation for equations in (86) are given in Appendix D.

5.3.1.3 *Level set evolution*

Finally, for fixed g_{in}, g_{out}, f_{in} and f_{out}, the gradient descent equation for the level set ϕ can be written as

$$\frac{\partial\phi}{\partial t} = -\delta(\phi)\Big([F_{in}-F_{out}]\cdot\vec{N}+\lambda\cdot\kappa\cdot\vec{N}\Big), \tag{87}$$

where,

$$F_{in} = (1-\beta)||\nabla f_{in}||^2+\beta\{[I_1-f_{in}]^2+[I_2\circ g_{in}-f_{in}]^2\cdot|\frac{\partial g_{in}}{\partial X}|\},$$

$$F_{out} = (1-\beta)||\nabla f_{out}||^2+\beta\{[I_1-f_{out}]^2+[I_2\circ g_{in}-f_{out}\circ {g_{out}}^{-1}\circ g_{in}]^2\cdot|\frac{\partial g_{in}}{\partial X}|\}, \tag{88}$$

and $\vec{N}$ is the outward normal of the zero level set of ϕ, κ is the curvature of C and $\delta(\phi)$ is a smooth approximation of the delta function.

5.3.2 Multi-scale Relaxed Motion Segmentation

By varying the weight β in our energy functional, we can generate a multi-scale relaxed motion segmentation algorithm. Let us first consider two special cases: $\beta = 0$ and $\beta = 1$.

a) $\beta = 0$

For the case where the weight on the data fidelity terms is set to zero ($\beta = 0$), we encounter total smoothing of the foreground and background layers. We can replace the smooth estimates f_{in} and f_{out} with the respective means u_{fin} and u_{fout}. With $\beta = 0$, the equivalent energy formulation of E given in (81) is

$$\begin{aligned} E(\beta = 0) = & \int_{\Omega \cap R} (I_1 - u_{fin})^2 d\Omega + \int_{\Omega \setminus R} (I_1 - u_{fout})^2 d\Omega \\ & + \int_{\Omega \cap g_{in}(R)} (I_2 - u_{fin})^2 d\Omega + \int_{\Omega \setminus g_{in}(R)} (I_2 - u_{fout})^2 d\Omega + \lambda \int_C ds. \end{aligned} \tag{89}$$

b) $\beta = 1$

Let us denote the foreground and background layers for $\beta = 1$, as $I_{in} : \Omega_{I_{in}} \rightarrow \mathbb{R}$ over a domain $\Omega_{I_{in}} \subset \mathbb{R}^2$ and $I_{out} : \Omega_{I_{out}} \rightarrow \mathbb{R}$ over a domain $\Omega_{I_{out}} \subset \mathbb{R}^2$. With no weight on the smoothness terms, the minimal region of support for I_{in} and I_{out} are now

$$\Omega_{I_{in}} = {g_{in}}^{-1}(\Omega) \cap R, \tag{90}$$

$$\Omega_{I_{out}} = (\Omega \setminus R \cap {g_{out}}^{-1}(\Omega)) \setminus (({g_{out}}^{-1} \circ g_{in})(R)). \tag{91}$$

Since I_{in} and I_{out} are not defined over domains Ω_{in} and Ω_{out}, the modified energy functional in (81) becomes

$$\begin{aligned} E(\beta = 1) = & \int_{{g_{in}}^{-1}(\Omega) \cap R} (I_1 - I_{in})^2 d\Omega + \int_{\Omega \cap g_{in}(R)} (I_2 - I_{in} \circ {g_{in}}^{-1})^2 d\Omega \\ & + \int_{((\Omega \setminus R) \cap {g_{out}}^{-1}(\Omega)) - ({g_{out}}^{-1} \circ g_{in})(R)} (I_1 - I_{out})^2 d\Omega \\ & + \int_{\Omega \setminus g_{in}(R)} (I_2 - I_{out} \circ {g_{out}}^{-1})^2 d\Omega + \lambda \int_C ds. \end{aligned} \tag{92}$$

Not every pixel from images I_1 and I_2 are considered in these integrals, but only the pixels where I_{in} or I_{out} are defined is considered. Since the occluded regions and regions that extend from outside the image domain are not considered in the domain of integration, this energy functional has holes in the domain of integration. The domain of integration on the second term in (92) accounts for occlusion by excluding $({g_{out}}^{-1} \circ g_{in})(R)$ from the domain, and $(\Omega \setminus R) \cap {g_{out}}^{-1}(\Omega)$ accounts for the regions that do not have a value for the corresponding pixel in I_2. The domain of integration on the first term, ${g_{in}}^{-1}(\Omega) \cap R$, excludes the regions in the foreground that may move outside the domain of image I_2 due to foreground motion.

We do not use $E(\beta = 1)$ independently, since it is not defined over the entire image domain Ω. Instead, we consider a multi-scale relaxed motion segmentation energy model, which is a linear combination of the energy function for different weights of $\beta \in [0, 1]$. In the examples presented here, we consider a linear combination of $E(\beta = 0)$, $E(\beta = 0.5)$ and $E(\beta = 1)$.

$$E_{RMS} = a_0 \cdot E(\beta = 0) + a_1 \cdot E(\beta = 0.5) + a_2 \cdot E(\beta = 1), \tag{93}$$

where $a_0 + a_1 + a_2 = 1$ and $a_0, a_1, a_2 \geq 0$. This energy (E_{RMS}) is defined over the entire image domain. In the occluded regions and regions that extend from outside the image domain where I_{in} or I_{out} are not defined, we use only the smooth estimates f_{in}, u_{fin} and f_{out}, u_{fout}.

5.4 Experimental Results

The result of using the multi-scale relaxed energy model E_{RMS} (93) on Figure 46 (a) and Figure 46 (b) with the initialization given in Figure 46 (c) is shown in Figure 48 (a). We can see that the final curve evolves to match the object boundary. In the regions hampered by occlusion in the traditional model, our energy optimization model uses the smooth estimates of foreground and background layers to overcome the occlusion problem.

Figure 48 (b) shows the result of using the multi-scale relaxed motion segmentation model on images Figure 47 (a) and Figure 47 (b) with the initialization given in Figure 47 (c). Since our model estimates the background in the occluded regions, we avoid minima caused by occlusion and converge to the object boundary.

(a) Final Segmentation curve for images in Figure 46.

(b) Final Segmentation curve for images in Figure 47.

Figure 48: Results generated with the proposed multi-scale relaxed motion segmentation model.

Further we tested our relaxed motion segmentation algorithm on two image sequences: a) hockey player sequence with 15 consecutive frames, b) a teleconferencing image sequence with 100 consecutive frames. Once we converge to a minimum in the first frame of each sequence, we perform curve evolution for a fixed number of iterations on consequent frames, initializing the contour with the curve from the previous frame. We do not rely on motion continuity since the motion segmentation model uses only two consecutive frames. So we initialize the group action models g_{in} and g_{out} with no initial motion. Figures 49 and 50 show motion segmentation results on the hockey player sequence using the traditional approach and relaxed motion segmentation approach, respectively. Figures 51 and 52 compares motion segmentation results on a few intermediate frames of the teleconferencing image sequence.

We see that the traditional model does not converge to the object boundary correctly in the initial frames and hence the results degrade quickly in consequent frames. Whereas, the relaxed motion segmentation model overcomes the occlusion problem to converge to the object boundary in the initial frames and continues to detect the object in consecutive images of the sequence. Although the foreground motion is not purely translational due to the non-rigid motion of the object of interest, the segmentation results with our model are fairly robust.

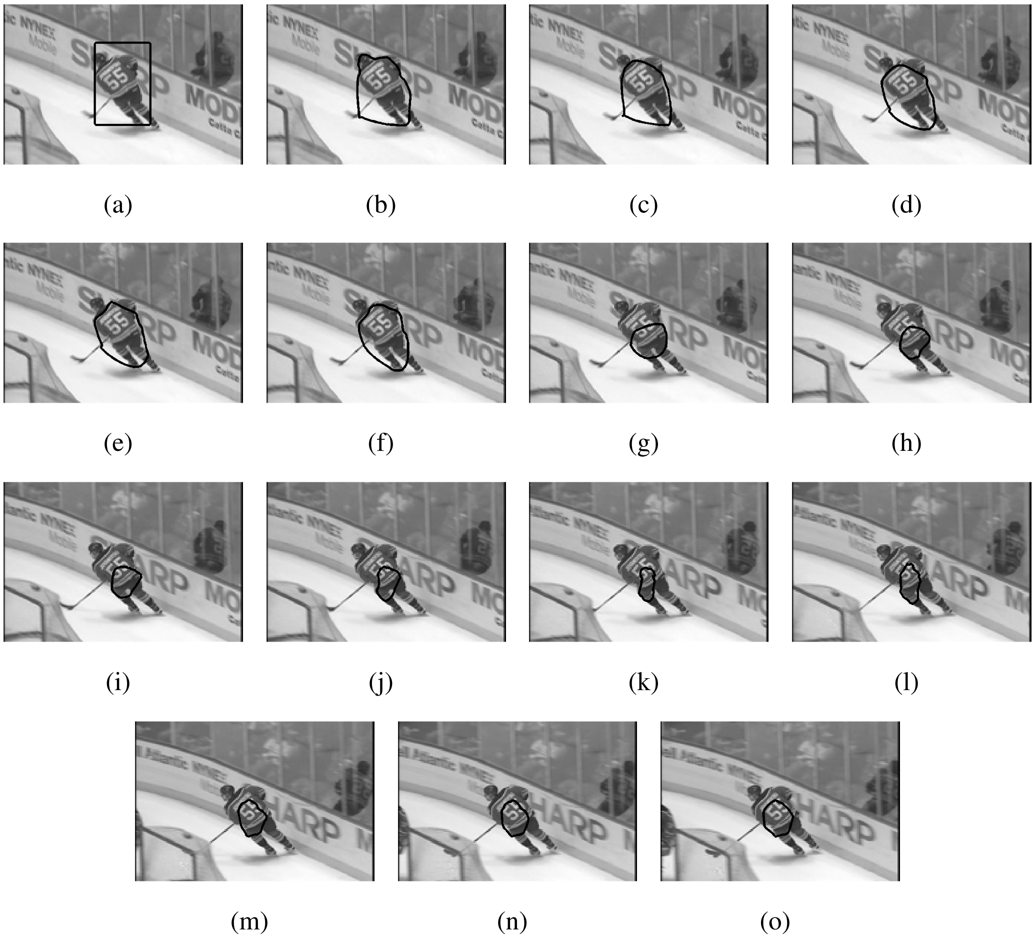

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) (m) (n) (o)

Figure 49: Tracking results: (a)Initialization on the first frame. (b-o) Motion segmentation results on the hockey player image sequence with the traditional motion segmentation model.

5.5 Conclusion

In this chapter, we proposed a novel relaxed motion segmentation model, which treats image sequences as overlapping foreground and background layers. We use smoothness constraint to model these layers using the available image information and also estimate smooth approximates for background in the occluded regions. Thus, we address the occlusion problem, which severely hampers existing motion segmentation models. We show results on some real-life images to illustrate the superiority of our model over the traditional

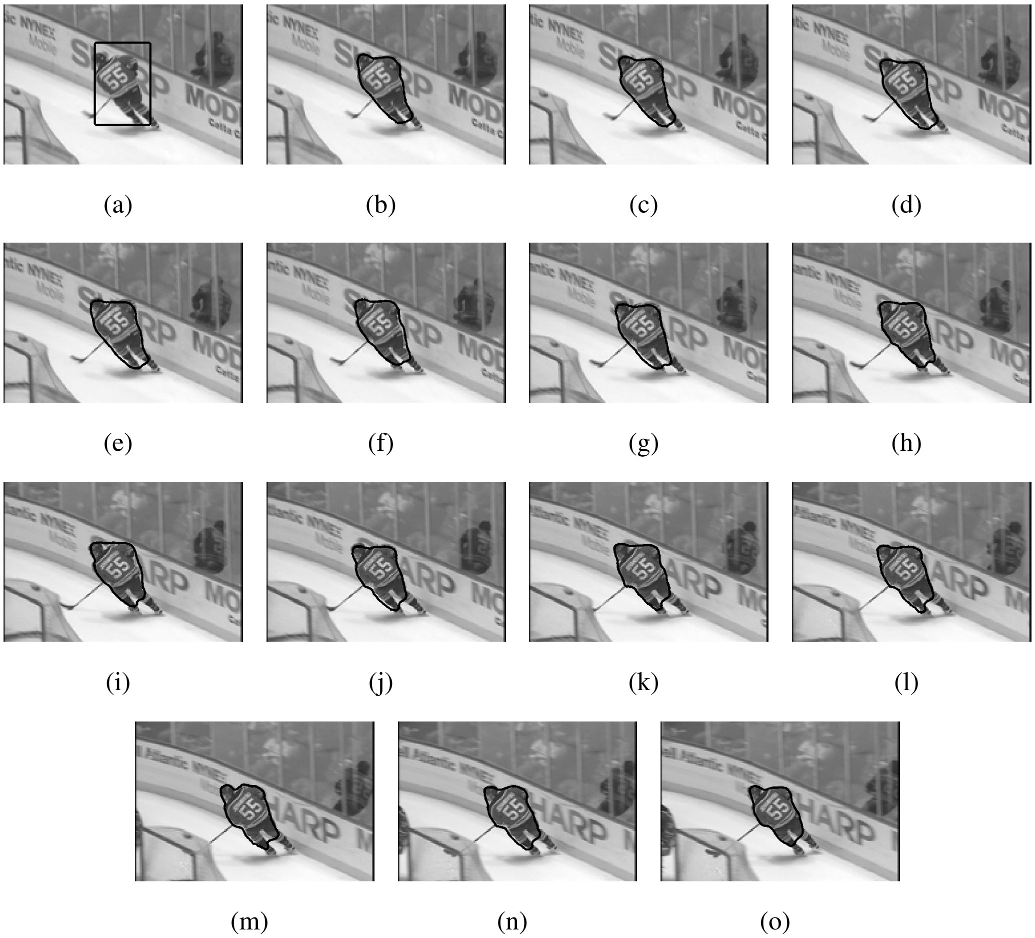
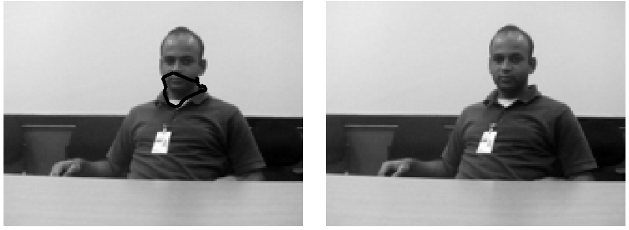

(a) (b) (c) (d)
(e) (f) (g) (h)
(i) (j) (k) (l)
(m) (n) (o)

Figure 50: Tracking results: (a)Initialization on the first frame. (b-o) Motion segmentation results on the hockey player image sequence with the proposed multi-scale relaxed motion segmentation model.

motion segmentation model.

(a) Initialization (b) Frame 1 (c) Frame 10

(d) Frame 20 (e) Frame 35 (f) Frame 45



(g) Frame 60 (h) Frame 70

Figure 51: Tracking results: (a) Initialization on the first frame. (b-g) Results on a few intermediate frames of the teleconferencing image sequence with the traditional motion segmentation model. (h) Due to accumulation of error over several frames, the contour completely disappears in frame 70.

(a) Initialization (b) Frame 1 (c) Frame 10

(d) Frame 20 (e) Frame 35 (f) Frame 45

(g) Frame 60 (h) Frame 70

Figure 52: Tracking results on a few intermediate frames of the teleconferencing image sequence with the proposed multi-scale relaxed motion segmentation model.

CHAPTER VI

CONCLUSION AND FUTURE WORK

In this thesis, we have developed three active contour models, which are robust against a larger class of local minima when compared to the traditional active contour models. We also developed two schemes to improve the numerical accuracy of the implementation of the traditional fast marching method. We present various experimental results, throughout the thesis, to illustrate the improvement in segmentation results caused by our approaches when compared to the existing state-of-the-art approaches for each active contour model.

6.1 Brief Summary of Contributions

- Developed *interpolated fast marching method* and *upsampled fast marching method* to remove directional bias and non-isotropy in the traditional numerical implementation of *fast marching methods*.

- Developed an *active geodesic* model, which constrains an evolving active contour to continually be a geodesic with respect to an edge-based metric throughout the evolution process. The contours generated by *active geodesic* model exhibit both local and global behaviors and therefore generate meaningful segmentation results.

- Developed an intuitive user interaction based segmentation algorithm using the *active geodesic model*.

- Developed *localized shape priors* based segmentation model to learn and fit curves that can account for local shape variations. This model segments various regions in the target image separately, and then combines them to obtain a single globally accurate segmentation, which retain local shape variation even as they are combined with global pose parameters.

- Developed a *relaxed motion segmentation* model to address the long-standing occlusion handling problem in motion segmentation models. This model uses a layering structure to estimate a foreground and background layer from given image sequence, and generates an approximation for background in the occluded region to circumvent the problem of detecting occlusions.

6.2 Future Work

There are various avenues for future research on each of the topics we have dealt with in this thesis. Below we enlist a few suggestions on the most interesting and piratical direction of future research for each of the research topics we have discussed in this thesis:

- We discuss the implementation of *interpolated fast marching method* and *upsampled fast marching method*, in Chapter 2, on Cartesian grids. Investigating the benefits of our approach on non-Cartesian grids is an interesting future direction for research.

- The current implementation of the *active geodesic* model, presented in Chapter 3, is strictly applicable to closed contours on 2D images. Extending our model to higher dimensional data is a challenging future direction of research.

- In the current implementation of the *localized shape prior* based segmentation approach, discussed in Chapter 4, we require user input for target mask generation. The target mask generation can be automated by using variational methods to generate masks that increase correlation in local shape variation within each mask and decrease the correlation in shape variation across different target masks.

- Incorporating intensity priors, to learn the intensity distribution in training images, along with shape priors to improve prior knowledge-based segmentation can be an interesting avenue to extend the localized shape prior based approach. Such intensity prior based segmentation approaches can be very useful in various medical imaging applications.

- Localized shape priors can be incorporated in the *relaxed motion segmentation* model, discussed in Chapter 5, on video teleconferencing image sequences. We can use various target regions to identify articulated motion of the subject in a teleconferencing sequence and use localized shape priors for applications such as background substitution and 2D to 3D conversion.

APPENDIX A

FERRARI'S SOLUTION TO QUARTIC EQUATIONS

In this appendix, we derive the solution to a generic fourth order equation [60]. Consider the quartic equation

$$ax^4 + bx^3 + cx^2 + dx + e = 0. \tag{94}$$

$$\implies x^4 + \frac{b}{a}x^3 + \frac{c}{a}x^2 + \frac{d}{a}x + \frac{e}{a} = 0. \tag{95}$$

Substituting, $x = (y - \frac{b}{4a})$ in equation (95), we get a depressed quartic equation,

$$\begin{aligned} y^4 &+ (-\frac{3b^2}{8a^2} + \frac{c}{a})y^2 + (\frac{b^3}{8a^3} - \frac{bc}{2a^2} + \frac{d}{a})y \\ &+ (-\frac{3b^4}{256a^4} + \frac{b^2c}{16a^3} - \frac{bd}{4a^2} + \frac{e}{a}) = 0. \end{aligned} \tag{96}$$

Substituting, $p = (-\frac{3b^2}{8a^2} + \frac{c}{a})$, $q = (\frac{b^3}{8a^3} - \frac{bc}{2a^2} + \frac{d}{a})$ and $r = (-\frac{3b^4}{256a^4} + \frac{b^2c}{16a^3} - \frac{bd}{4a^2} + \frac{e}{a})$ in (96) we get,

$$y^4 + py^2 + qy + r = 0. \tag{97}$$

$$\implies (y^2 + p)^2 = py^2 - qy + (p^2 - r). \tag{98}$$

Introducing a new variable z, by adding $(y^2 + p + z)^2 - (y^2 + p)^2$ to both sides of (98), we get,

$$(y^2 + p + z)^2 = (p + 2z)y^2 - qy + (z^2 + 2pz + p^2 - r). \tag{99}$$

We have to choose a value of z such that the R.H.S of (99) becomes a perfect square. Thus, the discriminant of the R.H.S of (99), which is a quadratic equation of the form $Ay^2 + By + C$, should be zero, *i.e.* $B^2 - 4AC = 0$. Thus, we have

$$q^2 - 4(p + 2z)(z^2 + 2pz + p^2 - r) = 0. \tag{100}$$

$$\implies z^3 + \frac{5}{2}pz^2 + (2p^2 - r)z + (\frac{p^3}{2} - \frac{pr}{2} - \frac{q^2}{8}) = 0. \tag{101}$$

Substituting, $z = (v - \frac{5}{6}p)$ in equation (100), we get a depressed cubic equation,

$$v^3 + (-\frac{p^2}{12} - r)v + (-\frac{p^3}{108} + \frac{pr}{3} - \frac{q^2}{8}) = 0. \tag{102}$$

Substituting, $P = (-\frac{p^2}{12} - r)$ and $Q = (-\frac{p^3}{108} + \frac{pr}{3} - \frac{q^2}{8})$ in (102) we get,

$$v^3 + Pv + Q = 0. \tag{103}$$

The solution to this depressed cubic equation using Cardano's method [60] gives us the solution to the cubic equation in (101). For $s = \sqrt[3]{-\frac{Q}{2} \pm \sqrt{\frac{Q^2}{4} + \frac{P^3}{27}}}$,

$$z = \begin{cases} -s + \frac{P}{3s} - \frac{5p}{6} & s \neq 0, \\ -\sqrt[3]{Q} - \frac{5p}{6} & \text{otherwise.} \end{cases} \tag{104}$$

Consider the equation $(Ay + B)^2$. We have

$$(Ay + B)^2 = ((\sqrt{A^2})y + \frac{2AB}{2\sqrt{A^2}})^2. \tag{105}$$

Using this expression in (99), we get,

$$(y^2 + p + z)^2 = (\sqrt{p + 2z}y + \frac{(-q)}{2\sqrt{p + 2z}})^2. \tag{106}$$

$$\implies y^2 + p + z = \pm(\sqrt{p + 2z}y + \frac{(-q)}{2\sqrt{p + 2z}}). \tag{107}$$

$$\implies y^2 \mp (\sqrt{p + 2z})y + (p + z \pm \frac{q}{2\sqrt{p + 2z}}) = 0. \tag{108}$$

Solution to the quadratic equation in (108) gives us the solution of the original quartic equation.

$$x = \begin{cases} -\frac{b}{4a} \pm \frac{\sqrt{p+2z}}{2} + \frac{\sqrt{(p+2z) - 4(p+z \pm \frac{q}{2\sqrt{p+2z}})}}{2}, \\ -\frac{b}{4a} \pm \frac{\sqrt{p+2z}}{2} - \frac{\sqrt{(p+2z) - 4(p+z \pm \frac{q}{2\sqrt{p+2z}})}}{2}. \end{cases} \tag{109}$$

APPENDIX B

NEWTON'S SOLUTION FOR 4-CONNECTED NEIGHBOR SCHEME

In this appendix, we prove the convergence of Newton's scheme for the interpolated 4-connected neighbor scheme. We also discuss solutions to the corner cases with multiple minima/maxima. Let us consider the higher order derivatives of $f(t)$ given in equation (19).

$$f''(t) = \frac{u_A - u_B}{\left(\sqrt{t^2 + (1-t)^2}\right)^3} + 4(\tau_B - \tau_A), \tag{110}$$

$$f^{(3)}(t) = -3(u_A - u_B)\Big\{\frac{(2t-1)}{\left(\sqrt{t^2 + (1-t)^2}\right)^5}\Big\}, \tag{111}$$

$$\begin{aligned} f^{(4)}(t) = -3(u_A - u_B)\Big\{&\frac{2}{\left(\sqrt{t^2 + (1-t)^2}\right)^5} \\ &- \frac{5(2t-1)^2}{\left(\sqrt{t^2 + (1-t)^2}\right)^7}\Big\}. \end{aligned} \tag{112}$$

We note that for $u_A - u_B \neq 0$, $f^{(4)}(t) \neq 0$. If $u_A - u_B = 0$, $f'(t)$ in equation (20) (in Chapter 2) becomes a linear function in t and thus it can have at most one zero crossing for $t \in [0,1]$. Thus, Newton's method has at least quadratic convergence and $u_C(t)$ can have at most four extrema. This also implies that f' can have a maximum of three zero crossings. We already know that $f'(0.5)$ is non-negative. Also, note that f'' is symmetric about $t = 0.5$, and $f^{(3)}(t)$ has a zero at $t = 0.5$; hence, there can be a single zero for f'' at $t = 0$ or a maximum of two zero crossings in f'' for $t \in [0,1]$. Using the symmetry of f'' and non-negativity of $f'(0.5)$, we can detect the number of zero crossings in $f(t)$

(equivalently the number of extrema in u_C) by looking at the sign of f' at $t = 0$ and $t = 1$, and sign of f'' at $t = 0$ and $t = 0.5$. The four possible cases are

(i) Maximum one zero crossing in f: $f'(0)$, $f'(1)$ have the same sign and $f''(0)$, $f''(0.5)$ have the same sign.

(ii) Maximum two zero crossings in f: $f'(0)$, $f'(1)$ have opposite signs and $f''(0)$, $f''(0.5)$ have the same sign.

(iii) Maximum three zero crossings in f: $f'(0)$, $f'(1)$ have the same sign and $f''(0)$, $f''(0.5)$ have opposite signs.

(iv) Maximum four zero crossings in f: $f'(0)$, $f'(1)$ have opposite signs and $f''(0)$, $f''(0.5)$ have opposite signs.

If we detect a single extremum, we can use $t_{init} = 0.5$ to initialize Newton's scheme. For two extrema, we can initialize Newton's scheme at the two boundaries. If we detect three extrema, it could mean 2 maxima and one minimum or one maximum and 2 minima. We can initialize the Newton's scheme at the two boundaries and converge to the corresponding extrema. If both the extrema have a higher value than the boundaries, we initialize Newton's method again with the midpoint of the two detected extrema. If the extrema have a lower value than the boundaries, the minimum of the two extrema corresponds to the global minimum. Finally, in the case with four extrema we initialize Newton's method at the two boundaries and detect one of the two local minima. We encountered the scenario with 2, 3 and 4 extrema on $\sim 1.5\%$, 0% and $\sim 0.02\%$ pixels, respectively, on a 500x500 random noise image.

APPENDIX C

NEWTON'S SOLUTION FOR 8-CONNECTED NEIGHBOR SCHEMES

The higher order derivatives of $f(t)$ for 8-connected neighbor scheme, given in equation (27) (in Chapter 2), are

$$f''(t) = \frac{u_A - u_B}{\left(\sqrt{1+t^2}\right)^3} + 2(\tau_A - \tau_B), \tag{113}$$

$$f^{(3)}(t) = -3(u_A - u_B)\Big\{\frac{t}{\left(\sqrt{1+t^2}\right)^5}\Big\}. \tag{114}$$

If $u_A - u_B = 0$, $f'(t)$ in equation (28) (in Chapter 2) becomes a linear function, which can have at the most a single zero crossing. Now, given $u_A - u_B \neq 0$, $f^{(3)}(t)$ has a single zero at $t = 0$. The symmetry of $f''(t)$ about $t = 0$ suggests that there can be at most one zero crossing for $t \in [0, 1]$. Since $f'(0)$ is non-negative, we can detect the number of extrema in the solution of the equation (23) by detecting the sign of f' and f'' at the boundaries $t = 0$ and $t = 1$. The three possible cases are

(i) Maximum one zero crossing in f: $f'(0)$, $f'(1)$ have the same sign and $f''(0)$, $f''(1)$ have the same sign.

(ii) Maximum two zero crossings in f: $f'(0)$, $f'(1)$ have opposite signs, irrespective of the sign of $f''(0)$, $f''(1)$.

(iii) Maximum three zero crossings in f: $f'(0)$, $f'(1)$ have the same sign and $f''(0)$, $f''(1)$ have opposite signs.

If $u_C(t)$ has a single extremum, we initialize Newton's method with $t_{init} = 0$. If $u_C(t)$ has 2 or 3 extrema, we use the strategy discussed in Appendix B. On the 500x500 random noise image, we encountered scenarios with 2 and 3 extrema in $\sim 6\%$ and $\sim 0.09\%$ pixels, respectively. A similar derivation can be employed for the 8-connected neighbor bilinear interpolation scheme.

APPENDIX D

DERIVATION FOR EQUATIONS IN UPDATING MOTION VECTORS FOR THE RELAXED MOTION SEGMENTATION MODEL

The update equation for any given parameter set p_i can be written as

$$p_i{}^{n+1} = p_i{}^{n} - \frac{\partial E}{\partial p_i} dt,$$

where $i = \{\text{in}, \text{out}\}$ and dt is a positive step-size.

Let us first derive $\frac{\partial E}{\partial p_{in}}$. Taking the derivative of the energy function for the relaxed motion segmentation (Equation (81)), we get

$$\frac{\partial E}{\partial p_{in}} = \frac{\partial}{\partial p_{in}} [\int_{\Omega \cap R} (I_2 - f_{in} \circ g_{in}{}^{-1})^2] d\Omega, \tag{115}$$

Since, $(I_2 - f_{in} \circ g_{in}{}^{-1})^2$ is the only term which depends on p_{in}, the other terms vanish. Splitting the integral on the R.H.S of equation (115) into the region term and the boundary term of the curve C, we get

$$\begin{aligned} \frac{\partial E}{\partial p_{in}} =& 2 \int_{\Omega \cap R} (I_2 - f_{in} \circ g_{in}{}^{-1}) \cdot (\nabla f_{in} \circ g_{in}{}^{-1}) \cdot \frac{\partial g_{in}{}^{-1}}{\partial p_{in}} d\Omega \\ &+ \int_{C} (I_2 - f_{in} \circ g_{in}{}^{-1})^2 \cdot \frac{\partial C}{\partial p_{in}} \cdot \vec{N} \cdot ds. \end{aligned} \tag{116}$$

Parameterizing the curve C such that $C(L) : [0, 1] \to \mathbb{R}^2$, we get

$$\begin{aligned} \frac{\partial E}{\partial p_{in}} =& 2 \int_{\Omega \cap R} (I_2 - f_{in} \circ g_{in}{}^{-1}) \cdot (\nabla f_{in} \circ g_{in}{}^{-1}) \cdot \frac{\partial g_{in}{}^{-1}}{\partial p_{in}} d\Omega \\ &+ \int_{0}^{1} (I_2 \circ C - f_{in} \circ g_{in}{}^{-1} \circ C)^2 \cdot \frac{\partial C}{\partial p_{in}} \cdot J \cdot \frac{\partial C}{\partial L} \cdot dL. \end{aligned} \tag{117}$$

$$\begin{aligned} \implies \frac{\partial E}{\partial p_{in}} =& 2 \int_{\Omega \cap R} (I_2 \circ g_{in} - f_{in}) \cdot (\nabla f_{in}) \cdot (\frac{\partial g_{in}{}^{-1}}{\partial p_{in}} \circ g_{in}) \cdot |\frac{\partial g_{in}}{\partial X}| d\Omega \\ &+ \int_{0}^{1} (I_2 \circ g_{in} \circ C - f_{in} \circ C)^2 \cdot \frac{\partial g_{in} \circ C}{\partial p_{in}} \cdot J \cdot \frac{\partial (g_{in} \circ C)}{\partial L} \cdot dL. \end{aligned} \tag{118}$$

Note that,

$$\frac{\partial}{\partial p_{in}}({g_{in}}^{-1}(g_{in}(X,p_{in}),p_{in})=0. \tag{119}$$

$$\implies \frac{\partial {g_{in}}^{-1}}{\partial p_{in}}(g_{in}(X,p_{in}),p_{in})+\frac{\partial {g_{in}}^{-1}}{\partial X}(g_{in}(X,p_{in}),p_{in})\cdot\frac{\partial g_{in}(X,p_{in})}{\partial p_{in}}=0. \tag{120}$$

Thus, substituting (120) in (118) we get

$$\begin{aligned}\frac{\partial E}{\partial p_{in}}=&-2\int_{\Omega\cap R}(I_2\circ g_{in}-f_{in})\cdot(\nabla f_{in})\cdot((\frac{\partial {g_{in}}^{-1}}{\partial X}\circ g_{in})\cdot\frac{\partial g_{in}}{\partial p_{in}})\cdot|\frac{\partial g_{in}}{\partial X}|d\Omega\\&+\int_0^1(I_2\circ g_{in}\circ C-f_{in}\circ C)^2\cdot\frac{\partial g_{in}}{\partial p_{in}}\circ C\cdot J\cdot(\frac{\partial g_{in}}{\partial X}\circ C)\cdot\frac{\partial C}{\partial L}dL.\end{aligned} \tag{121}$$

Now we have

$$(\frac{\partial {g_{in}}^{-1}}{\partial X}\circ g_{in})=(\frac{\partial g_{in}}{\partial X})^{-1}, \tag{122}$$

and

$$\frac{\partial C}{\partial L}=T\cdot|\frac{\partial C}{\partial L}|. \tag{123}$$

$$\implies \frac{\partial C}{\partial L}=J^T\cdot N\cdot|\frac{\partial C}{\partial L}|. \tag{124}$$

Thus substituting (122) and (124) in (121) we get

$$\begin{aligned}\frac{\partial E}{\partial p_{in}}=&-2\int_{\Omega\cap R}(I_2\circ g_{in}-f_{in})\cdot(\nabla f_{in})\cdot(\frac{\partial g_{in}}{\partial X})^{-1}\cdot|\frac{\partial g_{in}}{\partial X}|\cdot\frac{\partial g_{in}}{\partial p_{in}}d\Omega\\&+\int_0^1(I_2\circ g_{in}\circ C-f_{in}\circ C)^2\cdot\frac{\partial g_{in}}{\partial p_{in}}\circ C\cdot J\cdot(\frac{\partial g_{in}}{\partial X}\circ C)\cdot J^T\cdot N\cdot|\frac{\partial C}{\partial L}|dL.\end{aligned} \tag{125}$$

By definition of the adjoint matrix we have,

$$J\cdot(\frac{\partial g_{in}}{\partial X}\circ C)\cdot J^T=\text{adj}(\frac{\partial g_{in}}{\partial X}\circ C)^T. \tag{126}$$

Substituting (126) in (125), we get

$$\begin{aligned}\frac{\partial E}{\partial p_{in}}=&-2\int_{\Omega\cap R}(I_2\circ g_{in}-f_{in})\cdot(\nabla f_{in})\cdot\text{adj}\left(\frac{\partial g_{in}}{\partial X}\right)\cdot\frac{\partial g_{in}}{\partial p_{in}}d\Omega\\&+\int_C(I_2\circ g_{in}-f_{in})^2\cdot\frac{\partial g_{in}}{\partial p_{in}}\cdot\text{adj}(\frac{\partial g_{in}}{\partial X})^T\cdot N\cdot ds.\end{aligned} \tag{127}$$

Following a similar analysis we can derive $\frac{\partial E}{\partial p_{out}}$ to update p_{out}.

$$\begin{aligned}\frac{\partial E}{\partial p_{out}} =& -2\int_{{g_{out}}^{-1}(\Omega \backslash g_{in}(R))} (I_2 \circ g_{out} - f_{out}) \cdot (\nabla f_{out}) \cdot \text{adj}\left(\frac{\partial g_{out}}{\partial X}\right) \cdot \frac{\partial g_{out}}{\partial p_{out}} d\Omega \\ &+ \int_C (I_2 \circ g_{in} - f_{out} \circ {g_{out}}^{-1} \circ g_{in})^2 \cdot \frac{\partial g_{in}}{\partial p_{out}} \cdot \text{adj}\left(\frac{\partial g_{in}}{\partial X}\right)^T \cdot N \cdot ds.\end{aligned} \tag{128}$$

REFERENCES

[1] Abufadel, A., Yezzi, A. J., and Schafer, R., “4d segmentation of cardiac data using active surfaces with spatiotemporal shape priors,” pp. 77–100, 2008.

[2] Adalsteinsson, D. and Sethian, J. A., “A fast level set method for propagating interfaces,” *Journal of computational physics*, vol. 118, pp. 269–277, 1994.

[3] Alvarez, L., Weickert, J., and Sanchez, J., “Reliable estimation of dense optical flow fields with large displacements,” *International Journal of Computer Vision*, vol. 39, no. 1, pp. 41–56, 2000.

[4] Amberg, M., Luthi, M., and Vetter, T., “Local regression based statistical model fitting,” in *Proceedings of the 32nd DAGM conference on Pattern recognition*, pp. 452–461, 2010.

[5] Appia, V., Ganapathy, B., Yezzi, A., and Faber, T., “Localized principal component analysis based curve evolution: A divide and conquer approach,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1981 –1986, nov. 2011.

[6] Appia, V., Patil, U., and Das, B., “Lung fissure detection in ct images using global minimal paths,” *Medical Imaging, SPIE*, vol. 7623, 2010.

[7] Appia, V. and Yezzi, A., “Fully isotropic fast marching methods on cartesian grids,” ECCV’10, (Berlin, Heidelberg), pp. 71–83, Springer-Verlag, 2010.

[8] Appia, V. and Yezzi, A., “Active geodesics: Region-based active contour segmentation with a global edge-based constraint,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1975 –1980, nov. 2011.

[9] APPIA, V. V., GANAPATHY, B., ABUFADEL, A., YEZZI, A. J., and FABER, T., "A regions of confidence based approach to enhance segmentation with shape priors," *Computational Imaging, SPIE*, vol. 7533, no. 1, 2010.

[10] APPLETON, B., "Optimal geodesic active contours: Application to heart segmentation," 2003.

[11] APPLETON, B. and TALBOT, H., "Globally optimal geodesic active contours," *Jorunal of mathematical imaging and vision*, vol. 23, pp. 67–86, 2005.

[12] ARDON, R. A., COHEN, L. D., and YEZZI, A. J., "Fast surface segmentation guided by user input using implicit extension of minimal paths," *Journal of Mathematical Imaging and Vision*, vol. 25, no. 3, pp. 289–305, 2006.

[13] AYVACI, A. and SOATTO, S., "Motion segmentation with occlusions on the superpixel graph," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pp. 727 –734, 27 2009-oct. 4 2009.

[14] BENMANSOUR, F., BONNEAU, S., and COHEN, L., "Finding a closed boundary by growing minimal paths from a single point on 2d or 3d images," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1 –8, oct. 2007.

[15] BENMANSOUR, F. and COHEN, L. D., "Tubular anisotropy segmentation," in *Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision*, (Berlin, Heidelberg), pp. 14–25, Springer-Verlag, 2009.

[16] BLACK, M. J. and ANANDAN, P., "The robust estimation of multiple motions: parametric and piecewise-smooth flow fields," *Computer vision image understanding*, vol. 63, pp. 75–104, 1996.

[17] BOYKOV, Y. and JOLLY, M.-P., "Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, pp. 105 –112 vol.1, 2001.

[18] BRESSON, X., ESEDOGLU, S., VANDERGHEYNST, P., THIRAN, J.-P., and OSHER, S., "Fast global minimization of the active contour/snake model," *Journal of Mathematical Imaging and Vision*, vol. 28, no. 2, pp. 151–167, 2007.

[19] BROX, T. and MALIK, J., "Large displacement optical flow: Descriptor matching in variational motion estimation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, pp. 500 –513, march 2011.

[20] BROX, T., BRUHN, A., and WEICKERT, J., "Variational motion segmentation with level sets," in *ECCV (1)*, pp. 471–483, 2006.

[21] CANNY, J., "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, pp. 679 –698, nov. 1986.

[22] CASELLES, V., KIMMEL, R., and SAPIRO, G., "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, pp. 61–79, 1997.

[23] CHAKRABORTY, A. and DUNCAN, J., "Game-theoretic integration for image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, pp. 12 –30, jan 1999.

[24] CHAN, T. and VESE, L., "Active contours without edges," *Image Processing, IEEE Transactions on*, vol. 10, pp. 266 –277, feb 2001.

[25] CHAN, T. and VESE, L., "A level set algorithm for minimizing the mumford-shah functional in image processing," in *Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on*, pp. 161 –168, 2001.

[26] CHAN, T. F., SANDBERG, Y. B., and VESE, L. A., "Active contours without edges for vector-valued images," *Journal of visual communication and image representation*, vol. 11, pp. 130–141, 2000.

[27] CHEN, Y., THIRUVENKADAM, S., TAGARE, H., HUANG, F., WILSON, D., and GEISER, E., "On the incorporation of shape priors into geometric active contours," in *Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on*, pp. 145 –152, 2001.

[28] CHOPP, D. L., "Some improvements of the fast marching method," *SIAM Journal on Scienctific Computing*, vol. 23, pp. 230–244, January 2001.

[29] COHEN, L. and KIMMEL, R., "Edge integration using minimal geodesics," *Technical report*, 1995.

[30] COHEN, L. and KIMMEL, R., "Global minimum for active contour models: A minimal path approach," *International Journal of Computer Vision*, vol. 24, no. 1, pp. 57–78, 1997.

[31] COHEN, L. D., "On active contour models and balloons," *Computer vision, graphics, and image processing*, vol. 53, pp. 211–218, 1991.

[32] COHEN, L. D. and DESCHAMPS, T., "Multiple contour finding and perceptual grouping as a set of energy minimizing paths," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 560–575, 2001.

[33] COHEN, L. and KIMMEL, R., "Global minimum for active contour models: a minimal path approach," in *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, pp. 666 –673, jun 1996.

[34] Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J., "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[35] Cremers, D., "A multiphase level set framework for motion segmentation," in *4th International Conference on Scale Space Theories in Computer Vision*, pp. 599–614, Springer, 2003.

[36] Cremers, D. and Soatto, S., "Motion competition: A variational approach to piecewise parametric motion segmentation," *International Journal of Computer Vision*, vol. 62, pp. 249–265, 2005.

[37] Cremers, D., Sochen, N., and Schnörr, C., "A multiphase dynamic labeling model for variational recognition-driven image segmentation," *International Journal Computer Vision*, vol. 66, pp. 67–81, January 2006.

[38] Cremers, D. C., Osher, S., and Soatto, S., "Kernel density estimation and intrinsic alignment for knowledge-driven segmentation: Teaching level sets to walk," in *Pattern Recognition, 26th DAGM-Symposium*, pp. 36–44, 2004.

[39] Dambreville, S., Rathi, Y., and Tannenbaum, A., "A framework for image segmentation using shape models and kernel space shape priors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, pp. 1385 –1399, aug. 2008.

[40] Danielsson, P.-E. and Lin, Q., "A modified fast marching method," in *13th Scandinavian Conference on Image Analysis*, vol. 2749, pp. 631–644, 2003.

[41] Davatzikos, C., Tao, X., and Shen, D., "Hierarchical active shape models, using the wavelet transform," *Medical Imaging, IEEE Transactions on*, vol. 22, pp. 414 – 423, march 2003.

[42] DESCHAMPS, T. and COHEN, L. D., "Fast extraction of minimal paths in 3d images and applications to virtual endoscopy," *Medical Image Analysis*, vol. 5, no. 4, pp. 281–299, 2001.

[43] FARNEBACK, G., "Fast and accurate motion estimation using orientation tensors and parametric motion models," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 1, pp. 135 –139 vol.1, 2000.

[44] GONZALEZ, R. C. and WOODS, R. E., *Digital Image Processing*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2nd ed., 2001.

[45] GRADY, L., "Random walks for image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, pp. 1768 –1783, nov. 2006.

[46] HARALICK, R. M. and SHAPIRO, L. G., "Image segmentation techniques," *Computer vision, graphics, and image processing*, vol. 29, no. 1, pp. 100 – 132, 1985.

[47] HASSOUNA, M. and FARAG, A., "Multistencils fast marching methods: A highly accurate solution to the eikonal equation on cartesian domains," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, pp. 1563 –1574, sept. 2007.

[48] HORN, B. K. P. and SCHUNCK, B. G., "Determining optical flow," *Artificial intelligence*, vol. 17, pp. 185–203, 1981.

[49] INCE, S. and KONRAD, J., "Occlusion-aware optical flow estimation," *Image Processing, IEEE Transactions on*, vol. 17, pp. 1443 –1451, aug. 2008.

[50] JACKSON, J., YEZZI, A. J., and SOATTO, S., "Dynamic shape and appearance modeling via moving and deforming layers," *International Journal of Computer Vision*, vol. 79, no. 1, pp. 71–84, 2008.

[51] KASS, M., WITKIN, A., and TERZOPOULOS, D., "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[52] KAUL, V., TSAI, Y., and YEZZI, A. J., "Detection of curves with unknown endpoints using minimal path techniques," in *British Machine Vision Conference*, pp. 1–12, 2010.

[53] KICHENASSAMY, S., KUMAR, A., OLVER, P. J., TANNENBAUM, A., and YEZZI, A. J., "Gradient flows and geometric active contour models," in *Computer Vision (ICCV), 1995 IEEE International Conference on*, (Washington, DC, USA), pp. 810–, IEEE Computer Society, 1995.

[54] KICHENASSAMY, S., KUMAR, A., OLVER, P. J., TANNENBAUM, A., and YEZZI, A. J., "Conformal curvature flows: From phase transitions to active vision," *Archive for Rational Mechanics and Analysis*, vol. 134, no. 3, pp. 275–301, 1996.

[55] KIM, S., "An o(n) level set method for eikonal equations," *SIAM journal on scientific computing*, vol. 22, pp. 2178–2193, 2000.

[56] KIMMEL, R. and SETHIAN, J. A., "Computing geodesic paths on manifolds," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, pp. 8431–8435, July 1998.

[57] KIMMEL, R. and SETHIAN, J. A., "Optimal algorithm for shape from shading and path planning," *Journal of Mathematical Imaging and Vision*, vol. 14, no. 3, pp. 237–244, 2001.

[58] KIMMEL, R., SHAKED, D., KIRYATI, N., and BRUCKSTEIN, A. M., "Skeletonization via distance maps and level sets," *Computer Vision and Image Understanding*, vol. 62, no. 3, pp. 382–391, 1995.

[59] KOHLBERGER, T., SCHNORR, C., BRUHN, A., and WEICKERT, J., "Domain decomposition for variational optical-flow computation," *Image Processing, IEEE Transactions on*, vol. 14, pp. 1125 –1137, aug. 2005.

[60] KORN, G. A. and KORN, T. M., *Mathematical handbook for scientists and engineers*. Courier Dover Publications, 2000.

[61] LEVENTON, M., GRIMSON, W., and FAUGERAS, O., "Statistical shape influence in geodesic active contours," in *Biomedical Imaging, 2002. 5th IEEE EMBS International Summer School on*, p. 8 pp., june 2002.

[62] LI, H. and YEZZI, A., "Vessels as 4-d curves: Global minimal 4-d paths to extract 3-d tubular surfaces and centerlines," *Medical Imaging, IEEE Transactions on*, vol. 26, pp. 1213 –1223, sept. 2007.

[63] LIN, Q., *Enhancement, Extraction, and Visualization of 3D Volume Data*. PhD thesis, Linkpings Universitet, Sweden, 2003.

[64] LUCAS, B. D. and KANADE, T., "An iterative image registration technique with an application to stereo vision," in *International joint conference on artificial intelligence*, pp. 674–679, 1981.

[65] MALLADI, R., SETHIAN, J. A., and VEMURI, B. C., "Evolutionary fronts for topology-independent shape modeling and recovery," in *ECCV (1)*, pp. 3–13, 1994.

[66] MORTENSEN, E. N. and BARRETT, W. A., "Intelligent scissors for image composition," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, (New York, NY, USA), pp. 191–198, ACM, 1995.

[67] MUMFORD, D. and SHAH, J., "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on pure and applied mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[68] MUMFORD, D. and SHAH, J., "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on pure and applied mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[69] NIKOLOVA, M., ESEDOGLU, S., and CHAN, T. F., "Algorithms for finding global minimizers of image segmentation and denoising models," *SIAM Journal of Applied Mathematics*, vol. 66, no. 5, pp. 1632–1648, 2006.

[70] OSHER, S. and SETHIAN, J. A., "Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations," *Journal of computational physics*, vol. 79, pp. 12–49, 1988.

[71] OSHER, S. and SETHIAN, J. A., "Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations," *Journal of computational physics*, vol. 79, no. 1, pp. 12–49, 1988.

[72] PARAGIOS, N. and DERICHE, R., "Geodesic active contours and level sets for the detection and tracking of moving objects," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, pp. 266 –280, mar 2000.

[73] PARAGIOS, N. and DERICHE, R., "Coupled geodesic active regions for image segmentation: A level set approach," in *ECCV*, pp. 224–240, 2000.

[74] PETRES, C., PAILHAS, Y., PATRON, P., PETILLOT, Y., EVANS, J., and LANE, D., "Path planning for autonomous underwater vehicles," *Robotics, IEEE Transactions on*, vol. 23, pp. 331 –341, april 2007.

[75] POLYMENAKOS, L., BERTSEKAS, D., and TSITSIKLIS, J., "Implementation of efficient algorithms for globally optimal trajectories," *Automatic Control, IEEE Transactions on*, vol. 43, pp. 278 –283, feb 1998.

[76] RONFARD, R., "Region-based strategies for active contour models," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 229–251, 1994.

[77] ROTHER, C., KOLMOGOROV, V., and BLAKE, A., ""grabcut": interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.

[78] ROUSSON, M. and PARAGIOS, N., "Shape priors for level set representations," in *ECCV (2)*, pp. 78–92, 2002.

[79] ROUSSON, M. and PARAGIOS, N., "Prior knowledge, level set representations & visual grouping," *International Journal of Computer Vision*, vol. 76, no. 3, pp. 231–243, 2008.

[80] ROUY, E. and TOURIN, A., "A viscosity solutions approach to shape-from-shading," *SIAM journal of numerical analysis*, vol. 29, pp. 867–884, 1992.

[81] SAMSON, C., BLANC-FÉRAUD, L., AUBERT, G., and ZERUBIA, J., "A level set model for image classification," in *Scale-Space Theories in Computer Vision*, pp. 306–317, 1999.

[82] SETHIAN, J. A., *Level set methods and fast marching methods*. Cambridge University Press, 1999.

[83] SETHIAN, J. A., *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 2 ed., June 1999.

[84] SHI, J. and MALIK, J., "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, pp. 888 –905, aug 2000.

[85] SKOCAJ, D., LEONARDIS, A., and BISCHOF, H., "Weighted and robust learning of subspace representations," *Pattern Recognition*, vol. 40, no. 5, pp. 1556–1569, 2007.

[86] STEINBRUCKER, F., POCK, T., and CREMERS, D., "Large displacement optical flow computation withoutwarping," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 1609 –1614, 29 2009-oct. 2 2009.

[87] SUN, C. and PALLOTTINO, S., "Circular shortest path in images," *Pattern Recognition*, vol. 36, no. 3, pp. 709–719, 2003.

[88] SUNDBERG, P., BROX, T., MAIRE, M., ARBELAEZ, P., and MALIK, J., "Occlusion boundary detection and figure/ground assignment from optical flow," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 2233 –2240, june 2011.

[89] TA, V.-T., ELMOATAZ, A., and SCHÜPP, S., "Graph-based tools for microscopic cellular image segmentation," *Pattern Recognition*, vol. 42, no. 6, pp. 1113–1125, 2009.

[90] TELEA, A., "An image inpainting technique based on the fast marching method," *Journal of Graphics, GPU and Game Tools*, vol. 9, no. 1, pp. 23–34, 2004.

[91] TELEA, A. and VAN WIJK, J. J., "An augmented fast marching method for computing skeletons and centerlines," in *Proceedings of the symposium on Data Visualisation 2002*, VISSYM '02, (Aire-la-Ville, Switzerland, Switzerland), pp. 251–ff, Eurographics Association, 2002.

[92] TSAI, A., YEZZI, A., J., WELLS, W., TEMPANY, C., TUCKER, D., FAN, A., GRIMSON, W., and WILLSKY, A., "A shape-based approach to the segmentation of medical imagery using level sets," *Medical Imaging, IEEE Transactions on*, vol. 22, pp. 137 –154, feb. 2003.

[93] TSAI, A., YEZZI, A., J., and WILLSKY, A., "Curve evolution implementation of the mumford-shah functional for image segmentation, denoising, interpolation, and

magnification," *Image Processing, IEEE Transactions on*, vol. 10, pp. 1169 –1186, aug 2001.

[94] TSAI, Y., KAUL, V., and YEZZI, A. J. in *Automating the Crack Map Detection Process for Machine Operated Crack Sealer*, vol. 90th Annual Meeting, 2011.

[95] TSITSIKLIS, J., "Efficient algorithms for globally optimal trajectories," in *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, vol. 2, pp. 1368 –1373 vol.2, dec 1994.

[96] WANG, J. and ADELSON, E., "Representing moving images with layers," *Image Processing, IEEE Transactions on*, vol. 3, pp. 625 –638, sep 1994.

[97] WEBER, O., DEVIR, Y. S., BRONSTEIN, A. M., BRONSTEIN, M. M., and KIMMEL, R., "Parallel algorithms for approximation of distance maps on parametric surfaces," *ACM transactions on graphics*, vol. 27, no. 4, 2008.

[98] YATZIV, L., BARTESAGHI, A., and SAPIRO, G., "O(N) Implementation of the Fast Marching Algorithm," *Journal of Computational Physics*, vol. 212, pp. 393–399, 2005.

[99] YEZZI, A. J., TSAI, A., and WILLSKY, A., "A fully global approach to image segmentation via coupled curve evolution equations," *Journal of Visual Communication and Image Representation*, vol. 13, pp. 195–216, 2002.

[100] YEZZI, A.J., J. and PRINCE, J., "An eulerian pde approach for computing tissue thickness," *Medical Imaging, IEEE Transactions on*, vol. 22, pp. 1332 –1339, oct. 2003.

[101] ZHAO, H., "A fast sweeping method for eikonal equations," *Mathematics of Computation*, vol. 74, pp. 603–627, 2005.

[102] ZHAO, H.-K., OSHER, S., and FEDKIW, R., “Fast surface reconstruction using the level set method,” in *Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on*, pp. 194 –201, 2001.

[103] ZHAO, Z., AYLWARD, S., and TEOH, E., “A novel 3D partitioned active shape model for segmentation of brain MR images,” *MICCAI 2005*, pp. 221–228, 2005.

[104] ZHU, S. C. and YUILLE, A., “Region competition: unifying snakes, region growing, and bayes/mdl for multiband image segmentation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, pp. 884 –900, sep 1996.

VITA

Vikram Appia was born in Madras (Chennai), India. He received his schooling in various cities in India and his Bachelor's in Electronics and Communications Engineering from Nirma Institute of Technology, Ahmedabad, India in 2006. He completed his Master's Thesis under the guidance of Dr. Russell Mersereau and received his Master's degree in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta in 2008. He completed his PhD thesis under the guidance of Dr. Anthony Yezzi and received his PhD degree in Electrical and Computer Engineering in 2012. His research interests are primarily in the fields of computer vision and image processing with a special focus in the areas of Image Segmentation, Active Contour Models, Prior-based Segmentation Models, Optical Flow/Motion Segmentation, PDE-based Methods and Level set based Methods.