

Multiscale Active Contour Methods in Computer Vision with Applications in Tomography

A Dissertation
Presented to
The Academic Faculty

by

Christopher V. Alvino

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
April 2005

Multiscale Active Contour Methods in Computer Vision with Applications in Tomography

Approved by:

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

Dr. Monson H. Hayes III
School of Electrical
and Computer Engineering
Georgia Institute of Technology

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

Dr. Greg Turk
College of Computing
Georgia Institute of Technology

Dr. Magnus Egerstedt
School of Electrical
and Computer Engineering
Georgia Institute of Technology

Date Approved: April 7, 2005

Dedicated to:
My parents, Vincent and Janice Alvino,
for dedicating their lives to us.

ACKNOWLEDGEMENTS

I've been extremely fortunate to have had excellent teachers and mentors throughout my entire education. First and foremost, I would like to thank my advisor, Tony Yezzi. I am extremely grateful for his unique combination of creativity, patience, and selflessness.

Thanks to Professors Tannenbaum and Egerstedt, for giving me advice and encouragement, as well as for being on my reading committee.

Thanks to my committee members: Professors Hayes and Turk, each of whom I also had the pleasure of having courses with while at Georgia Tech.

Thanks to my Masters advisor, Professor James Flanagan, for encouraging me to go to Georgia Tech. I am particularly indebted to Lucas Parra for his collaboration and friendship.

I'd like to acknowledge the Department of Medical Imaging and Radiation Sciences, Monash University, Victoria, Australia for the brain CT image. Also, I'd like to acknowledge Tracy Faber at Emory for providing real and simulated projection data.

Thanks to all my colleagues in the lab for making the days pleasant *and* enriching: Jeremy Jackson, Siddharth Manay, Ganesh Sundaramoorthi, Gözde Ünal, Kelvin Rocha, Miguel Lopez, Hua “Carol” Li, and Namrata Vaswani. Thanks to the members of Allen Tannebaum's lab past and present, particularly Eric Pichon, Marc Niethammer, Cecilia Curry, Andrew Stein, Delphine Nain, Jincheol Ha, Gallagher Pryor, and Yan Yang.

At the risk of leaving out certain friends, I'd like to particularly thank: Rob Emanuele, A.J. Mihalic, Vahe Hagopian, Jay Silver, Rick Shilling, Todd Eddie, Ellene Smith, Volkan Cevher, and Kate Elmore.

Special thanks to Avery Morrison for being the only person to give me a good reason to quickly graduate from Georgia Tech. Thanks to the entire Morrison family for making me feel at home away from home. Thanks to Ed Morrison for being extremely generous with computer hardware in times of need.

Most of all, thanks to my parents, Vince and Janice, and my brothers, Brian and Nick.

TABLE OF CONTENTS

DEDICATION	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	ix
LIST OF FIGURES	x
SUMMARY	xiv
I INTRODUCTION	1
1.1 Motivation	1
1.2 Specific Contributions	6
1.2.1 Tomography Reconstruction	6
1.2.2 Multigrid Computation of Non-linear Rotationally Invariant Optical Flow	6
1.2.3 Rigid Contour Registration using Minimal Surfaces	6
1.2.4 Fast Mumford-Shah Segmentation using Reduced Basis	7
II VARIATIONAL TOMOGRAPHY FRAMEWORK	8
2.1 Two-Dimensional Tomography Model	10
2.2 Proposed Reconstruction Method	10
2.2.1 Piecewise Smooth Cost Functional	11
2.2.2 Evolution Equations	12
2.2.3 Three-Dimensional Evolution	14
2.3 Implementation	16
2.3.1 Image Intensity Evolution	17
2.3.2 Contour/Surface Evolution	17
2.3.3 Algorithmic Implementation	19
2.3.4 Weighting Parameters	19
2.4 Summary	20
III MULTISCALE TOMOGRAPHY COMPUTATION	23
3.1 Introduction	23
3.2 Evolution Equations	24

3.2.1	Image Intensity Evolution	25
3.2.2	Gradient Descent	25
3.2.3	Multiscale Method	26
3.3	Computational Cost	28
IV	VARIATIONAL REGION INITIALIZATION	30
4.1	Introduction	30
4.2	Tomography Problem Statement	31
4.3	Proposed Initial Segmentation	32
4.3.1	Image Model and Functional	32
4.3.2	Evolution of Reconstruction Parameters	33
4.3.3	Thresholding	35
4.4	Implementation	36
4.5	Simulations	36
4.5.1	Example Segmentation	36
4.5.2	Comparison with Minimum Norm	37
V	EXPERIMENTAL VALIDATION	41
5.1	Two-Dimensional Reconstructions	41
5.1.1	Bone Reconstruction	41
5.1.2	Brain Reconstruction	42
5.1.3	Method Comparison	44
5.2	Three-Dimensional Reconstructions	48
5.2.1	Modified Cardiac Torso Phantom	48
5.2.2	Multiscale Reconstructions	48
5.2.3	Variational Surface Initialization	49
5.2.4	Heart Perfusion Detection	49
VI	MULTIGRID COMPUTATION OF NON-LINEAR OPTICAL FLOW	59
6.1	Introduction	59
6.2	Optical Flow Functional	60
6.2.1	Rotational Invariance	62
6.3	Computation	63

6.3.1	Gradient Descent	63
6.3.2	Multigrid	64
6.4	Simulations	65
VII	MINIMAL SURFACES FOR CONTOUR REGISTRATION	68
7.1	Introduction	68
7.1.1	Proposed Work	68
7.1.2	Past Work in Registration	68
7.2	Method	69
7.2.1	Initialization	69
7.2.2	Evolution to Minimal Surface	70
7.2.3	Gradient Evolution for Rigid Registration	71
7.2.4	Motivation	73
7.3	Scale Space	77
7.3.1	Scale Space Parameter	77
7.3.2	Extreme Cases	77
7.3.3	Local Minima Decreasing	78
7.4	Implementation	79
7.4.1	Level Set Representation of Surface	80
7.4.2	Freezing of Top and Bottom Contours	81
7.4.3	PDE Implementation	81
7.5	Conclusion	83
VIII	FAST MUMFORD-SHAH SEGMENTATION USING IMAGE SCALE SPACE BASES	85
8.1	Introduction	85
8.2	Theory	86
8.2.1	Mumford-Shah Model	86
8.2.2	Proposed Function Model	88
8.2.3	Choice of Basis	89
8.3	Implementation	90
8.3.1	Jacobi Mumford-Shah	91
8.3.2	Multigrid Mumford-Shah	91

8.3.3	Reduced Basis Implementation	91
8.3.4	Contour Evolution	92
8.3.5	Multiregion Labeling	93
8.4	Experiments	93
8.4.1	LHE vs. GHE Comparison	94
8.4.2	Two Region vs. Multi-Region Results	94
8.4.3	Segmentation in Noise	96
8.4.4	Computational Improvement	96
8.4.5	Brain CT Segmentation	98
8.5	Conclusion	99
IX	CONCLUSION	100
APPENDIX A	— EVOLUTION EQUATION DERIVATION	102
APPENDIX B	— SUFFICIENCY PROOF OF CONVERGENT TIME STEP	105
APPENDIX C	— EXISTENCE CRITERIA FOR MINIMAL SURFACES	110
APPENDIX D	— FLOOD FILL LABELING ALGORITHM	113
REFERENCES	116
VITA	122

LIST OF TABLES

1	Computational performance of algorithms for various images.	29
2	Mean squared error comparison of various 2D reconstruction algorithms. .	47
3	Computational performance of proposed method compared with Multigrid Mumford Shah (MG Mum-Shah), Jacobi Mumford Shah with various number of function evolution iterations, K , per contour evolution, and proposed technique.	98

LIST OF FIGURES

1	Local minimum in piecewise smooth segmentation of simple image. Evolution of contour from left to right starting from the initial image and initial contour.	3
2	Top row: Local minimum in segmentation of reconstructed tomography image of a cube in three dimensions. Bottom row: Global minimum in segmentation of reconstructed cube using multiscale method.	4
3	Illustration of two-dimensional Radon transform. Dotted lines illustrate several lines of constant s , along which Radon transform integrates values of the density function, f to produce projections, $p(s, \theta_i)$	11
4	Illustration of two cases to show effect of projection matching term, J_1 , to the cost functional. Top: higher value of J_1 . Bottom: lower value of J_1 . . .	21
5	Block diagram depicting implementation of tomographic reconstruction evolution equations.	22
6	Illustration of two scale method for solving for image density functions. . .	28
7	Illustration of multiscale method for solving image density functions. . . .	29
8	Integrand of the second term in Eq. (43) as a function of $u(\mathbf{x})$. This term penalizes $u(\mathbf{x})$ for deviating from 0 or 1.	33
9	True synthetic piecewise smooth image of three ellipses (left). Vertical and horizontal projections of image (center and right, respectively).	37
10	Proposed $u(\mathbf{x})$ function (top row) and segmentations (bottom row) of image shown in Fig. 9. From left to right, the columns represent 2, 5, 9 and 15 equally spaced projections respectively.	38
11	Reconstruction of image in Fig. 9 using piecewise smooth image model when initialized with proposed initial segmentation. From left to right is shown the piecewise smooth algorithm at 200, 500, 700, and 1000 iterations respectively. This reconstruction uses 9 equally spaced projections.	38
12	True lung CT image (left). Vertical and horizontal projections of image (center and right, respectively). Image size is 100 by 100 pixels.	39
13	Comparison of segmentations of the image in Fig. 12. Comparison is between proposed method (top row) with segmentations from thresholded minimum norm reconstruction (bottom row). From left to right, the columns represent different segmentation thresholds.	40
14	Evolution of tomographic reconstruction algorithm when initialized with the proposed method. From left to right: initial image, image after 50 reconstruction iterations, after 100 iterations, and after 200 iterations.	40
15	True 71×71-pixel bone image.	41

16	Several stages of bone reconstruction from 9 equally-spaced projections using proposed method. Algorithm initialization in top left image. Evolution proceeds from left to right and is continued in the second row. Final reconstruction on bottom right.	42
17	Comparison of bone reconstruction methods from 9 equally-spaced projections. True image (top left), filtered back projection reconstruction (FBP) (top right), piecewise constant reconstruction (bottom left), and proposed piecewise smooth reconstruction (bottom right).	43
18	True 145×145-pixel brain CT image (courtesy of the Department of Medical Imaging and Radiation Sciences at Monash University.)	44
19	Several stages of brain reconstruction from 25 equally-spaced projections using proposed method. Algorithm initialization in top left image. Evolution proceeds from left to right and is continued in the second row. Final reconstruction on bottom right.	45
20	Comparison of brain reconstruction methods from 25 equally-spaced projections. True image (top left), filtered back projection reconstruction (FBP) (top right), piecewise constant reconstruction (bottom left), and proposed piecewise smooth reconstruction (bottom right).	46
21	Comparison of reconstructed brain image for different reconstruction methods for varying number of projections. First row: Filtered back projection reconstruction. Second row: Algebraic reconstruction technique. Third row: Sinogram restoration. Fourth row: proposed method with fixed weighting parameters. Fifth row: modified proposed method with function smoothness deemphasized once an accurate segmentation is obtained.	51
22	Several slices of modified NCAT phantom showing cross sections of heart and other simulated organs.	52
23	Three view angles of a segmentation surface of the known phantom image using a binary segmentation method. Top image: overhead view of segmentation surface. Middle image: side view. Bottom image: underside view.	53
24	Evolution of surface for simulated three-dimensional cube with gradient descent technique. The top left image shows the initialization of the surface. The evolution of the surface progresses from left to right and from left to right in the second row. Note the suboptimal local minimum on the top and around the sides of the cube.	54
25	Evolution of surface for simulated three-dimensional cube with proposed multiscale method. The top left image shows the initialization of the surface. The middle and right images in the top row show the surface after 50 and 100 iterations, respectively. The left, middle, and right images in the bottom row show the surface after 200, 300, and 600 iterations respectively. Note the lack of local minima on the surface of the cube.	55

26	Evolution of the surface starting from bimodal surface initialization method. The left image is the initialization, the middle and right images are the surface after 10 and 40 iterations respectively.	55
27	Left column: slice of initial three-dimensional density function using bimodal surface initialization and the corresponding initial surface below it. Middle column: slice of density function during the evolution. Right column: slice of reconstructed density function and its corresponding surface.	56
28	Detection of simulated heart perfusion in a different location using 64 noiseless projections with the proposed reconstruction technique.	56
29	Surface resulting from segmentation produced by the proposed technique when using 64 projections for which noise, attenuation and scatter were added. Bottom right image shows heart perfusion as a hole in the surface. .	57
30	Reconstruction using proposed multiscale method and projections with simulated noise. Left image: slice of the true phantom. Second image: filtered back projection reconstruction. Third image: reconstruction using proposed method. Fourth image: close-up of 3D surface showing heart defect.	58
31	Left: frame from Hamburg Taxi sequence. Right: proposed optical of Hamburg Taxi sequence.	66
32	Functional cost versus computation time for the gradient descent and FAS multigrid algorithms.	67
33	Oblique view of three different stages of surface minimizing evolution for stationary shifted circular contours. Surface created from initialization (left), surface after 125 seconds of surface evolution (center), and final minimal surface (right).	71
34	An example of rigid registration. Initial surface between two unregistered rectangles (left), minimal surface between unregistered rectangles (second from left), surface during rigid registration (second from right), and surface after rigid registration (right).	74
35	An illustration of the smoothing effect of minimal surface on the intermediate contours. Original contour (left), minimal surface connecting this contour with replica of itself (center), middle contour of minimal surface (right). . .	75
36	Contour and registration local minimum. “E”-shaped contour (left), minimal surface connecting contour and translated replica of itself (center), and registration local minimum (right).	76
37	Registration overcoming local minimum by increasing separation distance to 25. Minimal surface connecting contour and translated replica (left), registration overcoming local minimum (center), and final correct registration of top contour (right).	77
38	Contour with features at multiple scales (left) and minimal surface resulting from correct registration of contour with a separation distance of 10 (right).	79

39	Two local minima for contour with multiple scale features. The method can not correct for incorrect initial alignment due to local minima. The separation distances for the surfaces 5 and 10 for the left and right surfaces respectively.	79
40	Number of local minima as a function of the separation distance between the two contours for the contour in Fig. 38. On the right is shown the pinch-off distance, i.e., the separation distance at which the minimal surface fails to exist.	80
41	Peregrine falcon segmentation: For each row, progression from left to right shows segmentation evolution from initial contour to final segmentation. (Top row) Multigrid Mumford-Shah segmentation with pixel-by-pixel basis. (Middle row) Proposed LHE has very similar function regions when using drastically reduced basis. (Bottom row) Proposed GHE produces similar segmentation but function regions preserve image edges better than in top two rows.	95
42	Fruits image: (Top row) Segmented using full pixel-by-pixel basis Mumford-Shah segmentation. (Middle row) Segmented using Proposed GHE method using two distinct regions, foreground and background. Note the lack of accuracy in mean intensity for each function region. (Bottom row) Segmented using Proposed GHE method using multiple regions. Note the ability to represent each different fruit as a separate region with distinct basis weight coefficients.	96
43	Noisy fruit image: (Top row) Initial contour overlayed on noisy image along with final Multigrid Mumford-Shah segmentation of image. (Middle row) Initial contour and final segmentation using two-region proposed GHE method. (Bottom row) Initial contour and final segmentation using multiregion proposed GHE method.	97
44	Segmentation of brain CT image using Multiregion Proposed GHE method.	98
45	Gerschgorin disks showing location of eigenvalues of Laplacian operator matrix, S	107
46	Example labeling for 6×6 pixel grid. Only input to the algorithm is the level set function, Ψ , at each pixel. Output of algorithm is a labeling, L , for each pixel into its appropriate connected region.	114

SUMMARY

Most applications in computer vision suffer from two major difficulties. The first is they are notoriously ridden with sub-optimal local minima. The second is that they typically require high computational cost to be solved robustly. The reason for these two drawbacks is that most problems in computer vision, even when well-defined, typically require finding a solution in a very large high-dimensional space.

It is for these two reasons that multiscale methods are particularly well-suited to problems in computer vision. Multiscale methods, by way of looking at the coarse scale nature of a problem before considering the fine scale nature, often have the ability to avoid sub-optimal local minima and obtain a more globally optimal solution. In addition, multiscale methods typically enjoy reduced computational cost.

This thesis applies novel multiscale active contour methods to several problems in computer vision, especially in simultaneous segmentation and reconstruction of tomography images. In addition, novel multiscale methods are applied to contour registration using minimal surfaces and to the computation of non-linear rotationally invariant optical flow. Finally, a methodology for fast robust image segmentation is presented that relies on a lower dimensional image basis derived from an image scale space.

The specific advantages of using multiscale methods in each of these problems is highlighted in the various simulations throughout the thesis, particularly their ability to avoid sub-optimal local minima and their ability to solve the problems at a lower overall computational cost.

CHAPTER I

INTRODUCTION

1.1 Motivation

Most applications in computer vision suffer from two major difficulties. The first is they are notoriously ridden with sub-optimal local minima [19, 37, 38, 75]. The second is that they typically require high computational cost to be solved robustly. The reason for these two drawbacks is that, even when well-defined, these applications typically require finding the minimizer of a complicated cost functional in a very high-dimensional space. In this thesis we introduce novel multiscale active contour and active surface methods to avoid sub-optimal local minima and to reduce computational cost of various computer vision problems.

For instance, the problem of image segmentation can be defined as distinguishing objects from their background [7]. The discrete version of the problem can be seen as labelling each pixel as either foreground or background. Thus, the space of possible segmentations has as many dimensions as there are pixels in the image. Searching over all such segmentations is an intractable problem even for small images. Therefore, it is necessary to use heuristic techniques for segmenting an image.

Simple point-based methods for segmentation, such as thresholding, are not sufficiently robust since they are extremely sensitive to the noise in an image, and since they rely on unrealistic assumptions about the difference between foreground and background intensity. Furthermore, they use no knowledge of geometric properties, such as the object's shape or connectedness [21, 57]. Edge detection methods that are based on thresholding a local statistic of the image, such as those using the Laplacian of Gaussian (LoG) operator, often have convenient mathematical properties and efficient implementation. In addition, edge detection with the LoG produces edges that are closed contours. However, since they involve the second derivative of the image, they are extremely sensitive to noise [45]. Furthermore,

it is difficult to incorporate prior shape information into these techniques, especially in regions where contour lines are not connected when some part of an object’s boundary has weak, noisy, or blurred edges.

Snakes, or active contours¹, introduced by Kass et al. [39, 69], proved to be a sufficiently powerful and robust framework for object segmentation since they model the object geometry directly. Active contour models are attractive since they are able to incorporate prior knowledge about the geometric nature of the objects to be segmented, such as shape, size, or connectedness. Many works have appeared based on geometric models for active contours [15, 16, 44, 73] including edge-based models [39, 44] and region-based models [37, 38, 59, 76, 80]. Mumford and Shah presented a variational model for representing piecewise smooth functions, thus simultaneously representing the boundaries of an object within an image and the smoothness of the functions within the boundaries [48].

Much of the early work done in active contours was done using parametric representations of the contours or surfaces. Level set methods, first presented by Osher and Sethian, gave a flexible way to represent the contours or surfaces of interest implicitly without regard to parameterization, and without the encumbrance of explicitly handling topological changes [17, 51]. This thesis will concentrate on active contour methods for solving problems in computer vision using the level set framework, specifically in tomographic reconstruction.

While active contours have a number of attractive features, one of which is avoiding sub-optimal local minima by directly modeling object geometry, local minima still appear frequently using active contour methods, largely due to their variational formulation and the way in which they are implemented, i.e., as gradient flows. Figure 1 illustrates an example of a segmentation local minimum that occurs due to the choice of contour initialization using the active contour segmentation technique proposed in Chapter 8. This local minimum is avoidable by starting with a different contour initialization, but often it is difficult to know *a priori* what initialization to use. Chapter 4 addresses how to choose initial contours for the application of simultaneous segmentation and reconstruction of tomography images.

¹Active surfaces are the generalization of active contours for three-dimensional images.

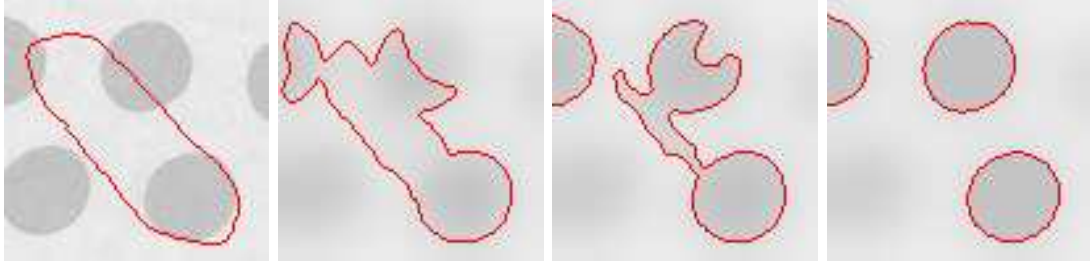


Figure 1: Local minimum in piecewise smooth segmentation of simple image. Evolution of contour from left to right starting from the initial image and initial contour.

As the images get larger, more complex, have higher dimensionality, and more complicated construction, more local minima appear due to the increase in complexity of the cost functional. Furthermore, the way a coupled gradient flow² is numerically implemented often affects the solution, since the evolution of the contour often depends on the simultaneous evolution of some other function. The top row of Fig. 2 illustrates a local minimum that occurs while segmenting a cube using a non-multiscale version of the active surface method for reconstructing tomography images presented in Chapter 2. Evolution of the surface progresses from left to right starting from the initial surface on the left and ending in the cube with inverted surface on the right. Such local minima are often avoidable by computing the coupled gradient flow more accurately using multiscale methods. The bottom row of Fig. 2 illustrates the global minimum obtained when using the multiscale technique described in Chapter 3. In this thesis, we present various novel multiscale active contour methods in computer vision to overcome local minima in various problems such as simultaneous Mumford-Shah segmentation and reconstruction of tomographic images, Mumford-Shah segmentation of two-dimensional images, and rigid registration of two-dimensional contours.

Variational active contour methods in computer vision, although they reduce the search space by incorporating geometric properties of the object, pay a price for their robustness, i.e., they typically require high computational cost. One way of reducing this computational

²A coupled gradient flow refers to the simultaneous evolution of a contour and a function on the image domain, specifically where the evolution of the contour depends on the value of the function and the evolution of the function depends on the location of the contour.

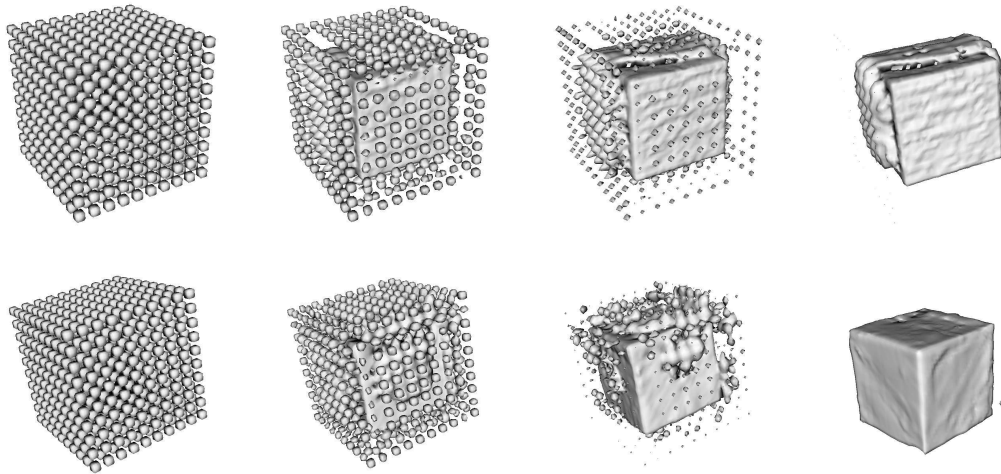


Figure 2: Top row: Local minimum in segmentation of reconstructed tomography image of a cube in three dimensions. Bottom row: Global minimum in segmentation of reconstructed cube using multiscale method.

cost is to implement these techniques with multiscale methods that take advantage of the specific nature of the solutions, such as smoothness. For example, the image segmentation model developed by Mumford and Shah has a regularization term that penalizes the estimated image functions for deviating from smoothness. Also, optical flow functionals, although not an active contour method, typically contain a regularization term to ensure smoothness of the resulting vector field³. Multigrid methods are a class of multiscale solution techniques that have been shown to improve the computational performance when solving partial differential equations (PDE's) whose solutions are spatially smooth [11, 12]. Multigrid methods have been used for the computation of optical flow [2, 8, 22], as well as in other computer vision and image processing problems [1, 27, 68] including in tomographic reconstruction [6, 54].

In this thesis we present various multiscale methods for the purpose of computational

³The purpose of this regularization term is to combat the well known aperture problem in which it is impossible to uniquely determine the optical flow field in directions tangent to isobrightness contours when using only the optical flow constraint.

speed improvement, such as in the computation of the image density functions in the proposed Mumford-Shah based tomographic reconstruction technique, in multigrid computation of non-linear rotationally invariant optical flow, and in fast reduced basis image segmentation. The specific advantages of using multiscale methods in each of these problems are highlighted in the various simulations throughout the thesis, particularly their ability to avoid sub-optimal local minima and their ability to solve the problems at a lower overall computational cost.

In order of appearance, Chapter 2 presents the novel extension of the Mumford-Shah framework for image segmentation to simultaneous segmentation and reconstruction of tomographic images. The high computational cost of solving for the image density functions in Chapter 2 along with the smooth nature of the functions that is guaranteed by the model call for a multiscale technique for their solution. Chapter 3 presents a novel multiscale technique for their solution and reports the computational improvement. An important part of using an active contour model for segmentation is choosing an appropriate contour initialization. The appropriate contour initialization can greatly reduce both convergence time and the presence of sub-optimal local minima in the segmentation. Chapter 4 presents a novel variational technique to produce a fuzzy bimodal segmentation that yields an initial contour. Chapter 5 shows validation of the techniques presented in the previous three chapters, illustrating the power of the proposed tomographic reconstruction technique, the ability of the multiscale technique for reducing computational cost and avoiding local minima, and the improved accuracy of the reconstructions when initializing the contour appropriately.

Chapter 6 presents the multigrid computation of a non-linear rotationally invariant edge-preserving optical flow. Chapter 7 presents a novel multiscale technique for rigid two dimensional contour registration that avoids local minima by using scale space properties of minimal surfaces. Chapter 8 presents a novel method for a fast approximation to Mumford-Shah segmentation using a reduced image basis whose components are part of a scale space of diffused images.

1.2 Specific Contributions

1.2.1 Tomography Reconstruction

- Formulation of cost functional and coupled gradient flow for simultaneous segmentation and reconstruction of piecewise smooth 2D and 3D tomography images.
- Implementation of proposed tomography reconstruction technique in 2D and 3D.
- Analysis of sufficient convergent forward Euler step size to guaranteed numerical convergence of function evolution for tomographic reconstruction technique.
- Development of a novel multiscale technique for solving partial differential equations resulting from proposed cost functional.
- Implementation of multiscale tomography technique and performance shown on 2D and 3D tomography reconstructions. The ability to avoid suboptimal local minima and reduce computational cost are stressed.
- Development of novel variational method to obtain a binary segmentation and thus an initial contour or surface for tomography reconstruction technique.
- Validation of proposed method by comparison with other reconstruction techniques and by segmentation of simulated heart perfusion abnormalities.

1.2.2 Multigrid Computation of Non-linear Rotationally Invariant Optical Flow

- Formulation of a new optical flow regularizer that preserves edges in optical flow but is also rotationally invariant.
- Analysis that the proposed optical flow is invariant to rotation.
- Multigrid implementation of the proposed optical flow and illustration of reduced computational cost.

1.2.3 Rigid Contour Registration using Minimal Surfaces

- Development of a scale space technique for rigid registration of 2D contours using minimal surfaces.

- Analysis of sufficient condition for existence of a minimal surface using the proposed technique.
- Implementation of this technique using level set surface evolution.
- Validation on contours that shows the ability of the technique to overcome local minima present in set-symmetric difference registration.

1.2.4 Fast Mumford-Shah Segmentation using Reduced Basis

- Development of reduced basis Mumford-Shah segmentation.
- Implementation of proposed reduced basis method and illustration of reduced computational cost while maintaining very similar performance.
- Implementation of a multi-region labeling scheme to enable more than two regions with the proposed reduced basis Mumford-Shah segmentation method.

CHAPTER II

VARIATIONAL TOMOGRAPHY FRAMEWORK

In transmission tomography, the goal is to reconstruct an image from its measured projections as modeled by the Radon transform. While it is possible to invert the Radon transform in theory [30], in practice it requires a large amount of projection data to obtain a high-quality reconstruction. For a review of tomographic reconstruction methods, we refer the reader to [55]. Constraints in cost, scan time, and image accessibility limit the number and quality of projections. Therefore, it is often not feasible in practice to obtain the amount of projections necessary to reconstruct accurately. For this reason, several authors have considered solving the under-constrained or *limited-angle* tomographic reconstruction problem [35, 63, 56, 54, 58].

Additional assumptions must be made about the image function to properly constrain the possible solutions. Authors have attempted different constraints depending on the assumptions they wish to make about the unknown images. Sezan and Stark use projection onto convex sets (POCS) to incorporate priors [63]. Reeds and Shepp use a technique called “squashing” to convert limited angle problems into problems where all of the projections are known [56, 50]. Prince and Willsky introduce a technique for sinogram reconstruction by ensuring that the finest-grain sinogram is smooth [54].

In [4] we proposed tomographic reconstruction using a class of image priors that were addressed by Mumford and Shah’s pioneering work on region-based image segmentation [48]. That is, we considered the class of piecewise constant or, more generally, piecewise smooth images. Mumford and Shah considered segmenting images by assuming piecewise smooth regions separated by deformable or “active” contours, across which smoothness is not required. Chan and Vese [17] incorporated these ideas into the level set framework introduced by Osher and Sethian [51], allowing the contours to naturally handle topology changes. Active contour models are attractive because they are both well-principled and

robust. Since Mumford and Shah’s work, several ideas have emerged to make use of active contour models. [16, 39, 80].

In tomographic reconstruction, three major works have been presented that make use of active contour models. Yu and Fessler presented an early discrete approach to solve this problem [77]. They defined a discrete cost functional to model the reconstruction by an image partitioned into two regions separated by a deformable discrete boundary analogous to the contour proposed by Mumford and Shah. However, Yu’s method for maintaining a smooth surface depends on the accuracy of the image estimate. This accuracy can not be guaranteed upon the initialization of the algorithm. In fact, they ignore this term in their implementation.

Bruandet et al. have proposed a surface evolution technique to model the reconstructed image as two regions, one with density zero, and the other with an *a priori* known density, μ [14]. Whitaker and Elangovan have proposed a similar, although slightly more general, piecewise constant model where the two region densities are adaptive [72]. The models of Bruandet and Whitaker are adequate when the regions being imaged are known to be homogeneous in density, but this may not be the case in practice. Images often have smoothly varying densities, and often have three or more distinct regions. For this reason, it is desirable to generalize the piecewise constant models of Bruandet and Whitaker. By allowing the intensity of each pixel to vary independently, while enforcing that within regions, nearby pixels have similar intensities, the reconstructed images are successfully constrained to be piecewise smooth.

In this chapter, a deformable surface model is defined to constrain tomographic images to be piecewise smooth. A cost functional is proposed that attains a high value when the projections of the estimated reconstruction deviate from the measured projections, and attains a high value when the reconstructed image deviates from piecewise smoothness. By deforming the image model parameters to minimize this functional, the reconstructed image matches the measured projections and is piecewise smooth.

2.1 *Two-Dimensional Tomography Model*

Although easily generalized to higher dimensions, the reconstruction problem will be described in two dimensions, i.e., with two-dimensional image functions and one-dimensional projections. The reconstruction technique developed holds for the three-dimensional problem. Displaying the mathematical explanation and simulation results in two dimensions allows for a clearer explanation of the technique. The three-dimensional method is then a straightforward generalization of the two-dimensional method.

Unknown image density function, $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, will, via the Radon transform, create projections,

$$p(s, \theta) = \int_{\Omega} f(x, y) \delta(P_{\theta}(x, y) - s) d\bar{x}, \quad (1)$$

where $P_{\theta}(x, y) = x \cos \theta + y \sin \theta$, and $d\bar{x}$ stands for integration in the spatial variables x and y . Figure 3 illustrates the Radon transform of a two-dimensional density function, f , onto one-dimensional projections $p(s, \theta_1)$ and $p(s, \theta_2)$. The dotted line show lines of constant s for a specific function projection. The value of the point $p(s, \theta)$ can be interpreted as the line integral of the density function through the line at angle θ and corresponding to value s .

The tomographic reconstruction problem is that of obtaining an estimate of f , which we will denote \hat{f} , from P projections $p(s, \theta_1), \dots, p(s, \theta_P)$.

2.2 *Proposed Reconstruction Method*

In order to determine the shape of the contour that separates the piecewise smooth regions, is it possible to reconstruct \hat{f} by imposing a cost on the discrepancy between the projections of the estimated image and the measured projections. However, since the problem is under-constrained, imposing this cost is not enough. Many images that are not piecewise smooth will also have projections that match the measured projections.

Therefore, a cost is imposed on the deviation of the reconstructed image from piecewise smoothness. Properly enforcing piecewise smoothness requires penalizing two separate undesirable features of the model. Piecewise smoothness implies that large gradients in image intensity within regions should be penalized. In addition, total contour length should also

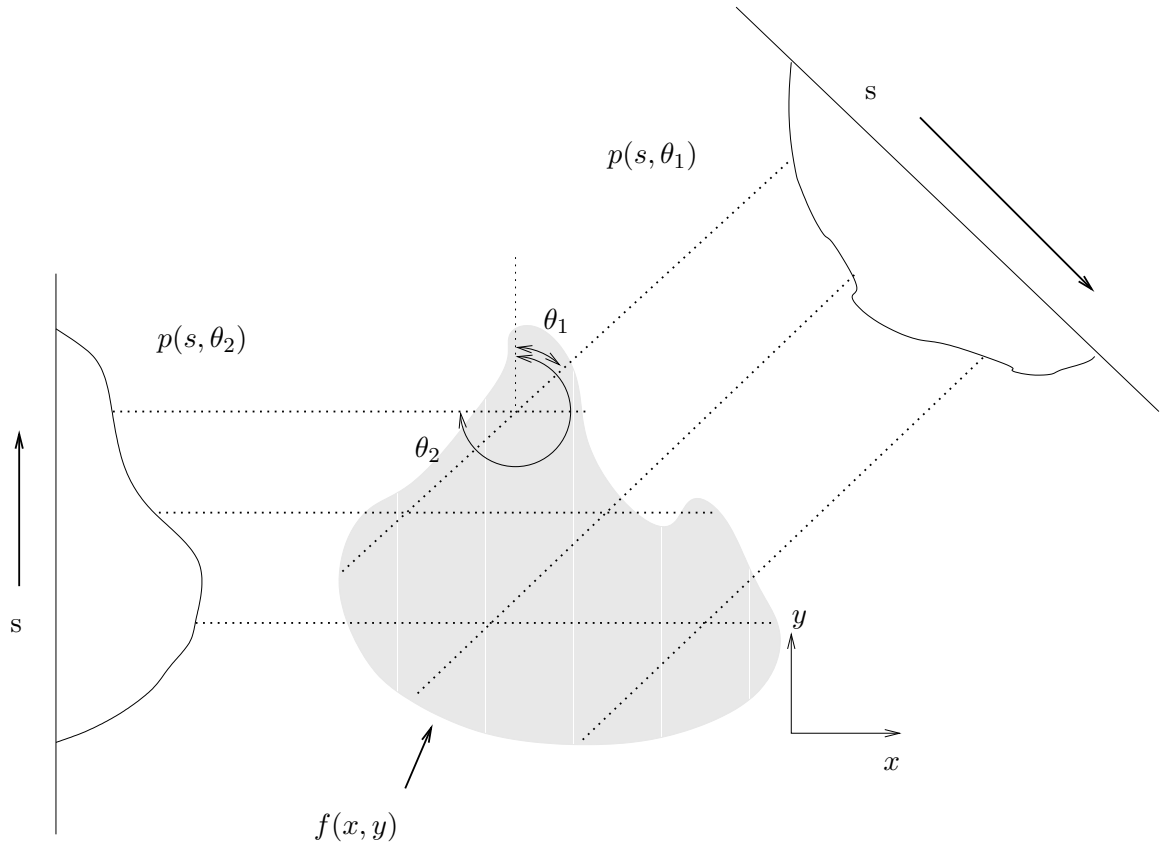


Figure 3: Illustration of two-dimensional Radon transform. Dotted lines illustrate several lines of constant s , along which Radon transform integrates values of the density function, f to produce projections, $p(s, \theta_i)$.

be penalized, for otherwise, the contour could converge to an irregular shape to match image noise.

2.2.1 Piecewise Smooth Cost Functional

A cost functional is proposed that has three terms: the projection matching term, J_1 ; the function smoothness term, J_2 ; and the contour length term J_3 . We combine these terms

with weighting parameters to obtain the overall cost functional $J = \alpha J_1 + \beta J_2 + \gamma J_3$, where,

$$J_1 = \sum_{\theta \in \Theta} \int_0^{L_\theta} (p(s, \theta) - \hat{p}(s, \theta))^2 ds, \quad (2)$$

$$J_2 = \int_{R_1} \|\nabla \hat{f}_{R_1}(x, y)\|^2 d\bar{x} + \int_{R_2} \|\nabla \hat{f}_{R_2}(x, y)\|^2 d\bar{x}, \quad (3)$$

$$J_3 = \int_0^L d\ell. \quad (4)$$

Note that $\Theta = \{\theta_1, \dots, \theta_P\}$ is the set of projection angles, L_θ is the length of the projection at angle θ . Also, $\hat{p}(s, \theta)$ is the Radon transform of the image estimate \hat{f} . The symbol ∇ is the standard gradient operator.

In J_2 , note that the function \hat{f} has been split into two open regions, R_1 and R_2 , that are subsets of the image domain. These are regions where \hat{f} is guaranteed to be differentiable. Also, \hat{f}_{R_i} will denote the restriction of \hat{f} to region R_i . The reason for this function restriction is that the piecewise smooth model can not guarantee differentiability on the contour, $\partial R = \partial R_1 = \partial R_2$, that separates the two regions. Thus, any attempt to differentiate on the boundary, ∂R , would be poorly defined. Finally, L is the length of the contour and $d\ell$ is the arc length element of the contour.

Figure 4 illustrates two cases that highlight the effect of the projection matching term, J_1 . The top illustration shows an estimated density function, \hat{f} and how its corresponding projections, \hat{p} are not close to the measured projections p . The bottom illustration shows an estimated density function whose projections more closely match the measured projections. Thus, the top illustration will produce a higher value of J_1 and thus a less desirable reconstruction than the bottom illustration. Note that this comparison must be performed by way of the projections of f and not directly f and \hat{f} since f is unknown *a priori*.

2.2.2 Evolution Equations

By introducing an evolution time parameter, t , into the image intensity functions and into the contour, thus augmenting $\hat{f}_{R_i}(x, y)$ to $\hat{f}(x, y, t)$ and $\vec{C}(s)$ to $\vec{C}(s, t)$, the cost functional is dependent on t . Differentiating the three terms in the cost functional, J , with respect to

t yields,

$$\begin{aligned} \frac{dJ_1}{dt} &= \int_0^L \left\langle \vec{C}_t(\ell), \left(\hat{f}_{R_1}(\vec{C}(\ell)) - \hat{f}_{R_2}(\vec{C}(\ell)) \right) \times \right. \\ &\quad \left. \sum_{\theta \in \Theta} (\hat{p} - p) (P_\theta(\vec{C}(\ell)), \theta) \vec{\nu}_{R_1}(\ell) \right\rangle d\ell \\ &\quad + \int_{R_1} \hat{f}_{R_1,t}(x, y) \sum_{\theta \in \Theta} (\hat{p} - p) (P_\theta(x, y), \theta) d\bar{x} \\ &\quad + \int_{R_2} \hat{f}_{R_2,t}(x, y) \sum_{\theta \in \Theta} (\hat{p} - p) (P_\theta(x, y), \theta) d\bar{x} \end{aligned} \quad (5)$$

$$\begin{aligned} \frac{dJ_2}{dt} &= \int_0^L \left\langle \vec{C}_t(\ell), \left(\|\nabla \hat{f}_{R_1}(\vec{C}(\ell))\|^2 - \|\nabla \hat{f}_{R_2}(\vec{C}(\ell))\|^2 \right) \vec{\nu}_{R_1}(\ell) \right\rangle d\ell \\ &\quad - \int_{R_1} \hat{f}_{R_1,t}(x, y) \Delta \hat{f}_{R_1}(x, y) d\bar{x} \\ &\quad - \int_{R_2} \hat{f}_{R_2,t}(x, y) \Delta \hat{f}_{R_2}(x, y) d\bar{x}, \end{aligned} \quad (6)$$

$$\frac{dJ_3}{dt} = - \int_0^L \left\langle \vec{C}_t(\ell), \kappa(\ell) \vec{\nu}_{R_1}(\ell) \right\rangle d\ell \quad (7)$$

where \vec{C}_t is the partial derivative of the contour with respect to time, $\langle \cdot \rangle$ denotes the two-dimensional Euclidean dot product, Δ represents the Laplacian operator, and κ is the curvature of the contour. Also, $\hat{f}_{R_1,t}$ and $\hat{f}_{R_2,t}$ are partial derivative of the two image functions with respect to time. The unit normal vector at each point in the contour that points from region R_1 to region R_2 is denoted as $\vec{\nu}_{R_1}$. Also, note that $(\hat{p} - p)(P_\theta(x, y), \theta)$ is shorthand for $\hat{p}(P_\theta(x, y), \theta) - p(P_\theta(x, y), \theta)$. Furthermore, note that \hat{f}_{R_i} and all its various forms of spatial and temporal derivatives have the arguments (x, y) unless they appear in an inner product with the term \vec{C}_t , in which case they have the argument $(\vec{C}(\ell))$. This is consistent with the variables of integration for each term.

The detailed derivations of Eqs. (5), (6), and (7) make use of the divergence theorem, and are included in Appendix A. Note that a factor of two is left out because it appears in all three equations. In the evolution implementation, this is equivalent to evolving the contours and image functions at half the speed and has no effect on the solution.

By evolving the contour and image intensity functions to make Eqs. (5), (6), and (7) negative, the cost functional, J , is guaranteed to be decreasing in evolution time, t . This is achieved by evolving both the contour and image intensity functions in the directions

opposite to the terms by which they are multiplied in Eqs. (5), (6), and (7). For instance, Eq. (5) reveals that by setting evolution of the contour to,

$$\vec{C}_t(\ell) = - \left(\hat{f}_{R_1}(\vec{C}(\ell)) - \hat{f}_{R_2}(\vec{C}(\ell)) \right) \sum_{\theta \in \Theta} (\hat{p} - p) (P_\theta(\vec{C}(\ell)), \theta) \vec{\nu}_{R_1}(\ell), \quad (8)$$

that it is guaranteed that the first term in Eq. (5) will be negative. Hence, this term in Eq. (5) will then contribute to a decrease in total energy. In this manner, and by the linearity of differentiation, the evolution equations that will minimize the proposed cost functional are,

$$\begin{aligned} \vec{C}_t(\ell) = & -\alpha \left(\hat{f}_{R_1}(\vec{C}(\ell)) - \hat{f}_{R_2}(\vec{C}(\ell)) \right) \sum_{\theta \in \Theta} (\hat{p} - p) (P_\theta(\vec{C}(\ell)), \theta) \vec{\nu}_{R_1}(\ell) \\ & -\beta \left(\|\nabla \hat{f}_{R_1}(\vec{C}(\ell))\|^2 - \|\nabla \hat{f}_{R_2}(\vec{C}(\ell))\|^2 \right) \vec{\nu}_{R_1}(\ell) \\ & + \gamma \kappa(\vec{C}(\ell)) \vec{\nu}_{R_1}(\ell) \end{aligned} \quad (9)$$

$$\begin{aligned} \hat{f}_{R_1,t}(x, y) = & -\alpha \sum_{\theta \in \Theta} (\hat{p} - p) (P_\theta(x, y), \theta) \\ & + \beta \Delta \hat{f}_{R_1}(x, y) \end{aligned} \quad (10)$$

$$\begin{aligned} \hat{f}_{R_2,t}(x, y) = & -\alpha \sum_{\theta \in \Theta} (\hat{p} - p) (P_\theta(x, y), \theta) \\ & + \beta \Delta \hat{f}_{R_2}(x, y). \end{aligned} \quad (11)$$

2.2.3 Three-Dimensional Evolution

Although the presented results have been explicitly for two dimensions, the technique is easily generalized to three-dimensional image functions. Much of the discussion remains the same, however, in three dimensions, a surface represents the boundary between regions instead of a contour. The resulting evolution equations for the three-dimensional problem are obvious three-dimensional generalizations of the previously discussed two-dimensional evolution equations. The main difference is that the curvature term in Eq. (7) for the contour embedded in two dimensions becomes mean curvature for the surface in three dimensions.

The three-dimension version of the tomography problem assumes image density function, $f : \mathbb{R}^3 \rightarrow \mathbb{R}$, that will, via the Radon transform, create projections with two spatial

dimensions,

$$p(s, t, \theta, \phi) = \int_{\Omega} f(x, y, z) \delta(P_{\theta, \phi}(x, y, z) - s, Q_{\theta, \phi}(x, y, z) - t) d\bar{x}, \quad (12)$$

where $P_{\theta, \phi}(x, y, z) = \cos \phi(x \cos \theta + y \sin \theta) - z \sin \phi$, $Q_{\theta, \phi} = y \cos \theta - x \sin \theta$ and $d\bar{x}$ stands for integration in the spatial variables x , y and z .

The three-dimensional cost functions are,

$$J_1 = \sum_{i=1}^P \int_{\Xi_i} (p(s, t, \theta_i, \phi_i) - \hat{p}(s, t, \theta_i, \phi_i))^2 d\bar{s}, \quad (13)$$

$$J_2 = \int_{R_1} \|\nabla \hat{f}_{R_1}(x, y)\|^2 d\bar{x} + \int_{R_2} \|\nabla \hat{f}_{R_2}(x, y)\|^2 d\bar{x}, \quad (14)$$

$$J_3 = A, \quad (15)$$

where R_1 and R_2 are now regions in \mathbb{R}^3 and are separated by a surface whose area is A . Also note the region of integration for the integral in J_1 is now Ξ_i , a two-dimensional integral domain in projection variables s and t , whose area element is denoted $d\bar{s}$.

Similarly combining these with weighting parameters yields the overall cost function $J = \alpha J_1 + \beta J_2 + \gamma J_3$. The evolution equations function \hat{f} and for the surface S for the three-dimensional problem are,

$$\vec{S}_t(v) = -\alpha (\hat{f}_{R_1} - \hat{f}_{R_2}) (\vec{S}(v)) \quad (16)$$

$$\begin{aligned} & \times \sum_{i=1}^P (\hat{p} - p) (P_{\theta_i, \phi_i}(\vec{S}(v)), Q_{\theta_i, \phi_i}(\vec{S}(v)), \theta_i, \phi_i) \vec{\nu}_{R_1}(\vec{S}(v)) \\ & - \beta (\|\nabla \hat{f}_{R_1}\|^2 - \|\nabla \hat{f}_{R_2}\|^2) (\vec{S}(v)) \vec{\nu}_{R_1}(\vec{S}(v)) \\ & + \gamma H(\vec{S}(v)) \vec{\nu}_{R_1}(\vec{S}(v)) \end{aligned} \quad (17)$$

$$\begin{aligned} \hat{f}_{R_1, t}(x, y, z) &= -\alpha \sum_{i=1}^P (\hat{p} - p) (P_{\theta_i, \phi_i}(x, y, z), Q_{\theta_i, \phi_i}(x, y, z), \theta_i, \phi_i) \\ & + \beta \Delta \hat{f}_{R_1}(x, y, z) \end{aligned} \quad (18)$$

$$\begin{aligned} \hat{f}_{R_2, t}(x, y, z) &= -\alpha \sum_{i=1}^P (\hat{p} - p) (P_{\theta_i, \phi_i}(x, y, z), Q_{\theta_i, \phi_i}(x, y, z), \theta_i, \phi_i) \\ & + \beta \Delta \hat{f}_{R_2}(x, y, z), \end{aligned}$$

where $\vec{S}(v)$ represents a point on the surface. Note that the three-dimensional evolution equations are nearly identical to the two-dimensional evolution equations with a few small

exceptions. Since the three-dimensional cost function contains surface area instead of contour length, it is necessary to evolve in the direction of the unit normal with speed proportional to the local mean curvature of the surface, $H(\vec{S}(v))$.

2.3 *Implementation*

The following discussion will refer to the evolution equations for the two-dimensional case but also holds for the three-dimensional case. In the three-dimensional case, instances of the word *contour* would be replaced by *surface* with no loss of generality.

Evolution Eqs. (9), (10), and (11) are implemented in a two-step iterative algorithm. The first step of the algorithm is to evolve the image intensity functions by Eqs. (10) and (11) until they are converged. This step is performed while holding the contour fixed. The second step of the algorithm is to evolve the contour by one forward time step of Eq. (9). We iterate these two steps until the contour no longer moves; at this point the algorithm is considered to be converged.

Note that, although the evolution equations are continuous both spatially and temporally (evolution being time), the algorithm has been implemented discretely. Thus, all image derivatives have been approximated by appropriate finite differencing schemes to ensure stability of the algorithm. Likewise, the evolution steps have been approximated by the standard forward Euler scheme.

In addition, it is important to note that, while image derivatives within regions are well-defined, image derivatives do not exist on the boundaries of the image nor on the contour separating the two regions. Neumann boundary conditions have been assumed to handle these cases, i.e., at the boundaries, the directional derivatives of the functions in the directions normal to the boundaries are assumed to be zero.

Note, that the choice of Neumann boundary conditions is not for the purpose of simplification. When solving a equation of the form of Eqs. (10) and (11), a choice of boundary conditions is necessary. For each point on the boundary is it necessary to either specify the value of the function (Dirichlet boundary conditions), or the value of the directional

derivative normal to the boundary (Neumann boundary conditions) [23]. Since it is desirable that the value of the function should be able to vary at each point on the boundary, it is necessary to chose Neumann boundary conditions. Normal derivatives of zero are the most natural choice since it is desired that the estimated density function, \hat{f} , is as smooth as possible.

2.3.1 Image Intensity Evolution

Each of the evolution equations, Eqs. (10) and (11), are implemented by updating the image intensity functions, \hat{f}_{R_1} and \hat{f}_{R_2} , by the forward Euler scheme, i.e.,

$$\hat{f}_{R_1}^{n+1} = \hat{f}_{R_1}^n + \Delta t \hat{f}_{R_1,t}^n \quad (19)$$

$$\hat{f}_{R_2}^{n+1} = \hat{f}_{R_2}^n + \Delta t \hat{f}_{R_2,t}^n, \quad (20)$$

where the superscript n denotes iteration number and Δt is the iteration time step parameter. Note that the evolutions, $\hat{f}_{R_1,t}^n$ and $\hat{f}_{R_2,t}^n$, depend on the reconstruction estimates, \hat{p} , and thus is it required to update the projections each time these updates are performed. Thus, the update equations are guaranteed to minimize the cost functional, thereby ensuring both smoothness of the reconstruction and that the reconstruction projections match the measured projections.

A sufficient time step to ensure convergence of the iterative numerical scheme is,

$$\Delta t < \frac{2}{\alpha NP + 4\beta D}, \quad (21)$$

where N is the maximum length dimension of the image density function, including diagonal lengths, P is the number of projections, and D is the dimensionality of the image function. This result is proved in Appendix B.

2.3.2 Contour/Surface Evolution

Again, this section will include discussion of the *contour* for the two-dimensional model. All statements hold for the surface in the three-dimensional model except when explicitly noted otherwise.

The contour in this model is represented using level set techniques. That is, the contour is the set of points in the image domain where some level set function, $\Psi : \Omega \rightarrow \mathbb{R}$, equals

a chosen constant. Level set techniques implicitly represent the contour and thus have the ability to naturally handle topology changes while avoiding cumbersome and numerically unstable parameterizations.

The level set function is evolved by,

$$\frac{d\Psi}{dt} = -\vec{C}_t \cdot \nabla \Psi, \quad (22)$$

with the Forward Euler method, where \vec{C}_t is given by Eq. (9). This evolution guarantees that the contour of interest moves according to the evolution given by Eq. (9). In fact, for computational efficiency, the level set function, Ψ , is only evolved in a thin band near the contour. After each explicit level set function evolution step, the level set function is reinitialized to preserve distance. This reinitialization maintains a numerically well-behaved level set function while maintaining the position of the contour or surface.

Curvature of the contour, κ , and mean curvature of the surface, H , are computed by non-linear functions of derivatives of Ψ ,

$$\kappa = \frac{-\Psi_x^2 \Psi_{yy} + 2\Psi_x \Psi_y \Psi_{xy} - \Psi_y^2 \Psi_{xx}}{(\Psi_x^2 + \Psi_y^2)^{3/2}} \quad (23)$$

$$H = \frac{\begin{pmatrix} \Psi_{xx}(\Psi_y^2 + \Psi_z^2) + \Psi_{yy}(\Psi_x^2 + \Psi_z^2) + \Psi_{zz}(\Psi_x^2 + \Psi_y^2) \\ -2\Psi_{xy}\Psi_x\Psi_y - 2\Psi_{yz}\Psi_y\Psi_z - 2\Psi_{xz}\Psi_x\Psi_z \end{pmatrix}}{2(\Psi_x^2 + \Psi_y^2 + \Psi_z^2)^{3/2}}. \quad (24)$$

Convergence of this algorithm is guaranteed since the evolution itself is derived from the first variation of a non-negative Lagrangian originating from the cost functional. Thus, the evolution of Eq. (9) with small enough time step ensures that each iteration decreases the cost functional.

The evolution in Eq. (9) contains three terms. The terms containing α and β are essentially a transport equation while the term containing γ is essentially a diffusion equation. Numerical level set implementations for equations such as this is well understood and are used in various works [16, 20]. This evolution is guaranteed to be stable as long the transport terms are stable and the diffusion term is stable. In this implementation, the time step is chosen so that the transport term obeys the Courant-Friedrichs-Lewy (CFL) condition

and so that the diffusion term is also stable. A sufficient time step for these two conditions is,

$$\Delta t < \frac{1}{|\Gamma| + 2(D - 1)}, \quad (25)$$

where D is the dimensionality of the image density function, and,

$$\begin{aligned} \Gamma = & -\alpha \left(\hat{f}_{R_1}(\vec{C}(\ell)) - \hat{f}_{R_2}(\vec{C}(\ell)) \right) \sum_{\theta \in \Theta} (\hat{p} - p) (P_{\theta}(\vec{C}(\ell)), \theta) \\ & - \beta \left(\|\nabla \hat{f}_{R_1}(\vec{C}(\ell))\|^2 - \|\nabla \hat{f}_{R_2}(\vec{C}(\ell))\|^2 \right). \end{aligned} \quad (26)$$

2.3.3 Algorithmic Implementation

Note that the evolutions, $\hat{f}_{R_1,t}^n$ and $\hat{f}_{R_2,t}^n$, depend on the reconstruction estimates, \hat{p} , and thus is it required to update the projections each time these updates are performed. In the present implementation, repetitive recomputation of the estimated projections, \hat{p} , is avoided by updating a copy of the \hat{p} each time a pixel in the estimated density function, \hat{f} , is changed. The estimated projections are completely recomputed from the estimated density function after a fixed number of contour iterations to avoid numerical error. In the present implementation, this number of iterations is 50 since it was found to produce a reasonable tradeoff between speed and numerical accuracy, but the specific choice of when to recompute the projections is not important.

Figure 5 offers a block diagram depiction of the algorithm for implementing these evolution equations. First, the contour and image functions are initialized. We will discuss this initialization more in Chapter 4. Next, the function evolution is shown in more detail on the right. Note that the evolution of the function is computed completely before the image pixels are updated. Next, the projections are updated in the same loop as the pixels themselves, thereby avoiding the use of multiple loops and thus multiple lookups of the same memory addresses. As mentioned above, every certain number of iterations of the contour, the projections are recomputed.

2.3.4 Weighting Parameters

It should be noted that the weighting parameters α , β , and γ control the tradeoff between data fidelity, function smoothness, and surface area penalty. Most of the simulation results

in thesis have been obtained by only scaling these parameters to account for differences in image size. Thus, the proposed method does not require significant fine tuning to achieve reasonable results since the piecewise smooth nature of most natural images is similar. In the validation section it is specifically noted that computing a segmentation with one set of weights is often beneficial to obtain an accurate segmentation, and once the accurate segmentation is obtained, then the weight of the function smoothness is lowered to produce a more accurate reconstruction. This is denoted in Chapter 5 as the proposed modified method. To obtain accurate reconstructions of data whose nature is very much less or more piecewise smooth than most medical images, fine tuning of these parameters may be necessary.

2.4 Summary

In this chapter, a technique has been presented for reconstruction of tomographic images from Radon transform projections that explicitly deals with the under-constrained nature of the limited-angle tomography problem by enforcing priors on the reconstructed image. The technique, in addition to accurately reconstructing images, also separates the image into regions, thereby segmenting the image. Segmenting the reconstructions of the methods that have been compared against would produce inferior results to the segmentations obtained directly from this model. In limited-angle tomography filtered back projection produces reconstructions that are streaky and hence difficult to segment. Piecewise constant reconstruction produces segmentations but they are inaccurate when the image is not piecewise constant.

The computational requirements of this technique are much higher than that of filtered back projection and significantly higher than that of piecewise constant reconstructions due to the evolution of the image intensity functions. Chapter 3 specifically addresses this issue by introducing a multiscale technique for computation of the estimated density functions. In addition, little discussion has been made about the type of initial contour. This will be discussed in Chapter 4.

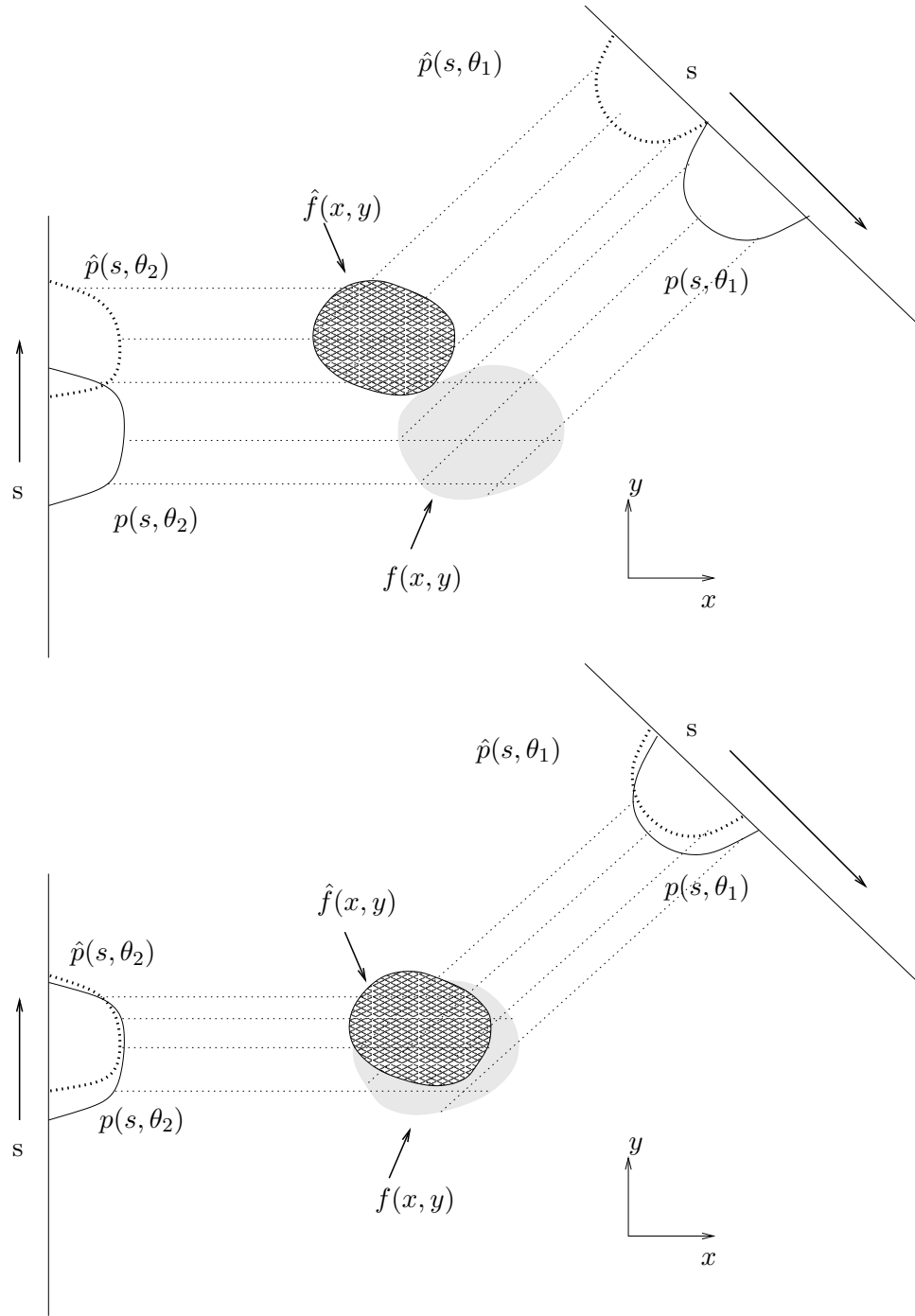


Figure 4: Illustration of two cases to show effect of projection matching term, J_1 , to the cost functional. Top: higher value of J_1 . Bottom: lower value of J_1 .

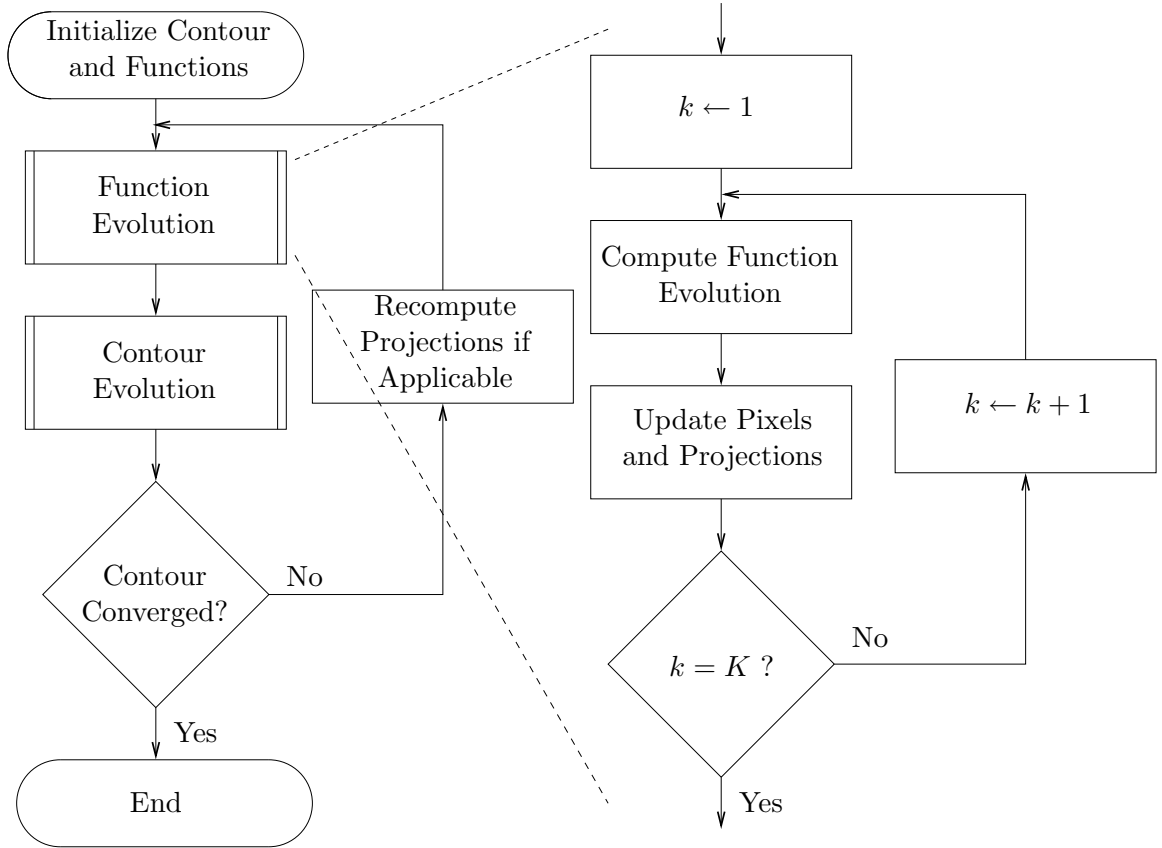


Figure 5: Block diagram depicting implementation of tomographic reconstruction evolution equations.

CHAPTER III

MULTISCALE TOMOGRAPHY COMPUTATION

3.1 Introduction

In Chapter 2, a model was developed for simultaneous segmentation and reconstruction of piecewise smooth tomographic images. The reconstructed image was modeled as a disjoint union of two regions that each have separate smooth density functions and whose boundary is deformable. A cost was imposed on the deviation of the reconstructed image from piecewise smoothness. Piecewise smoothness implies that large gradients in image intensity within regions should be penalized. In addition, total contour length should also be penalized, for otherwise, the contour could converge to an irregular shape to match image noise.

Although the method is powerful and has the ability to perform excellent reconstructions on tomography images with a limited number of projections, the tremendous power of deformable surface models does not come without a cost. Typically, the computational requirements of such algorithms are high and specific techniques must be developed to make the algorithms computationally feasible. The elliptic partial differential equation (PDE) that appears due to the smoothness constraint in our model lends itself well to using multiscale techniques similar to multigrid relaxation techniques [11]. Multigrid relaxation methods yield a significant speed improvement for the solution of such PDE's. The authors refer the reader to [12] for an introduction to multigrid relaxation methods. Multigrid relaxation methods or related methods have had much success in addressing computational problems in computer vision including anisotropic diffusion [1], optical flow [8, 22], and image processing and analysis [27, 68].

In this chapter, the previously developed model is reviewed for the purpose of describing a multiscale method for the solution of the PDE's that define the image density functions. The multiscale technique has the advantage of having lower overall computational cost than

gradient based methods, and also avoids suboptimal local minima that occur in gradient based methods where only a partial solution is attained. Finally, the computational speed improvement is shown and the ability of the proposed multiscale technique to avoid local minima with ideal and noisy simulated projection data is illustrated.

3.2 Evolution Equations

Recall from Chapter 2 that the two-dimensional evolution equations that will minimize the proposed cost functional are,

$$\begin{aligned}\vec{C}_t(\ell) &= -\alpha \left(\hat{f}_{R_1}(\vec{C}(\ell)) - \hat{f}_{R_2}(\vec{C}(\ell)) \right) \sum_{\theta \in \Theta} (\hat{p} - p) (P_\theta(\vec{C}(\ell)), \theta) \vec{\nu}_{R_1}(\ell) \\ &\quad - \beta \left(\|\nabla \hat{f}_{R_1}(\vec{C}(\ell))\|^2 - \|\nabla \hat{f}_{R_2}(\vec{C}(\ell))\|^2 \right) \vec{\nu}_{R_1}(\ell) \\ &\quad + \gamma \kappa(\vec{C}(\ell)) \vec{\nu}_{R_1}(\ell)\end{aligned}\tag{27}$$

$$\begin{aligned}\hat{f}_{R_1,t}(x, y) &= -\alpha \sum_{\theta \in \Theta} (\hat{p} - p) (P_\theta(x, y), \theta) \\ &\quad + \beta \Delta \hat{f}_{R_1}(x, y)\end{aligned}\tag{28}$$

$$\begin{aligned}\hat{f}_{R_2,t}(x, y) &= -\alpha \sum_{\theta \in \Theta} (\hat{p} - p) (P_\theta(x, y), \theta) \\ &\quad + \beta \Delta \hat{f}_{R_2}(x, y).\end{aligned}\tag{29}$$

The three-dimensional evolution equations are,

$$\begin{aligned}\vec{S}_t(v) &= -\alpha \left(\hat{f}_{R_1} - \hat{f}_{R_2} \right) (\vec{S}(v)) \\ &\quad \times \sum_{i=1}^P (\hat{p} - p) (P_{\theta_i, \phi_i}(\vec{S}(v)), Q_{\theta_i, \phi_i}(\vec{S}(v)), \theta_i, \phi_i) \vec{\nu}_{R_1}(\vec{S}(v)) \\ &\quad - \beta \left(\|\nabla \hat{f}_{R_1}\|^2 - \|\nabla \hat{f}_{R_2}\|^2 \right) (\vec{S}(v)) \vec{\nu}_{R_1}(\vec{S}(v)) \\ &\quad + \gamma H(\vec{S}(v)) \vec{\nu}_{R_1}(\vec{S}(v))\end{aligned}\tag{30}$$

$$\begin{aligned}\hat{f}_{R_1,t}(x, y, z) &= -\alpha \sum_{i=1}^P (\hat{p} - p) (P_{\theta_i, \phi_i}(x, y, z), Q_{\theta_i, \phi_i}(x, y, z), \theta_i, \phi_i) \\ &\quad + \beta \Delta \hat{f}_{R_1}(x, y, z)\end{aligned}\tag{31}$$

$$\begin{aligned}\hat{f}_{R_2,t}(x, y, z) &= -\alpha \sum_{i=1}^P (\hat{p} - p) (P_{\theta_i, \phi_i}(x, y, z), Q_{\theta_i, \phi_i}(x, y, z), \theta_i, \phi_i) \\ &\quad + \beta \Delta \hat{f}_{R_2}(x, y, z),\end{aligned}\tag{32}$$

3.2.1 Image Intensity Evolution

Equations (28) and (29) in the two-dimensional case, or Eqs. (31) and Eqs. (32) in the three-dimensional case, represent a coupled evolution that will bring the estimated density functions, \hat{f}_{R_1} and \hat{f}_{R_2} , to a state that represents, for a fixed contour, a minimum of the cost functional, J . Implementing this coupled evolution with gradient descent is numerically stable but very computationally expensive, especially in the three-dimensional version of the tomography problem. It is for this reason that the use of a multiscale technique inspired by multigrid methods is proposed. The proposed multiscale method allows for faster convergence of the algorithm and avoids local minima in the surface evolution. The implementation of this coupled evolution with gradient descent is discussed first, followed by the details of the proposed multiscale technique.

3.2.2 Gradient Descent

The evolution in Eqs. (28) and (29) can be implemented by updating image intensity functions, \hat{f}_{R_1} and \hat{f}_{R_2} , by the forward Euler scheme in the direction of the negative gradient, i.e.,

$$\hat{f}_{R_1}^{n+1} = \hat{f}_{R_1}^n + \Delta t \hat{f}_{R_1,t}^n \quad (33)$$

$$\hat{f}_{R_2}^{n+1} = \hat{f}_{R_2}^n + \Delta t \hat{f}_{R_2,t}^n, \quad (34)$$

where the superscript n denotes iteration number and Δt is the iteration time step parameter. This time step parameter, Δt , is chosen to ensure stability of the algorithm. A sufficient condition for numerical stability of the algorithm is when,

$$\Delta t < \frac{2}{\alpha NP + 4\beta D}, \quad (35)$$

where D is the dimensionality of the image density function, N is the length of the longest dimension of the image density function, and P is the number of projections. The proof of this appears in Appendix B.

In this manner, the update equations minimize the cost functional, thereby ensuring both smoothness of the reconstruction and that the reconstruction projections match the measured projections.

However, in the three-dimensional version of the tomography problem, this gradient descent algorithm is slow to converge to the minimum of the cost function, J . Iterating Eqs. (33) and (34) until convergence is very computationally costly. Since a two-step algorithm is used, whereby the surface is held fixed while the estimated density functions are being evolved, accuracy of the surface evolution depends on the accuracy obtained during the density function evolution. To ensure fast convergence of the density functions and thereby fast and accurate convergence of the surface, a multiscale method is proposed that takes advantage of the smoothness property of the estimated density functions.

3.2.3 Multiscale Method

Convergence of Eqs. (33) and (34) necessarily happens when $\hat{f}_{R_1,t} = \hat{f}_{R_2,t} = 0$, which occurs when,

$$-\alpha \sum_{i=1}^P p(P_{\Theta_i}(\mathbf{x}), Q_{\Theta_i}(\mathbf{x}), \Theta_i) = -\alpha \sum_{i=1}^P \hat{p}(P_{\Theta_i}(\mathbf{x}), Q_{\Theta_i}(\mathbf{x}), \Theta_i) + \beta \Delta \hat{f}_{R_i}, \quad (36)$$

for all $\mathbf{x} \in R_i$ for $i = 1, 2$. When discretely implemented, Equation (36) can be seen as a linear equation, $\mathcal{P} = \mathcal{LF}$. The linear operator, \mathcal{L} , acts on a pixel in the estimated image density function, \mathcal{F} , by summing all of the values of the projections at the points on the projections to which that pixel projects. Thus, the projections can be seen as an intermediate calculation step in the process of computing the linear operator. While this process corresponds to a matrix multiplication, all of the elements of the matrix are not explicitly computed since the matrix would require too much memory to store each element¹. Thus \mathcal{LF} is not computed as a direct matrix multiplication. It is for this reason that standard multigrid methods, i.e., operating on a downsampled image by a coarse scale operator, would be difficult to apply.

A simpler way to use the general concepts behind multigrid methods is proposed to create a multiscale technique for solving Eq. (36). The residual is computed in the projection domain instead of in the image domain. This allows downsampling to occur purely in the projection domain. Then the downsampled image that projects to the downsampled residual

¹Even for a modest 2-D image, e.g., 100×100 pixels, and with the values of the matrices stored as 4-byte floating point numbers, storing the matrix would require 373 Megabytes. A 3-D image of size $100 \times 100 \times 100$ pixels would require 3.73 Gigabytes.

is solved for and used as a coarse scale correction to the image density function. Finally, gradient descent relaxation is used on the corrected image on the original grid to achieve fine scale accuracy of the solution.

More formally, let \hat{f}_{fine} be the discrete version of the estimated image density function on the original (fine scale) image grid and p_{fine} be the discrete version of the true measured projections on the original (fine scale) projection grid. Then by computing the projection residual,

$$r_{\text{fine}} \leftarrow p_{\text{fine}} - \text{Pr}_{\text{fine}}\{\hat{f}_{\text{fine}}\}, \quad (37)$$

on the finest grid, where Pr_{fine} is the fine grid projection operator, and downsampling the projection residual by downsampling operator, D , the downsampled residual,

$$r_{\text{coarse}} \leftarrow Dr_{\text{fine}}, \quad (38)$$

is obtained.

We then solve for the coarse scale correction, c_{coarse} by solving,

$$r_{\text{coarse}} = \text{Pr}_{\text{coarse}}\{c_{\text{coarse}}\}, \quad (39)$$

for c_{coarse} with the gradient descent techniques mentioned in Section 3.2.2. The coarse scale computation incurs less computational cost than solving Eqs. (33) and (34) on the fine grid.

The coarse scale correction is then upsampled by $c_{\text{fine}} \leftarrow Uc_{\text{coarse}}$, and is added to the fine scale estimated image density by,

$$\hat{f}_{\text{fine}} \leftarrow \hat{f}_{\text{fine}} + c_{\text{fine}}. \quad (40)$$

We have illustrated this process in Figure 6. The smooth nature of the image density functions, which is due to the function smoothness regularizer, ensures that the update of the image density function by the coarse scale correction will make the residual smaller, thus bringing the image density function closer to convergence.

Note that while the above procedure describes the process for the two scale method, it is possible to coarsify the projection residual multiple times and solve for the coarse scale correction on an image grid corresponding to this coarse grid. That is, once the residual

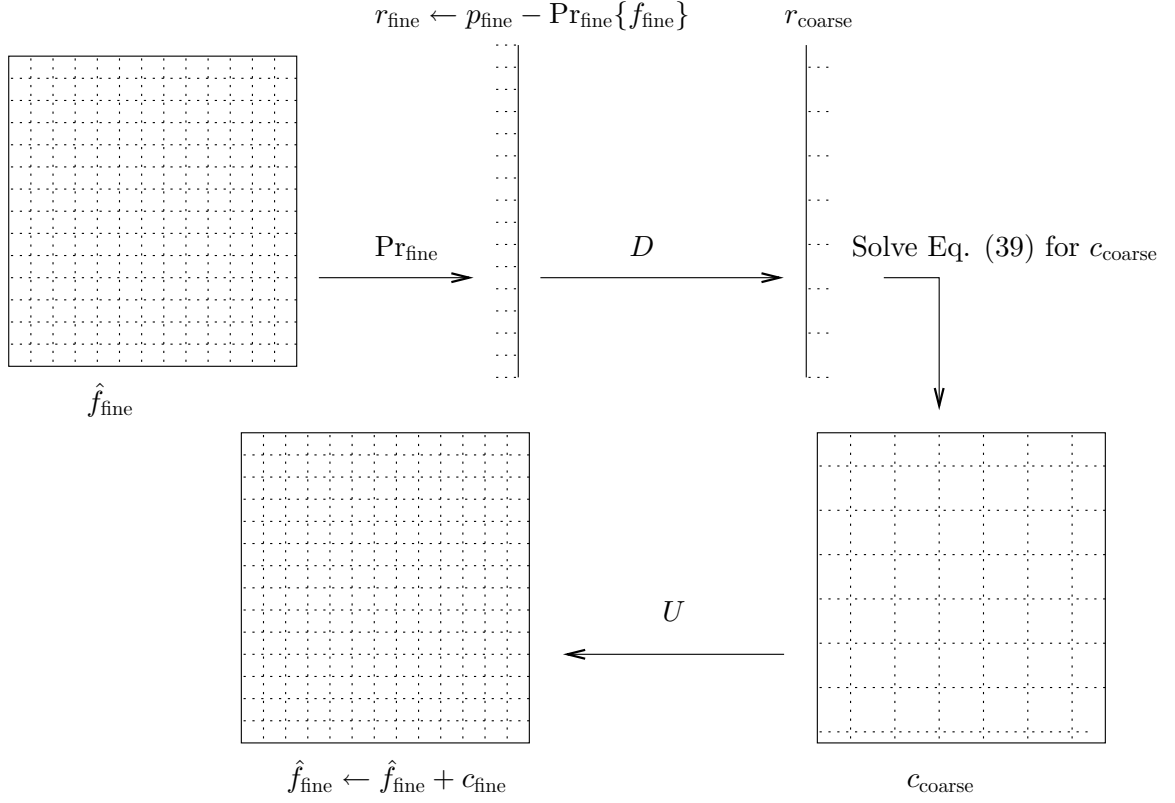


Figure 6: Illustration of two scale method for solving for image density functions.

is calculated on the finest grid, it is downsampled several times until a pre-selected coarse grid size is reached. At this point, the correction is solved for on this coarse grid. Then, it is upsampled in the image domain to the next finest grid level where the correction is obtained more accurately before it is upsampled again. Eventually, it is used to correct the original estimated density function. Figure 7 illustrates this multiscale solution for the coarse grid.

3.3 Computational Cost

In Table 3 we have listed the number of surface evolution iterations that each algorithm requires to converge and the total computation time for various images and algorithms. The number of image iterations on the fine grid is a parameter of each algorithm and is shown in the third column of the table. The multiscale algorithm requires less iterations on the fine grid to solve for the image function accurately. Note that the multiscale algorithm

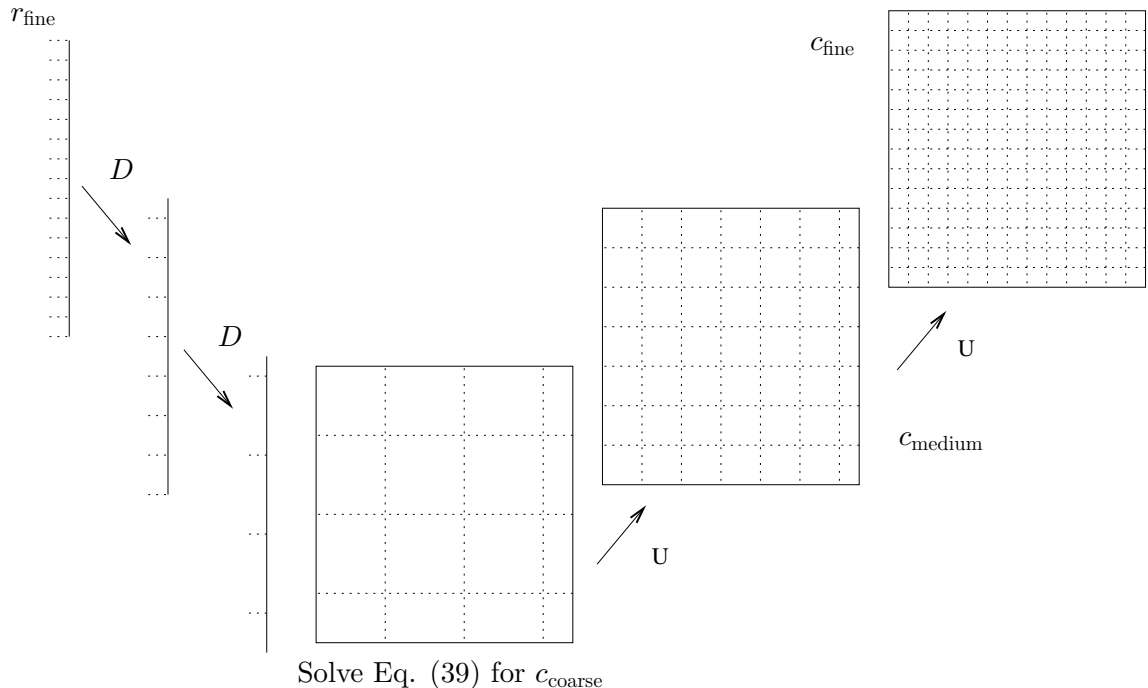


Figure 7: Illustration of multiscale method for solving image density functions.

requires less surface iterations to converge. This is because a more accurate solution of the image function enables faster convergence of the surface. The simulated cube image is $51 \times 51 \times 51$ pixels and uses 15 simulated projections evenly spaced within a half-circle on the $x - y$ plane. The heart phantom image is $64 \times 47 \times 31$ pixels and uses 40 simulated noisy projections evenly spaced within a half-circle on the $x - y$ plane. All computation times reported are processor times on a 1.7 GHz workstation.

Table 1: Computational performance of algorithms for various images.

Image	Technique	Image Iterations on Fine Grid	Surface Iterations until Convergence	Computation Time
Simulated Cube	Grad. Desc.	1	4100	60.2 minutes
Simulated Cube	Grad. Desc.	3	1800	48.8 minutes
Simulated Cube	Multiscale	1	600	21.5 minutes
Heart Phantom	Grad. Desc.	1	3100	87.1 minutes
Heart Phantom	Multiscale	1	500	60.4 minutes

CHAPTER IV

VARIATIONAL REGION INITIALIZATION

4.1 *Introduction*

The effect of allowing piecewise smooth images is a greatly increased number of free parameters in the image model. As the number of free parameters increase, it becomes more likely to reach a local minimum that is not the global energy minimum, and thus the dependence on algorithm initialization becomes more important. This dependence on initialization is particularly prevalent in surface evolution methods due to their local nature. However, in the above mentioned surface evolution papers, little attention is given to the surface initialization. Bruandet et. al. use thresholded filtered back projection (FBP) [14]. Whitaker and Elangovan briefly compare thresholded minimum norm reconstruction (MNR) with thresholded FBP [72]. Yu and Fessler do not explicitly mention the technique they used for surface initialization.

Thresholded FBP is a reasonable surface initialization since FBP is used in many commercial applications and since properties of filtered back projection are well-studied. However, FBP does not work well when the number of projections is low. A detailed description of filtered back projection can be found in [31]. Thresholded MNR seems natural mathematically, but minimizing the norm of the image pixel intensities has no justification for natural images. Another disadvantage that both thresholded FBP and MNR share is the choice of an arbitrary threshold.

Dividing the image domain into several small regions that span the entire image works well in theory. However, this is an extremely inefficient initialization, since computation time for level set implementations scales with total surface area.

In this paper we present a novel method for surface initialization of tomographic images. The proposed method does not depend on the choice of an arbitrary threshold or the geometry of the imaged object. For proof of concept, we use surfaces obtained with this method

as initializations for a piecewise smooth tomographic reconstruction model. We compare the segmentations obtained from the proposed method with other used segmentations from thresholded MNR.

4.2 Tomography Problem Statement

A D -dimensional tomographic image is a function, $f : \Omega \rightarrow \mathbb{R}$, where Ω , the image domain, is a compactly supported subset of \mathbb{R}^D . This function is often called the image density. The projections or sinograms of this image are N , $(D - 1)$ -dimensional functions given by,

$$p(\theta_i, \mathbf{s}) = \int_{\Omega} f(\mathbf{x}) \delta(R_{\theta_i}(\mathbf{x}) - \mathbf{s}) d\mathbf{x}, \quad (41)$$

for $i = 1, \dots, N$, where θ represents the angles at which the projection is taken, and R_{θ} is a function that maps a point in the image domain to its corresponding point, \mathbf{s} , on the projection at angle θ .

The tomographic reconstruction problem is that of determining an estimate, \hat{f} , of the unknown image function, f , from the N projections. Given a sufficient amount of projections, this problem is invertible. In practice however, obtaining this much data is not practical. Therefore, authors have attempted to put constraints on the class of admissible image functions to reduce degrees of freedom apparent in the under constrained problem.

The method proposed by Bruandet et. al. assumes the image function is binary and piecewise constant, one image region having density 0 and the other region having an *a priori* known density, μ . This restrictive image constraint is reasonable when imaging a homogeneous object. The image regions are separated by a surface, Γ , that is a level set of a function Φ . The authors derive a partial differential equation (PDE) describing the evolution of Φ . The evolution of Φ is chosen to minimize the back projection of the difference between the projections of the image and the measured projections.

The technique proposed by Whitaker and Elangovan also assumes piecewise constant image regions separated by a deformable surface. Whitaker does not assume prior knowledge of the image densities; The image densities, β_0 and β_1 , are updated iteratively so that the measured projections match the projections of the current image estimate. Whitaker's formulation is a generalized version of Bruandet's.

Yu and Fessler presented a discrete cost functional to model the reconstruction by an image domain partitioned into two functions under a local image smoothness constraint. In their method, the two functions are separated by a deformable discrete boundary analogous to the surfaces presented by Bruandet and Whitaker.

These authors have all added constraints to the under determined tomographic reconstruction problem. All three models have a common free parameter, i.e., the surface that borders image regions, but each model parameterizes the image function with an increasing number of free parameters. For a discrete image, Bruandet, Whitaker, and Yu respectively allow zero, two, and p parameters to model the image function's intensity, where p is the number of pixels in the image. Each of these authors have formulated a surface evolution to move the boundary of the image regions. The local nature of surface evolution algorithms can introduce local minima in the cost functionals. This is especially true with an increasing number of free image function parameters.

4.3 *Proposed Initial Segmentation*

4.3.1 Image Model and Functional

To obtain an initial segmentation, and a corresponding surface separating the foreground and background, we propose the following model. We model the reconstructed image as a function,

$$\hat{f}(\mathbf{x}) = (1 - u(\mathbf{x}))\alpha_0 + u(\mathbf{x})\alpha_1, \quad (42)$$

where $u(\mathbf{x}) : \Omega \rightarrow \mathfrak{R}$ represents whether a given point is in the foreground or the background, and α_0 and α_1 are two scalar constants. It is desired to obtain a segmentation of the tomographic image, f , with only knowledge of its projections, $p(\theta, s)$, as given by Eq. (41). The cost functional,

$$F(u, \alpha_0, \alpha_1) = \sum_{\theta \in \Theta} \int_{P(\theta)} (p(\theta, \mathbf{s}) - \hat{p}(\theta, \mathbf{s}))^2 d\mathbf{s} + \lambda \int_{\Omega} (u(\mathbf{x}) - 1)^2 u^2(\mathbf{x}) d\mathbf{x}, \quad (43)$$

where \hat{p} are the projections of the estimated image \hat{f} , and Θ is the set of angles at which projections were measured. Note also that $P(\theta)$ is the domain of the projection at angle θ . We can adjust the tradeoff between the two terms with the scalar constant λ . High

values of λ more strongly enforces the bimodality of the image. The first term ensures that the projections of the model image match the measured projections and the second term penalizes pixels for deviating from 0 or 1, i.e., background or foreground respectively. The integrand of the second term is shown as a function of $u(\mathbf{x})$ in Figure 8.

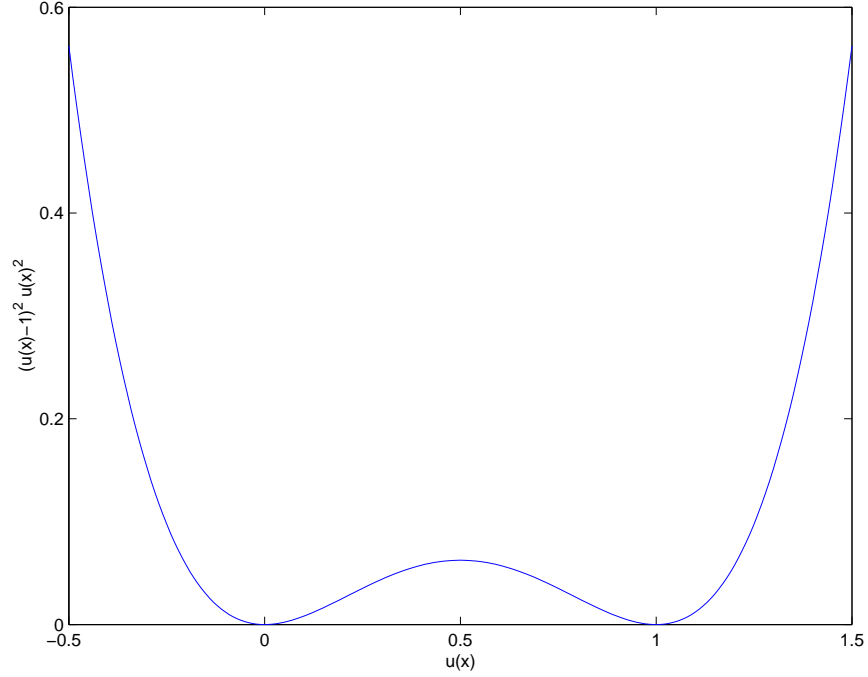


Figure 8: Integrand of the second term in Eq. (43) as a function of $u(\mathbf{x})$. This term penalizes $u(\mathbf{x})$ for deviating from 0 or 1.

4.3.2 Evolution of Reconstruction Parameters

The gradient of the cost functional with respect to the function $u(\mathbf{x})$ is obtained from the from its first variation. By allowing the function $u(\mathbf{x})$ to change with time, the cost functional also changes with time. Differentiating the cost functional with respect to time yields,

$$F' = \frac{d}{dt} \sum_{\theta \in \Theta} \int_{P(\theta)} ((p - \hat{p})(\theta, \mathbf{s}, t))^2 d\mathbf{s} + \lambda \frac{d}{dt} \int_{\Omega} (u(\mathbf{x}, t) - 1)^2 u^2(\mathbf{x}, t) d\mathbf{x} \quad (44)$$

$$= -2 \sum_{\theta \in \Theta} \int_{P(\theta)} (p - \hat{p})(\theta, \mathbf{s}, t) \frac{d\hat{p}}{dt} d\mathbf{s} + 2\lambda \int_{\Omega} ((u - 1)u^2 + (u - 1)^2 u) u_t d\mathbf{x}, \quad (45)$$

where u_t is the partial derivative of u with respect to t . The term $\frac{d\hat{p}}{dt}$ is,

$$\frac{d\hat{p}}{dt}(\theta, s, t) = \frac{d}{dt} \int_{\Omega} ((1 - u(\mathbf{x}, t)) \alpha_0 + u(\mathbf{x}, t) \alpha_1) \delta(R_{\theta_i}(\mathbf{x}) - \mathbf{s}) d\mathbf{x} \quad (46)$$

$$= (\alpha_1 - \alpha_0) \int_{\Omega} u_t(\mathbf{x}, t) \delta(R_{\theta_i}(\mathbf{x}) - \mathbf{s}) d\mathbf{x}. \quad (47)$$

Substituting $\frac{d\hat{p}}{dt}$ into Eq. (45), and using the sifting property of the δ function in the variable s , we obtain,

$$F' = 2 \int_{\Omega} u_t \left[(\alpha_0 - \alpha_1) \sum_{\theta \in \Theta} (p - \hat{p})(\theta, R_{\theta}(\mathbf{x})) + \lambda (2u^3 - 3u^2 + u) \right] d\mathbf{x}. \quad (48)$$

It follows that the direction of evolution of u that will most quickly minimize the cost functional is by setting,

$$u_t(x, t) = (\alpha_1 - \alpha_0) \sum_{\theta \in \Theta} (p - \hat{p})(\theta, R_{\theta}(\mathbf{x})) - \lambda (2u^3 - 3u^2 + u), \quad (49)$$

that is, we have obtained an evolution of $u(x, t)$ that will minimize our cost functional.

To solve for the optimal α_0 and α_1 at each evolution step, we minimize the cost functional with respect to these constants. That is by setting,

$$\frac{\partial F}{\partial \alpha_0} = \frac{\partial F}{\partial \alpha_1} = 0, \quad (50)$$

we obtain a system of equations given by,

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad (51)$$

where,

$$a_{11} = \sum_{\theta} \int_{P(\theta)} \left(\int_{\Omega} (1 - u) \delta d\mathbf{x} \right)^2 d\mathbf{s} \quad (52)$$

$$a_{12} = a_{21} = \sum_{\theta} \int_{P(\theta)} \left(\int_{\Omega} u \delta d\mathbf{x} \right) \left(\int_{\Omega} (1 - u) \delta d\mathbf{x} \right) d\mathbf{s} \quad (53)$$

$$a_{22} = \sum_{\theta} \int_{P(\theta)} \left(\int_{\Omega} u \delta d\mathbf{x} \right)^2 d\mathbf{s}, \quad (54)$$

and

$$b_1 = \sum_{\theta} \int_{P(\theta)} p(\theta, \mathbf{s}) \left(\int_{\Omega} (1 - u) \delta d\mathbf{x} \right) d\mathbf{s} \quad (55)$$

$$b_2 = \sum_{\theta} \int_{P(\theta)} p(\theta, \mathbf{s}) \left(\int_{\Omega} u \delta d\mathbf{x} \right) d\mathbf{s}. \quad (56)$$

Note that δ is shorthand for $\delta(R_\theta(\mathbf{x}) - \mathbf{s})$.

The proposed technique is to evolve the function $u(\mathbf{x}, t)$ according to Eq. (43), while at each evolution step to update α_0 and α_1 . Note that no explicit constraints are made that relate values of $u(\mathbf{x}, t)$ that are spatially close. Any such constraints would create a segmentation that would not be suitable for the initialization of a surface evolution technique.

4.3.3 Thresholding

An advantage of this technique is that we have a natural image threshold and therefore a natural way to determine the region boundary. As λ gets larger, the image points are closer to 0 or 1. Therefore, we propose to classify image points into two regions, R_1 and R_2 , based on the criterion: $\mathbf{x} \in R_1$ if $u(\mathbf{x}) \geq 0.5$ and $\mathbf{x} \in R_2$ if $u(\mathbf{x}) < 0.5$. We have thus specified a segmentation of foreground and background. The boundary of the two image regions, R_1 and R_2 , is a surface. Thus, the segmentation and surface are obtained by thresholding our proposed reconstruction.

The surface separating the two image regions is suitable as an initial surface for existing variational tomographic reconstruction methods [14, 72]. An accurate initial surface is also necessary for the generalization of Whitaker's work to piecewise smooth images. In the next section, we will show the performance of the proposed initial surface when used as an initialization for the piecewise smooth generalization of Whitaker's work.

As a remark, we will briefly compare the proposed technique with minimum norm reconstruction (MNR), in which the solution is the image that satisfies the projection data constraints and has the minimum norm. That is, the minimum norm reconstruction would be the \hat{f} that minimizes,

$$F(\hat{f}) = \sum_{\theta \in \Theta} \int_{P(\theta)} (p(\theta, \mathbf{s}) - \hat{p}(\theta, \mathbf{s}))^2 d\mathbf{s} + \lambda \int_{\Omega} |\hat{f}(\mathbf{x})|^2 d\mathbf{x}, \quad (57)$$

MNR will produce reconstructed images whose grayscale point tend to be close to each other in intensity. That is, the formulation in Eq. (57) will not enforce a reconstructed image where pixel values tend to be close to one constant or another. In addition, the chosen threshold in MNR is arbitrary since the image is not inherently bimodal. It is for

this reason that the cost functional we propose is more motivated than minimizing the norm of the reconstructed function.

4.4 *Implementation*

We initialize the constants, $\alpha_0 = 0$ and $\alpha_1 = 2c$, where,

$$c = \frac{\int_{\Omega} f(\mathbf{x}) d\mathbf{x}}{\int_{\Omega} d\mathbf{x}} = \frac{\int_{P(\theta)} p(\theta, \mathbf{s}) d\mathbf{s}}{\int_{\Omega} d\mathbf{x}}. \quad (58)$$

is the average pixel intensity in the image. Although we do not know $f(\mathbf{x})$ directly, it is known that $\int_{\Omega} f(\mathbf{x}) d\mathbf{x} = \int_{P(\theta)} p(\theta, \mathbf{s}) d\mathbf{s}$ for any of the projection angles, θ , since the projections conserve the total image intensity. We initialize the function $u(\mathbf{x}) = 0.5$. Therefore, the initial image function is $\hat{f}(\mathbf{x}) = 0.5\alpha_1 = c$, i.e., all image values are initially the same.

Then we update the current value of $u(\mathbf{x})$ using an forward Euler scheme with the evolution u_t given by Eq. (49). The update equation is of the form,

$$u^{n+1}(\mathbf{x}) = u^n(\mathbf{x}) + \gamma u_t^n(\mathbf{x}, t), \quad (59)$$

where the superscript denotes the iteration number and γ is the algorithm time step which must be kept sufficiently small to ensure stability. After each update of $u(\mathbf{x})$, the optimal values of α_0 and α_1 are solved by the system of equations in Eq. (51).

The algorithm is considered converged when the norm of $u_t^n(\mathbf{x}, t)$ decreases below a certain, predetermined threshold. Upon convergence of the algorithm, we threshold the function $u(\mathbf{x})$ to obtain the segmentations.

For ease of display, the results of the proposed method will be shown in two dimensions, i.e., with a two dimensional image function and with one dimensional projections. We computed on a discrete grid of pixels which represents the image function. The projections are computed using linear interpolation. The integrals in Eq. (49) and Eqs. (52) through (56) are sums in the discrete implementation.

4.5 *Simulations*

4.5.1 *Example Segmentation*

In Fig. 9 we show a synthetic piecewise smooth image that is typical of the types of images we wish to reconstruct. The figure also shows the vertical and horizontal projections of this

image. The image is 200 by 200 pixel and contains three ellipses, one is homogeneous and of high intensity and two smoothly blend into the black background. In Fig. 10 we show four instances of the proposed segmentation algorithm. Each column shows the outcome of the algorithm for a different number of projections, N . The values of N are, from left to right, 2, 5, 9, and 15. The top row contains the final values of $u(\mathbf{x})$. The bottom row contains the thresholded values of $u(\mathbf{x})$; These are the proposed segmentations of the image functions.

Fig. 11 shows the reconstruction of the tomographic image using the initial segmentation proposed in this paper assuming a piecewise smooth image model that is a generalization of the method proposed by Whitaker [72]. From left to right, the figure shows different stages of the piecewise smooth algorithm, i.e., at 200, 500, 700, and 1000 iterations respectively. This reconstruction uses only 9 projections and obtains a nearly perfect reconstruction. The only slight inaccuracy in the reconstructed image is the rounding off of the two upper ellipses. This rounding off occurs due to the unclear boundary of where the ellipses end and the black background begins.

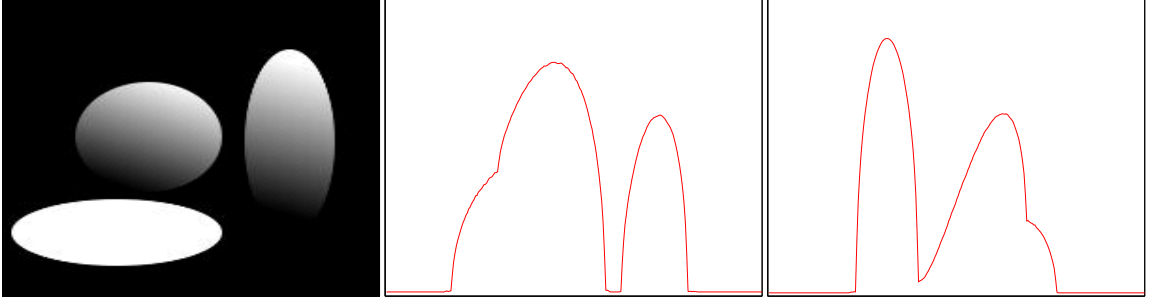


Figure 9: True synthetic piecewise smooth image of three ellipses (left). Vertical and horizontal projections of image (center and right, respectively).

4.5.2 Comparison with Minimum Norm

We will present a comparison with thresholded minimum norm reconstruction (MNR). Fig. 12 shows another piecewise smooth, 100 by 100 pixel, lung CT image and its two projections. Fig. 13 shows two rows of initial segmentations. The top row contains segmentations

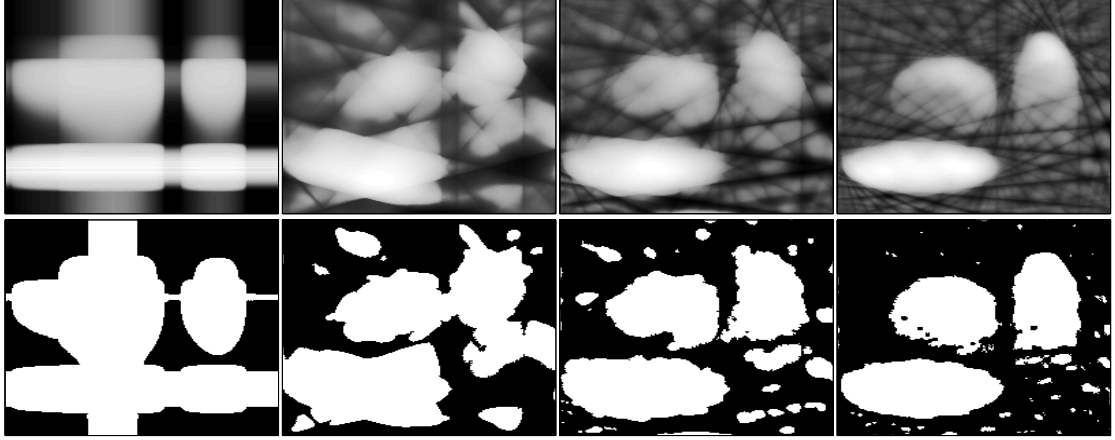


Figure 10: Proposed $u(\mathbf{x})$ function (top row) and segmentations (bottom row) of image shown in Fig. 9. From left to right, the columns represent 2, 5, 9 and 15 equally spaced projections respectively.

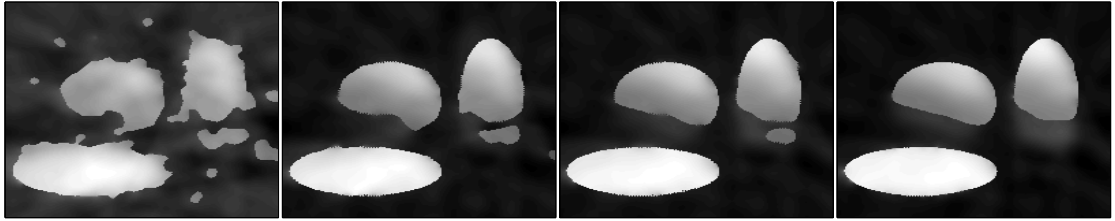


Figure 11: Reconstruction of image in Fig. 9 using piecewise smooth image model when initialized with proposed initial segmentation. From left to right is shown the piecewise smooth algorithm at 200, 500, 700, and 1000 iterations respectively. This reconstruction uses 9 equally spaced projections.

using the proposed method. The bottom row contains segmentations using thresholded MNR. From left to right, the columns in the top row represent segmentations for different thresholds using the proposed method with 15 equally spaced projections. From left to right, the columns in the bottom row represent segmentations for different thresholds with 15 equally spaced projections. We note how the proposed initial segmentation is more robust to choice of threshold.

It is important to notice that the proposed method (top row) more consistently finds the delineations between shapes whereas the thresholded MNR method (bottom row) tends to blend shapes with each other and with the sides of the image. This blending can cause local minima when the segmentation is used as an initialization for deformable surface

models for tomographic reconstruction. The blending is more likely to occur in thresholded MNR since minimizing the norm of the image function does not penalize the image from smoothly changing from foreground to background. The proposed functional penalizes image grayscale values for being between foreground and background. In this way it creates clearer, more accurate boundaries for binary or almost binary images. These accurate boundaries are desired in the segmentation problem and are a useful initialization for variational tomographic reconstruction methods.

In Fig. 14 we see the evolution of the tomographic reconstruction algorithm using 15 projections on the lung CT image. Notice how the algorithm captures much of the fine scale structure. This is made possible due to the accurate initial segmentation proposed in this paper.

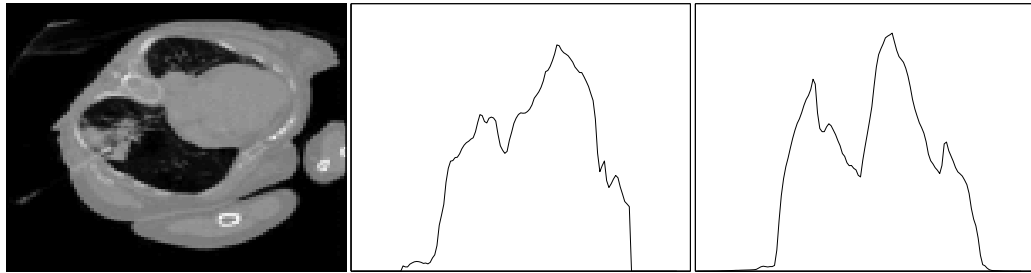


Figure 12: True lung CT image (left). Vertical and horizontal projections of image (center and right, respectively). Image size is 100 by 100 pixels.



Figure 13: Comparison of segmentations of the image in Fig. 12. Comparison is between proposed method (top row) with segmentations from thresholded minimum norm reconstruction (bottom row). From left to right, the columns represent different segmentation thresholds.

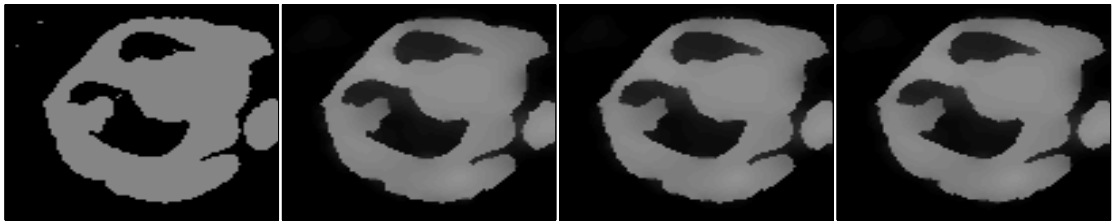


Figure 14: Evolution of tomographic reconstruction algorithm when initialized with the proposed method. From left to right: initial image, image after 50 reconstruction iterations, after 100 iterations, and after 200 iterations.

CHAPTER V

EXPERIMENTAL VALIDATION

5.1 Two-Dimensional Reconstructions

5.1.1 Bone Reconstruction

In Fig. 15 we show a 71×71 -pixel grayscale image that is a cross-section of a piece of bone. This image is nearly piecewise constant. However, unlike the technique presented by Whitaker that models only two image regions, this bone image has three distinct intensity regions. For this reason, it is desirable to consider the three regions separately.

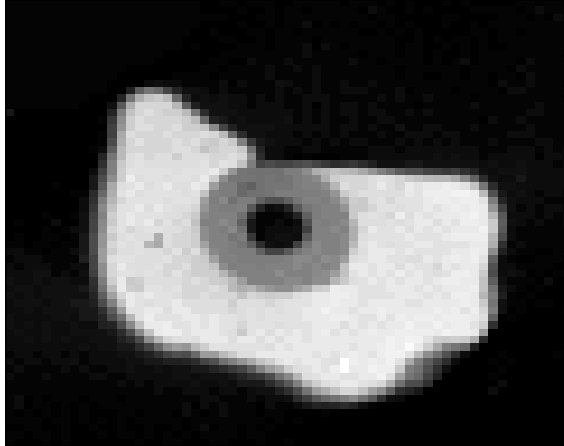


Figure 15: True 71×71 -pixel bone image.

In Fig. 16 we show several stages of evolution of the proposed method. The top left image shows the initialization of the algorithm, where the image domain is subdivided into a large number of densely-spaced separate regions. The evolution of the algorithm proceeds from left to right in the top row and then is continued on the bottom row where it again proceeds left to right. The final reconstruction is show on the bottom right. Note how the three distinct homogeneous regions are accurately captured by the proposed model.

We compare the proposed method with the reconstruction from other techniques in

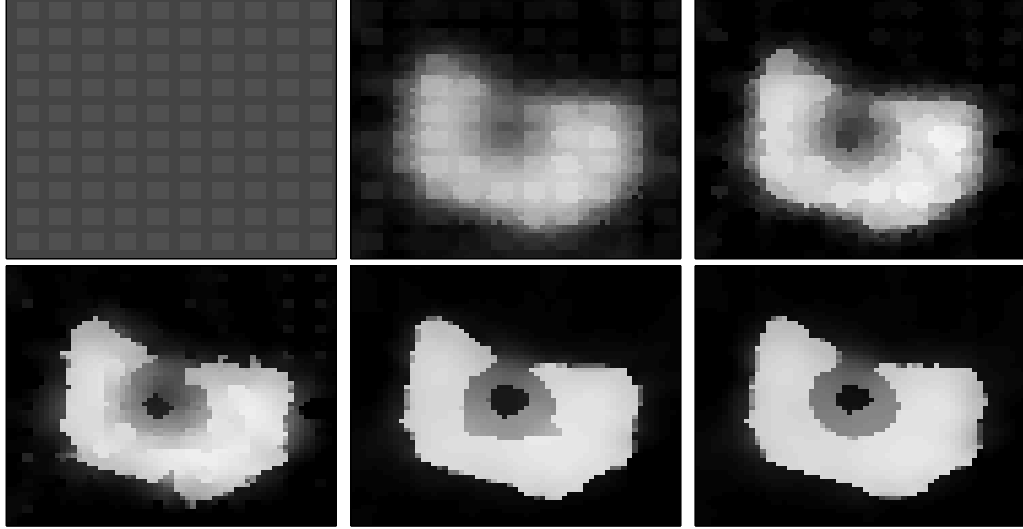


Figure 16: Several stages of bone reconstruction from 9 equally-spaced projections using proposed method. Algorithm initialization in top left image. Evolution proceeds from left to right and is continued in the second row. Final reconstruction on bottom right.

Fig. 17. All reconstructions in this figure have used 9 equally-spaced projections. In the top left, we show the true image for comparison. On the top right, we show the reconstruction using filtered back projection (FBP). On the bottom left, we display the reconstruction using the piecewise constant method proposed by Whitaker. On the bottom right, we show the reconstruction obtained from the proposed method.

Note that the proposed method most accurately represents the delineation between foreground and background. The reconstruction from FBP contains significant streak noise, as typically occurs with under-constrained FBP. The reconstruction from the piecewise constant model does not accurately represent the bone marrow region that lies inside of the bone. This occurs because the piecewise constant model is unable to represent more than two grayscale intensities.

5.1.2 Brain Reconstruction

In Fig. 18 we show a 145×145 -pixel grayscale image that is a cross-section of a human skull and brain. This image has significant piecewise smooth structure, the boundaries between the bone and the gray matter have large intensity variation, but within the bone gray matter regions, the image has significant smoothness.

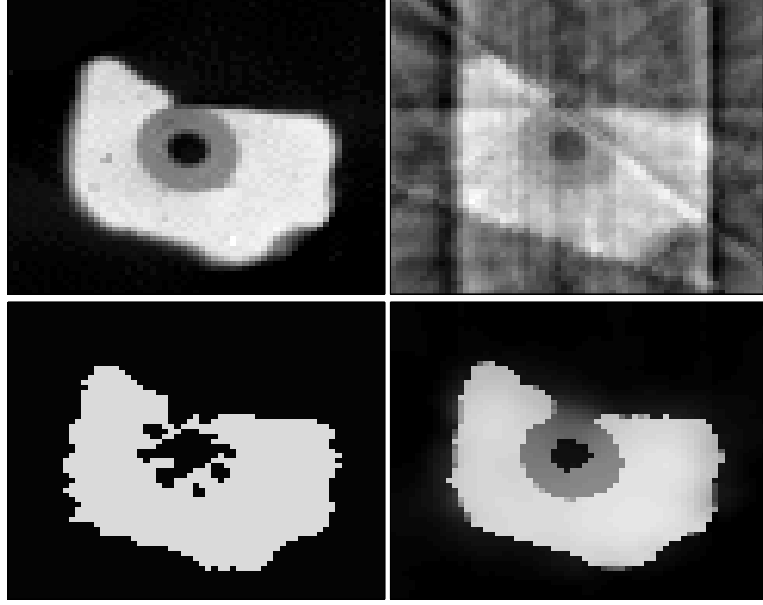


Figure 17: Comparison of bone reconstruction methods from 9 equally-spaced projections. True image (top left), filtered back projection reconstruction (FBP) (top right), piecewise constant reconstruction (bottom left), and proposed piecewise smooth reconstruction (bottom right).

We show, in Fig. 19, several stages of the brain reconstruction using the proposed method. In this reconstruction we used 25 equally-spaced projections. Again, in the top left, we show the initialization of the algorithm with densely-spaced separate square regions. The algorithm proceeds from left to right. Notice how accurately the edges of the skull region are segmented by the contour in the proposed model.

In Fig. 20, we compare the brain reconstruction with the reconstruction from other methods. All methods in this comparison used 25 equally-spaced projections. In the top left we show the true brain CT image. In the top right we show the reconstruction using filtered back projection (FBP). In the bottom left, we show the reconstruction using the piecewise constant model and in the bottom right we show the reconstruction using the proposed piecewise smooth model.

Again note the accuracy with which the proposed model segments some of the small scale features of the skull and brain. The piecewise constant model attempts to represent the medium-intensity gray matter inside the skull region by considering part of the region to be foreground and part of the region to be background. The segmentation produced by

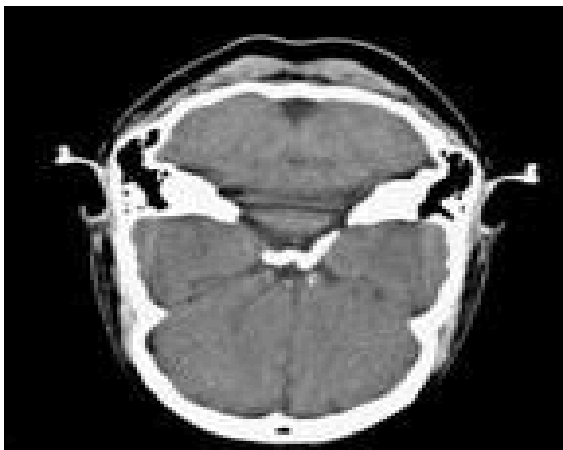


Figure 18: True 145×145 -pixel brain CT image (courtesy of the Department of Medical Imaging and Radiation Sciences at Monash University.)

the proposed model more accurately represents the physical boundaries between anatomical regions.

5.1.3 Method Comparison

We compared numerically and subjectively with two other methods, filtered back projection and the sinogram restoration algorithm proposed by Prince and Willsky [54]. The sinogram restoration algorithm of Prince and Willsky is a multistage algorithm that first normalizes the projections by shifting the center of mass and scaling so that each projection has unit mass. Then, the range of positions in each projection where the object could have projected to are estimated using statistical methods. These are called *support values*. Next, the *finest-grain* sinogram is restored by solving the PDE that minimizes a given cost functional. The cost functional simultaneously penalizes deviation from known projections, rewards smoothness of the reconstruction sinogram, and penalizes energy in the sinogram outside of the support values. In addition, that the mass of each projection is one and that the center of mass is at the image origin is enforced throughout the evolution. Finally, this reconstructed sinogram is shifted back to its original center of mass and its original mass and then it is used to reconstruct the image using filtered back projection.

In our implementation of this method, we used a simple thresholding technique to estimate the range of possible projection energy within the projection domain. However,

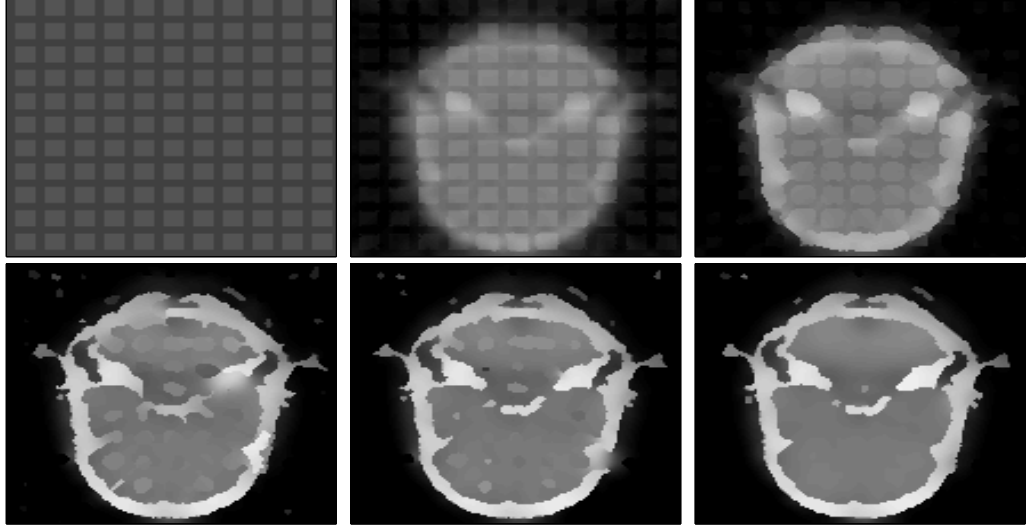


Figure 19: Several stages of brain reconstruction from 25 equally-spaced projections using proposed method. Algorithm initialization in top left image. Evolution proceeds from left to right and is continued in the second row. Final reconstruction on bottom right.

we did this in the case of noiseless projections and obtained very accurate values for the range of possible projection energy. Thus, our implementation represents a best possible case sinogram restoration with this technique.

Table 2 shows the mean squared error between the original image and the reconstructed image for different reconstructions techniques and for varying number of projections. The reconstruction techniques shown are filtered back projection, algebraic reconstruction technique (ART), the sinogram restoration of Prince and Willsky [54] and the reconstruction technique proposed in this thesis, initialized with the densely spaced contours as shown in Fig. 19, and using the multiscale function computation as describe in Chapter 3. The last column shows the result of a modified version of the proposed method that starts out with the segmentation and reconstruction in the fourth row, but where the weight of the function smoothness term in the cost functional is deemphasized. This produces a more accurate image where smoothness isn't enforced too much, but also maintains a very accurate segmentation. The mean squared error reported is given by,

$$\text{MSE} = \frac{\sqrt{\sum_{p \in \Omega} (I(p) - I_r(p))^2}}{|\Omega|}, \quad (60)$$

where Ω is the set of pixels in the image domain, $|\Omega|$ the number of pixels in the image

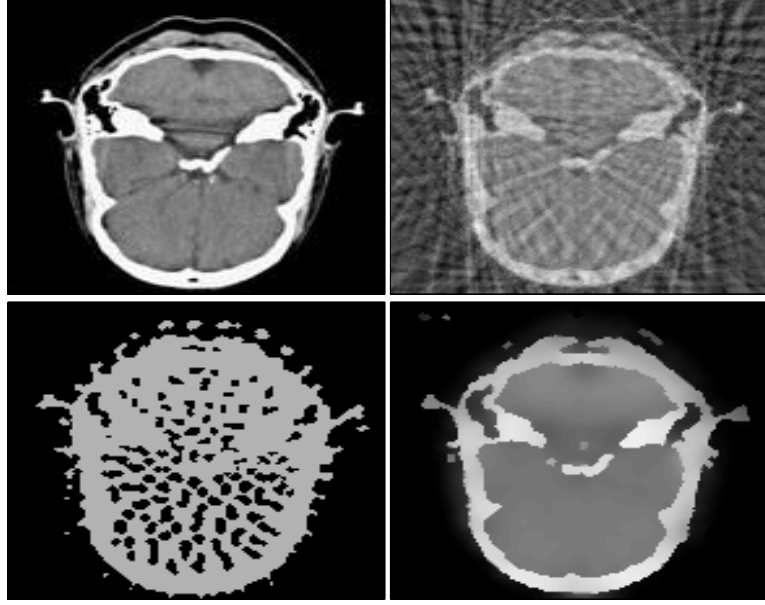


Figure 20: Comparison of brain reconstruction methods from 25 equally-spaced projections. True image (top left), filtered back projection reconstruction (FBP) (top right), piecewise constant reconstruction (bottom left), and proposed piecewise smooth reconstruction (bottom right).

domain, $I(p)$ is the value of the original image at pixel p , and $I_r(p)$ is the value of the reconstructed image at pixel p .

Note how, as the number of projections becomes large, the error of filtered back projection and especially ART becomes lower than the sinogram restoration and the proposed method. The reasons for this are because the projections are noiseless and because the sinogram restoration method and the proposed method assume a strong prior that modifies the image in favor of a more likely image. Filtered back projection does not modify the image or the projections as strongly. However, when the number of projections are low, the sinogram restoration method and the proposed method outperform filtered back projection and ART since both of these methods each successfully incorporate a more valid prior. Unlike the sinogram restoration method of Prince and Willsky, which enforces smoothness of the sinogram, the proposed method incorporate a more realistic prior, i.e., piecewise smoothness within the image domain.

A significant improvement to the proposed method is given by the proposed modified method mentioned above where the weight of the function smoothness term is lowered after

an initial segmentation and reconstruction are obtained. The results of this method also look qualitatively the best as can be seen below.

Table 2: Mean squared error comparison of various 2D reconstruction algorithms.

# Proj.	FBP	ART	Sino. Rest.	Prop. Method	Prop. Method Mod.
10	0.420	0.260	0.244	0.248	0.226
20	0.250	0.190	0.191	0.173	0.171
25	0.209	0.164	0.180	0.167	0.157
30	0.172	0.147	0.171	0.162	0.146
40	0.144	0.115	0.161	0.161	0.133

Figure 21 shows the comparison of the reconstructed brain using three different reconstruction methods. The top row shows filtered back projection reconstruction for 10, 20, 30, and 40 projections starting from the leftmost image and going towards the rightmost image. The second row from the top shows the reconstruction using algebraic reconstruction technique (ART) by producing the image with the minimum norm that matches the given projections. The third row shows the reconstruction results using Prince’s sinogram restoration for the same number of projections. The fourth row shows the results of the reconstruction method proposed in this thesis. Note how the important edges in the image are preserved well when using the proposed method as opposed to where they are blurred in the sinogram restoration method. The fifth and final row shows the proposed method that starts out with the segmentation and reconstruction in the fourth row, but where the weight of the function smoothness term in the cost functional is deemphasized. This produces a more accurate image where smoothness isn’t enforced too much, but also maintains a very accurate segmentation. Note how a significant portion of the fine scale details are incorporated into the reconstruction.

5.2 *Three-Dimensional Reconstructions*

5.2.1 Modified Cardiac Torso Phantom

For validation, we reconstructed three-dimensional images, i.e., single three-dimensional time frames from a sequence of three-dimensional images, that were taken from modifications of the 4-D NURBs-based CARdiac Torso (NCAT) phantom from the University of North Carolina [18, 61, 67]. The NCAT phantom software creates “gold standard” activity and attenuation maps. This NCAT phantom includes a model of the heart with many options for varying the simulation, including addition of cardiac and respiratory motions, inclusion of perfusion abnormalities, variations in radioactivity concentration in non-cardiac structures. The modification of the NCAT phantom was make allowances for the use of left ventricular (LV) boundaries and perfusion defect size and location obtained from actual SPECT perfusion studies [24].

Figure 22 shows four slices of the modified NCAT phantom. Figure 23 shows three view angles of a segmentation of the known phantom image using a binary segmentation method [76]. Note that segmentations such as this are essentially ground truth segmentations and rely on accurate knowledge of the image density function. Segmentations of this high quality are not attainable from limited projection data.

From each of the resulting 72 NCAT-created activity and attenuation maps, was created simulated projections using a projector that modeled all major degradation factors in SPECT. The projector included a low-energy high resolution collimator model that incorporated geometric detector response, simulated photon scatter using the Klein-Nishina formula, and added standard Poisson-distributed noise according to image count levels. More details regarding this projector are discussed in Chen, et al. [18].

5.2.2 Multiscale Reconstructions

Figures 24 and 25 illustrate the ability of the multiscale technique to avoid local minima in the surface evolution. Figure 24 shows the evolution of the surface for a simulated three-dimensional cube with the gradient descent technique where the function evolution is not solved to convergence. The final surface contains an inversion of the surface around the

ideal cube. Since the inversion around the top and sides of the cube would have to pass a state of higher cost in order to invert and model the cube accurately the surface evolution has reached a local minimum. Figure 25 illustrates the evolution of the cube when using the multiscale technique presented in Chapter 3. The density function is solved efficiently and the surface is able to reach the global energy minimum. This is shown by the accurate reconstruction of the cube.

Most of the benefits of the using multiscale tomographic reconstruction however, are in the computational improvement as shown in Chapter 3.

5.2.3 Variational Surface Initialization

The surface initialization method presented in Chapter 4 is an effective way of initializing the surface for the proposed tomographic reconstruction technique. Figure 26 illustrates the segmentation of the three-dimensional cube when initialized with the variational surface technique presented in Chapter 4. On the left of the figure is the initialization produced by this technique. Note that the initialization is extremely close to the cube in shape. The evolution therefore does not require many iterations to reach the final configuration. In this instance only 40 surface evolution steps are required whereas approximately 600 are needed when started with the dense initialization shown in Figs. 24 and 25.

Figure 27 illustrates the reconstruction obtained when using the bimodal surface initialization method presented in Chapter 4. The left column of this figure shows a slice of the density function and of the surface initialization. Notice how the initial surface captures the boundaries of the simulated heart wall very accurately. The figure then shows the evolution of the surface and density function when initializing the surface as shown on the left. The final reconstruction and its corresponding surface are shown on the right.

5.2.4 Heart Perfusion Detection

Figure 28 shows the ability of the proposed reconstruction technique to detect a simulated heart perfusion abnormality. The left image is the ground truth phantom showing the heart perfusion, which is on the bottom of the larger ventricle and is a region of faded image intensity. The right image shows the reconstruction using the proposed reconstruction

method with 40 equally spaced projections. Note how the perfusion abnormality is clearly visible and is able to be detected easily by inspection.

Figure 29 shows the surface resulting from a segmentation of the NCAT phantom when using 64 projections for which noise, attenuation and scatter were added. The top left image is the overhead view of the surface. The top right image is the side view. The bottom left image a back view and the bottom right image is a close up of the heart, showing a hole in the active surface at the location of the heart perfusion. The location of the hole is circled.

Figure 30 shows the ability of the proposed technique to model the edges in an image and to detect the heart perfusion abnormality. In in this example, 64 projection were used where noise, attenuation and scatter were added. The leftmost image is the original phantom. The second image is the filtered back projection reconstruction of the image. The third image is the reconstruction using the proposed method. Notice the clear thinning of the heart wall at the location of the perfusion abnormality. The final image shows a close up of the surface with the perfusion circled.

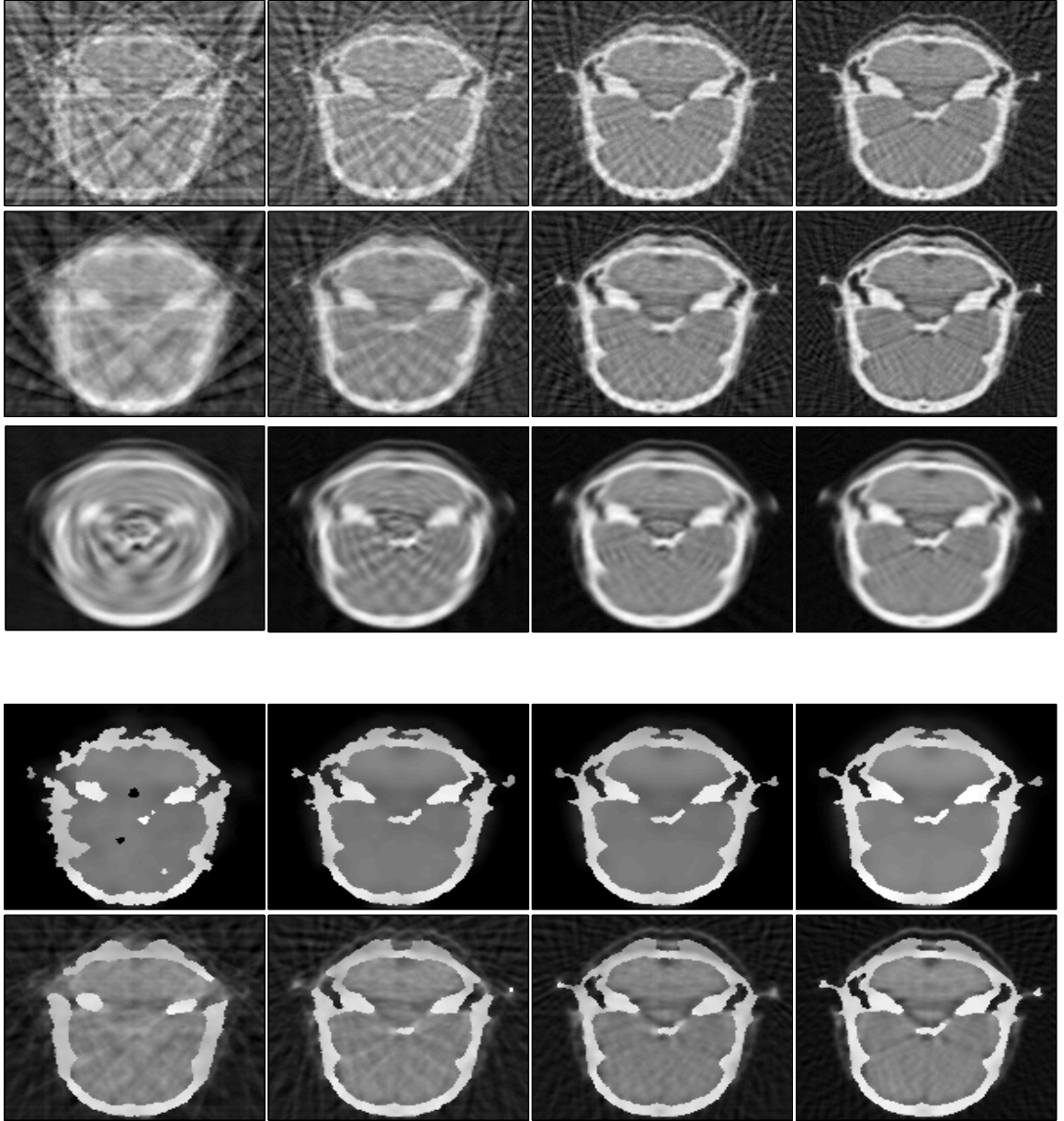


Figure 21: Comparison of reconstructed brain image for different reconstruction methods for varying number of projections. First row: Filtered back projection reconstruction. Second row: Algebraic reconstruction technique. Third row: Sinogram restoration. Fourth row: proposed method with fixed weighting parameters. Fifth row: modified proposed method with function smoothness deemphasized once an accurate segmentation is obtained.

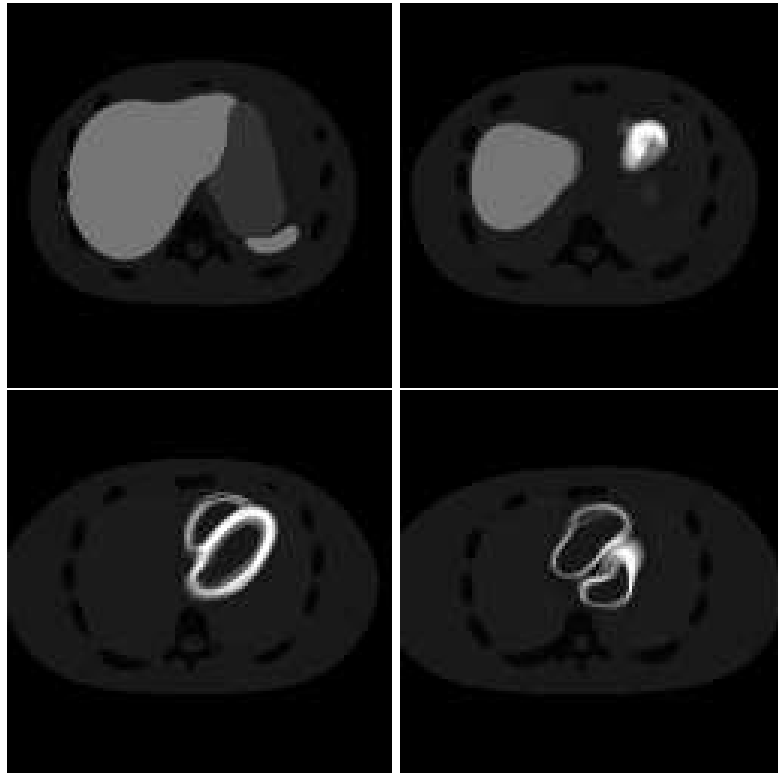


Figure 22: Several slices of modified NCAT phantom showing cross sections of heart and other simulated organs.



Figure 23: Three view angles of a segmentation surface of the known phantom image using a binary segmentation method. Top image: overhead view of segmentation surface. Middle image: side view. Bottom image: underside view.

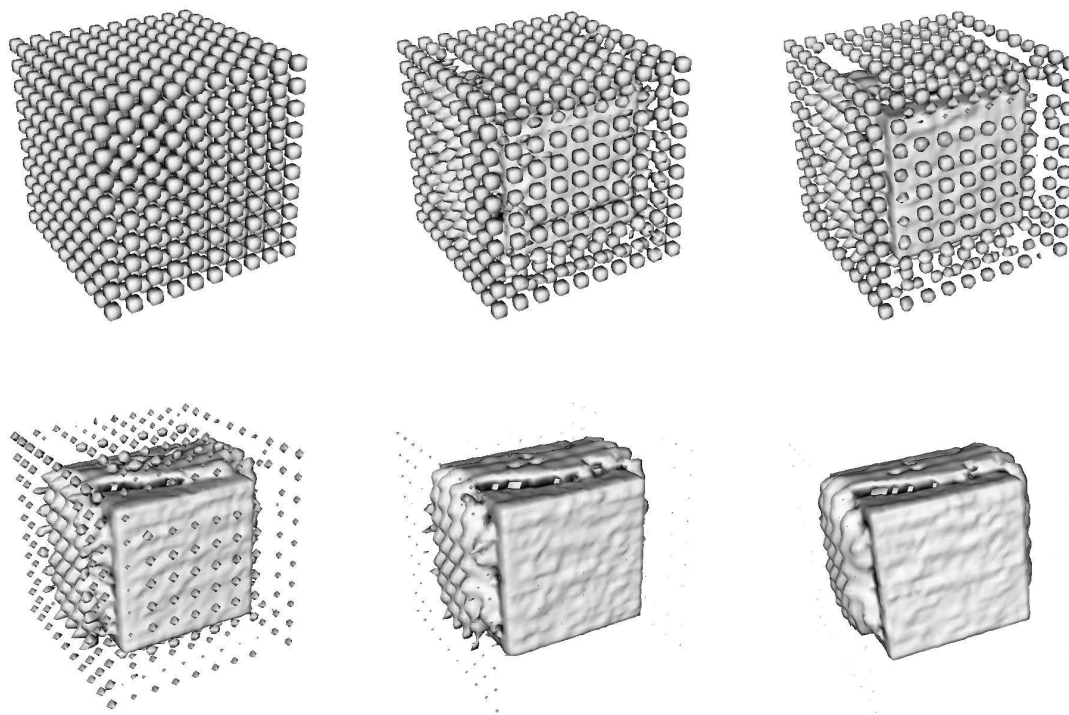


Figure 24: Evolution of surface for simulated three-dimensional cube with gradient descent technique. The top left image shows the initialization of the surface. The evolution of the surface progresses from left to right and from left to right in the second row. Note the suboptimal local minimum on the top and around the sides of the cube.

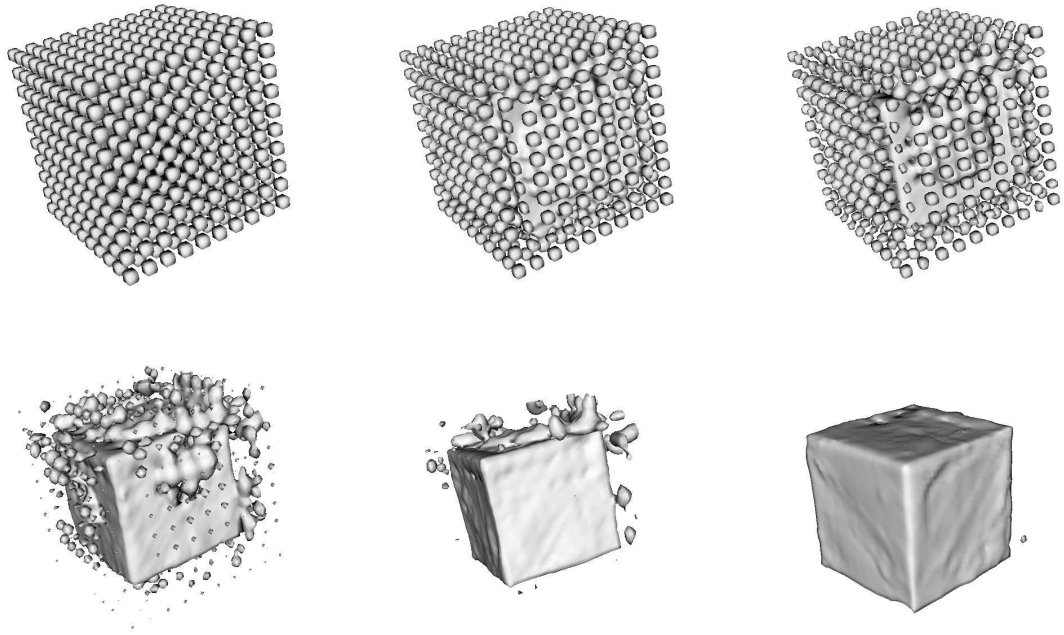


Figure 25: Evolution of surface for simulated three-dimensional cube with proposed multiscale method. The top left image shows the initialization of the surface. The middle and right images in the top row show the surface after 50 and 100 iterations, respectively. The left, middle, and right images in the bottom row show the surface after 200, 300, and 600 iterations respectively. Note the lack of local minima on the surface of the cube.

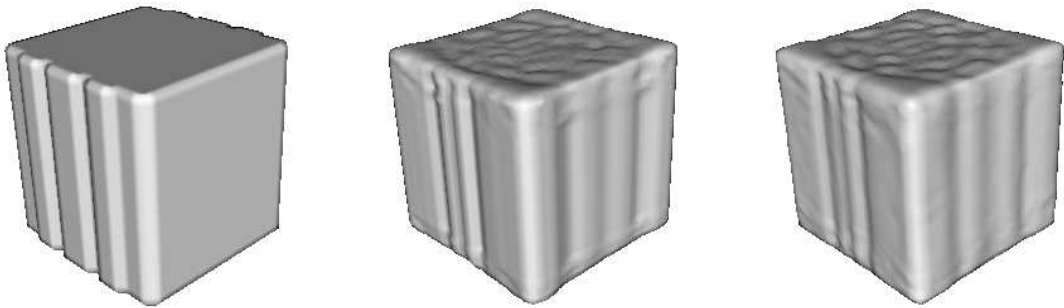


Figure 26: Evolution of the surface starting from bimodal surface initialization method. The left image is the initialization, the middle and right images are the surface after 10 and 40 iterations respectively.

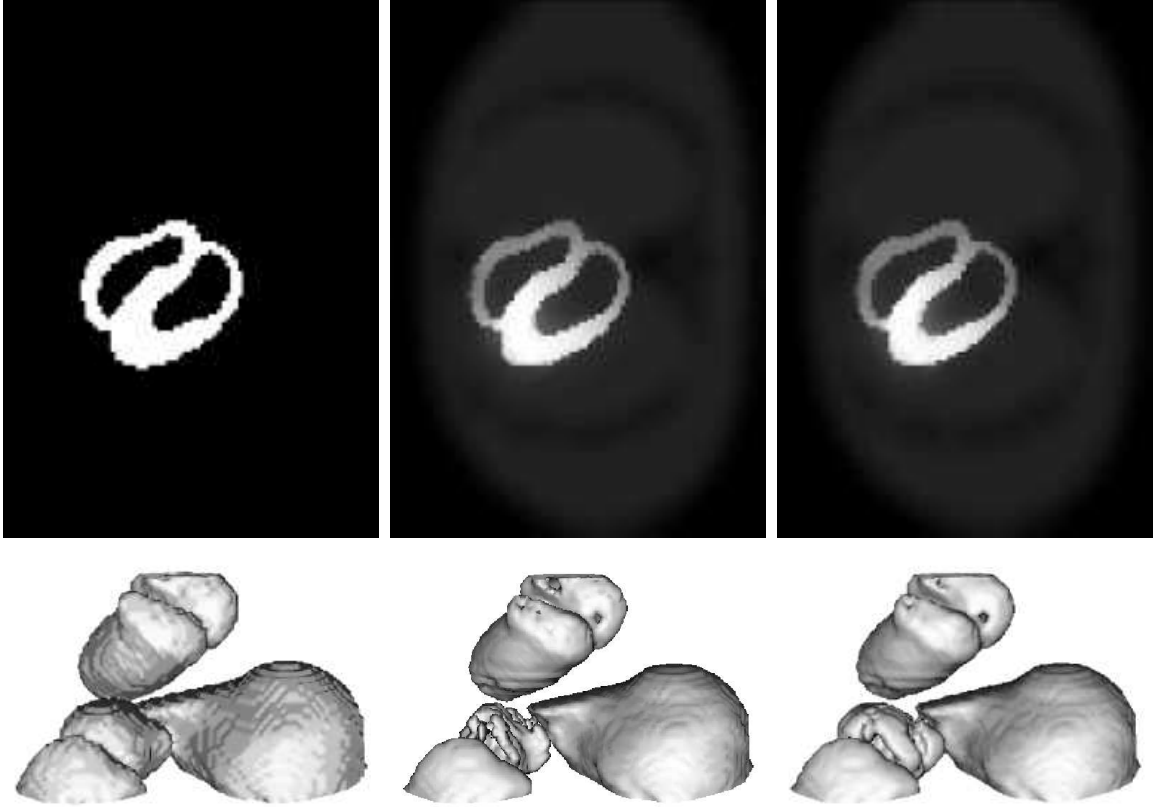


Figure 27: Left column: slice of initial three-dimensional density function using bimodal surface initialization and the corresponding initial surface below it. Middle column: slice of density function during the evolution. Right column: slice of reconstructed density function and its corresponding surface.

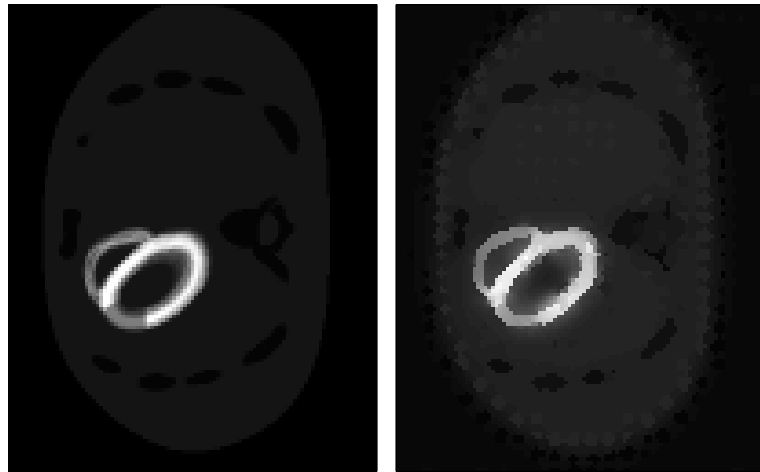


Figure 28: Detection of simulated heart perfusion in a different location using 64 noiseless projections with the proposed reconstruction technique.

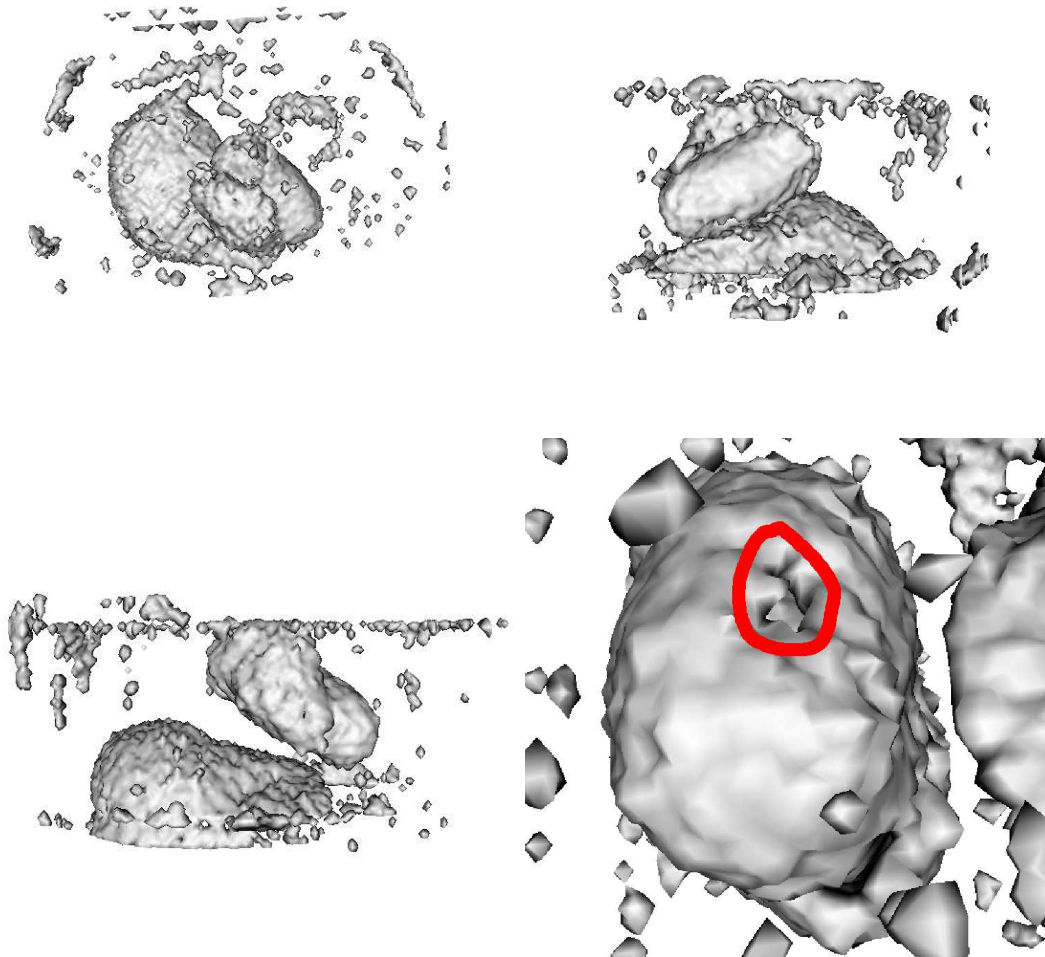


Figure 29: Surface resulting from segmentation produced by the proposed technique when using 64 projections for which noise, attenuation and scatter were added. Bottom right image shows heart perfusion as a hole in the surface.

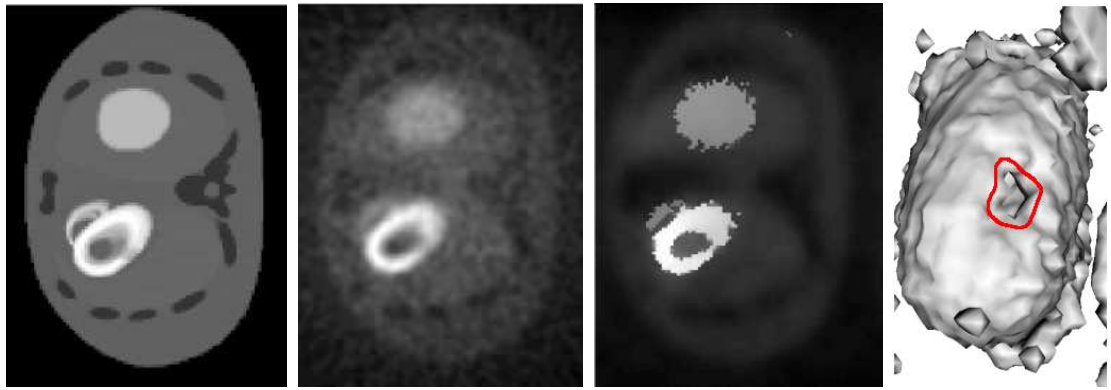


Figure 30: Reconstruction using proposed multiscale method and projections with simulated noise. Left image: slice of the true phantom. Second image: filtered back projection reconstruction. Third image: reconstruction using proposed method. Fourth image: close-up of 3D surface showing heart defect.

CHAPTER VI

MULTIGRID COMPUTATION OF NON-LINEAR OPTICAL FLOW

6.1 *Introduction*

Optical flow is used to extract motion information from a sequence of images. Optical flow offers a valuable quantification for motion such as for object tracking, time-to-impact calculation, or for determining fluid flow in an image. Optical flow fields representing large scale motion, i.e., those that vary smoothly over an image, are well captured by the optical flow method proposed by Horn and Schunck [34]. Horn and Schunck proposed a regularizer that heavily penalizes changes in the optical flow field. The result is that the optical flow functional presented by Horn and Schunck does not preserve the occluding edges of moving objects well because the functional globally enforces a smoothly varying flow field.

Authors have proposed different regularizers in order to capture certain desirable properties of optical flow fields. Hildreth presented a method to ensure smoothness along pre-specified contours [32]. Nagel proposed an oriented smoothness constraint that constrains the optical flow field in directions along which optical flow cannot be determined from gray value changes [49].

Kumar et. al. recommend the use of regularizer whose calculation requires solving a non-linear PDE for the purpose of preserving edges [42]. The functional presented in [42] is not invariant to rotations in the image. A regularization functional that is not invariant to rotation will produce optical flow fields that are dependent on orientation of the image with respect to the x- and y-axis. Unlike the linear partial differential equations (PDEs) resulting from Horn and Schunck's functional, the partial differential equations resulting in [42] are non-linear, and difficult to compute with traditional gradient descent methods.

Gradient descent methods, while simple to implement, are slow to approximate the

coarse scale features of a function. Multigrid relaxation methods, which address this issue, have been of wide interest in the mathematical and image processing communities [11, 12, 68, 1]. Multigrid relaxation methods take advantage of the coarse scale structure of the solution function by solving on a refined, coarser grid, where the computational cost is low [12]. Multigrid techniques therefore compute a coarse scale approximation to the solution quickly. Optical flow fields typically have significant coarse scale structure and thus are well suited to multigrid computation. Glazer applied multigrid techniques to Horn and Schunck’s optical flow functional [27]. Battiti et. al. have introduced an adaptive multiscale approach in the flavor of multigrid for computing Horn and Schunck optical flow [8]. Enkelmann applied multigrid techniques to the optical flow constraint presented by Nagel [22].

In this chapter, an altered functional is presented that is similar to that presented in [42], but is rotationally invariant. We present proof of the rotational invariance of this new functional along with experimental evidence of the improved computational performance. We also compute an optical flow fields that minimize this functional are computed with non-linear multigrid techniques. The increase in computational speed is significant.

6.2 *Optical Flow Functional*

Let $I : \mathbb{R}^3 \rightarrow \mathbb{R}$ represent a time varying image, i.e., $I(x, y, t)$ is the intensity value at spatial location (x, y) and at time t . Let $u(x, y)$ and $v(x, y)$ be the horizontal and vertical components of the optical flow, respectively. Under the assumptions of brightness constancy and smoothly varying intensity patterns, Horn and Schunck [34] motivate the use of an optical flow constraint,

$$I_x u + I_y v + I_t = 0, \tag{61}$$

where I_x , I_y , and I_t denote the partial derivatives of the image function with respect the subscripted variable. Optical flow functions u and v which solve Eq. (61) capture motion in image intensity in directions orthogonal to isobrightness contours.

However, as is well known in the optical flow literature [34, 9, 32], the solution to Eq. (61) is not well-posed since optical flow can not be determined along isobrightness contours. To resolve this ambiguity, authors have presented a regularization in the form of a term that

penalizes the departure from smoothness of the optical flow functions, u and v . [34, 42].

Horn and Schunck defined departure from smoothness as,

$$e_{\text{HS}}(u, v) = \iint_{\text{Image}} u_x^2 + u_y^2 + v_x^2 + v_y^2 dx dy, \quad (62)$$

where subscripted spatial variables again denote partial derivatives. More compactly, we can denote $e_{\text{HS}}(u, v)$ as $e_{\text{HS}}(\mathbf{u})$, where $\mathbf{u} = \begin{pmatrix} u(x, y) \\ v(x, y) \end{pmatrix}$.

By casting the optical flow constraint in Eq. (61) in terms of a functional,

$$e_c(u, v) = \iint_{\text{Image}} (I_x u + I_y v + I_t)^2 dx dy, \quad (63)$$

the problem of solving for optical flow becomes well-posed and we are able to find the u and v that minimize the combined functional, $\lambda e_c + e_{\text{HS}}$. Note that λ is a constant parameter that controls the penalty tradeoff between the optical flow term and the smoothness term.

The smoothness term, e_{HS} , presented by Horn and Schunck has the advantage of being well-behaved numerically since the Euler-Lagrange equations for the functional $\lambda e_c + e_{\text{HS}}$ produce the coupled pair of second-order elliptic partial differential equations (PDEs) given by,

$$\Delta u = \lambda(I_x u + I_y v + I_t)I_x, \quad (64)$$

$$\Delta v = \lambda(I_x u + I_y v + I_t)I_y, \quad (65)$$

where $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ denotes the Laplacian operator.

However, a disadvantage of Horn and Schunck's smoothness term is that it does not allow for the sharp changes in optical flow that occur near edges of moving objects. Thus, while Horn and Schunck's method is useful for optical flow fields which vary smoothly over the entire image, it also has the undesirable effect of smoothing the optical flow over the boundaries of moving objects.

To combat this problem, Kumar et. al. [42] present an edge-preserving smoothness term, defined as,

$$e_K(u, v) = \iint_{\text{Image}} \sqrt{u_x^2 + u_y^2} + \sqrt{v_x^2 + v_y^2} dx dy, \quad (66)$$

as a replacement for Horn and Schunck's smoothness term.

We present an optical flow smoothness term that has similar edge-preservation properties while maintaining the rotational invariance of Horn and Schunck's smoothness term. We propose the use of,

$$e_{\text{RI}}(u, v) = \iint_{\text{Image}} \sqrt{u_x^2 + u_y^2 + v_x^2 + v_y^2} dx dy, \quad (67)$$

In the following discussion, we will define the function, $L = \sqrt{u_x^2 + u_y^2 + v_x^2 + v_y^2}$, i.e., L is the integrand of Eq. (67). As a result from the calculus of variations, the proposed error functional, $\lambda e_c + e_{\text{RI}}$, is minimized by the solution to the non-linear partial differential equations given by,

$$-\frac{\partial}{\partial x} \frac{u_x}{L} - \frac{\partial}{\partial y} \frac{u_y}{L} + \lambda(I_x u + I_y v + I_t)I_x = 0, \quad (68)$$

$$-\frac{\partial}{\partial x} \frac{v_x}{L} - \frac{\partial}{\partial y} \frac{v_y}{L} + \lambda(I_x u + I_y v + I_t)I_y = 0. \quad (69)$$

The computation of Horn and Schunck's optical flow is very fast in comparison to the computation of the optical flow resulting from regularizer in (67). This is due to the parabolic nature of the error surface resulting from Horn and Schunck's regularizer as compared to the cone-like error surface resulting from the regularizer in (67). Thus, the need for increased computational speed is greater when using the regularizer in (67). We suggest the use of non-linear multigrid methods to solve these partial differential equations.

6.2.1 Rotational Invariance

Consider an optical flow field, \mathbf{u} , and a rotation by θ , $\mathbf{R}\mathbf{u}$, where,

$$\mathbf{R} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}, \quad (70)$$

is a two dimensional rotation matrix. Then the rotated optical flow field is given by $\mathbf{u}_r = \mathbf{R}\mathbf{u}$.

We claim that the functional in Eq. (66) is not invariant to rotation and the functional in Eq. (67) is invariant to rotation. To show that the cost functional in Eq. (67) is invariant

to rotation, we substitute the rotated optical flow into the cost functional. We will denote this as $e_{\text{RI}}(\mathbf{u}_r)$. This substitution yields that the integrand of $e_{\text{RI}}(\mathbf{u}_r)$ is,

$$\sqrt{\begin{aligned} &(u_x \cos \theta - v_x \sin \theta)^2 + (u_y \cos \theta - v_y \sin \theta)^2 \\ &+ (u_x \sin \theta + v_x \cos \theta)^2 + (u_y \sin \theta + v_y \cos \theta)^2 \end{aligned}} \quad (71)$$

$$= \sqrt{\begin{aligned} &u_x^2 \cos^2 \theta + v_x^2 \sin^2 \theta + u_y^2 \cos^2 \theta + v_y^2 \sin^2 \theta \\ &+ u_x^2 \sin^2 \theta + v_x^2 \cos^2 \theta + u_y^2 \sin^2 \theta + v_y^2 \cos^2 \theta \end{aligned}} \quad (72)$$

$$= \sqrt{u_x^2 + u_y^2 + v_x^2 + v_y^2}, \quad (73)$$

which is identical to the integrand of the functional without rotation of the optical flow. A similar computation with the function from Eq. (66) results in the integrand,

$$\begin{aligned} &\sqrt{\begin{aligned} &(u_x^2 \cos^2 \theta + v_x^2 \sin^2 \theta - 2u_x v_x \cos \theta \sin \theta) \\ &+ (u_y^2 \cos^2 \theta + v_y^2 \sin^2 \theta - 2u_y v_y \cos \theta \sin \theta) \end{aligned}} \\ &+ \sqrt{\begin{aligned} &(u_x^2 \sin^2 \theta + v_x^2 \cos^2 \theta + 2u_x v_x \cos \theta \sin \theta) \\ &+ (u_y^2 \sin^2 \theta + v_y^2 \cos^2 \theta + 2u_y v_y \cos \theta \sin \theta) \end{aligned}}, \end{aligned} \quad (74)$$

which is only necessarily equal to the original functional when θ is an integer multiple of 90 degrees.

6.3 Computation

6.3.1 Gradient Descent

As a result from the calculus of variations, the left hand side of Eqs. (68) and (69), act as infinite dimensional gradients for the functions u and v . That is by augmenting the functions u and v with an artificial evolution parameter, s , we obtain the evolutions,

$$\frac{\partial u}{\partial s} = -\frac{\partial}{\partial x} \frac{u_x}{L} - \frac{\partial}{\partial y} \frac{u_y}{L} + \lambda(I_x u + I_y + I_t)I_x, \quad (75)$$

$$\frac{\partial v}{\partial s} = -\frac{\partial}{\partial x} \frac{v_x}{L} - \frac{\partial}{\partial y} \frac{v_y}{L} + \lambda(I_x u + I_y + I_t)I_y. \quad (76)$$

We numerically implement this gradient descent using the forward Euler scheme, i.e., at each iteration, by replacing u and v by corrected versions as follows,

$$u^{n+1} = u^n - \gamma \frac{\widehat{\partial u^n}}{\partial s}, \quad (77)$$

$$v^{n+1} = v^n - \gamma \frac{\widehat{\partial v^n}}{\partial s}. \quad (78)$$

Note that the superscript denotes the iteration, γ is a time step parameter, and $\frac{\widehat{\partial u^n}}{\partial s}$ and $\frac{\widehat{\partial v^n}}{\partial s}$ denote numerical approximations to the gradient at step n . We compute these with central difference approximations in the center of the image and use Neumann boundary conditions, i.e., that normal derivatives at the image boundaries are zero.

6.3.2 Multigrid

We solve for the optical flow components u and v using multigrid relaxations methods. Since Eqs. (68) and (69) are nonlinear partial differential equations, it is necessary to use an algorithm that handles non-linear operators, known as the Full Approximation Storage (FAS) algorithm [53].

We recast Eqs. (68) and (69) in the following fashion,

$$-\frac{\partial}{\partial x} \frac{u_x}{L} - \frac{\partial}{\partial y} \frac{u_y}{L} + \lambda(I_x u + I_y v)I_x = -\lambda I_t I_x, \quad (79)$$

$$-\frac{\partial}{\partial x} \frac{v_x}{L} - \frac{\partial}{\partial y} \frac{v_y}{L} + \lambda(I_x u + I_y v)I_y = -\lambda I_t I_y, \quad (80)$$

which we can write compactly as $\mathcal{L}(u, v) = f$, whereby the left hand side of the equation is a non-linear operator on the optical flow functions u and v . It is nonlinear due to the appearance of the L in the denominator.

The algorithm originates with an initial guess of the optical flow functions, \tilde{u} and \tilde{v} . The multigrid method then seeks corrections, \hat{u} and \hat{v} , to the initial guess functions such that,

$$\mathcal{L}(\tilde{u} + \hat{u}, \tilde{v} + \hat{v}) = f. \quad (81)$$

In order to find \hat{u} and \hat{v} , we write,

$$\mathcal{L}(\tilde{u} + \hat{u}, \tilde{v} + \hat{v}) - \mathcal{L}(\tilde{u}, \tilde{v}) = f - \mathcal{L}(\tilde{u}, \tilde{v}) = -r. \quad (82)$$

where r is known as the *residual*. By decimating r , \tilde{u} , and \tilde{v} to a coarser grid, thus obtaining r_c , \tilde{u}_c , and \tilde{v}_c , and by also identifying a coarse grid version of the operator \mathcal{L} as \mathcal{L}_c , we can solve for coarse grid versions of the corrections \hat{u} and \hat{v} by solving,

$$\mathcal{L}_c(\hat{u}_c, \hat{v}_c) = \mathcal{L}_c(\tilde{u}_c, \tilde{v}_c) - r, \quad (83)$$

for \hat{u}_c and \hat{v}_c .

Eq. (83) is solved for \hat{u}_c and \hat{v}_c on the coarsified grid, where the computational cost is lower than on the original grid, using the gradient descent technique described in Section 6.3.1. Once \hat{u}_c and \hat{v}_c are obtained, the corrections \hat{u} and \hat{v} are computed by,

$$\hat{u} = \mathcal{U}(\hat{u}_c - \tilde{u}_c) \quad (84)$$

$$\hat{v} = \mathcal{U}(\hat{v}_c - \tilde{v}_c) \quad (85)$$

where \mathcal{U} denotes an upsampling operator. The initial guess functions are then updated by, $\tilde{u} \leftarrow \tilde{u} + \hat{u}$ and $\tilde{v} \leftarrow \tilde{v} + \hat{v}$, and finally, gradient descent is used on the resulting functions, \tilde{u} and \tilde{v} , in order to obtain the fine scale features of the solution.

The above mentioned algorithm is, in fact, a two grid solution method. However, note that while Eq. (83) can be solved using gradient descent or a similar relaxation technique, it is also in a form similar to Eq. (81) and thus can be solved by the same coarse grid correction process on an even coarser grid where the computational cost is even lower. This process can be carried out until the computational cost of solving with gradient descent on the coarsest grid is negligible. More details about the implementation of multigrid algorithms can be found in [53].

6.4 Simulations

The left side of Fig. (31) shows a frame from the well-known Hamburg Taxi sequence in which two cars, one with high intensity and one with low intensity, are moving in the image, while the rest of the scene remains stationary. The circular window was introduced to show the edge preserving nature of the proposed regularizer.

The right side of Fig. (31) shows the proposed optical flow field on a circular window of the Hamburg Taxi sequence. This image was the result of the FAS multigrid algorithm

explained above using 5 different grids. The original image was 189 pixels by 189 pixels and each successively coarser grid was downsampled by a factor of 2. Thus, the final grid was 5 pixels by 5 pixels. Note the edge preserving nature of the optical flow field around the circular window and the properly captured lack of motion between the two cars.

Figure (32) shows the value of the proposed functional, $\lambda e_c + e_{\text{RI}}$, for the gradient descent algorithm versus time and the value of the same functional for the FAS multigrid algorithm versus time for this frame of the Hamburg Taxi sequence. Note that the FAS multigrid algorithm nears complete convergence within 200 seconds. All computations were performed on a 1.8 GHz computer. The gradient descent algorithm requires several hours (not shown) to approach convergence under the same conditions due to the cone-like error surface mentioned above.

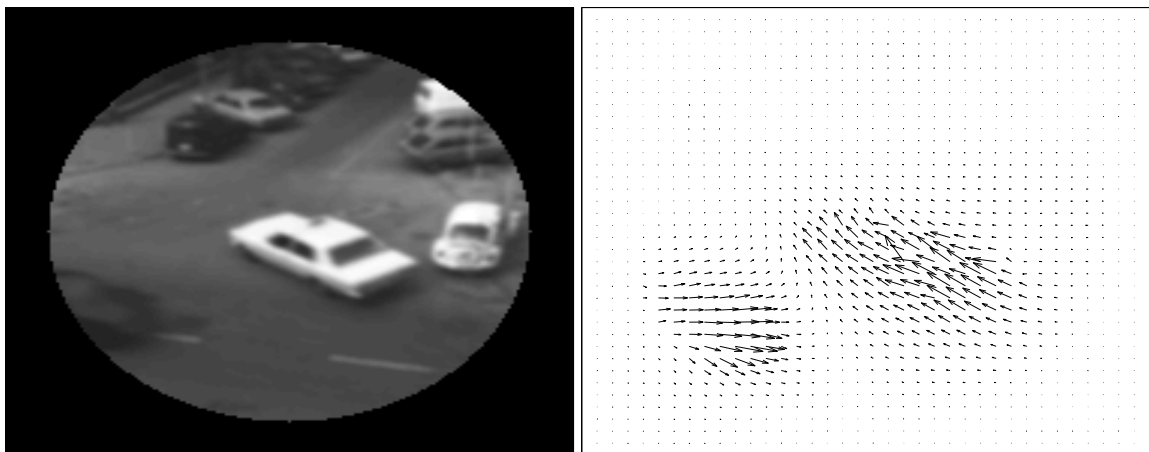


Figure 31: Left: frame from Hamburg Taxi sequence. Right: proposed optical of Hamburg Taxi sequence.

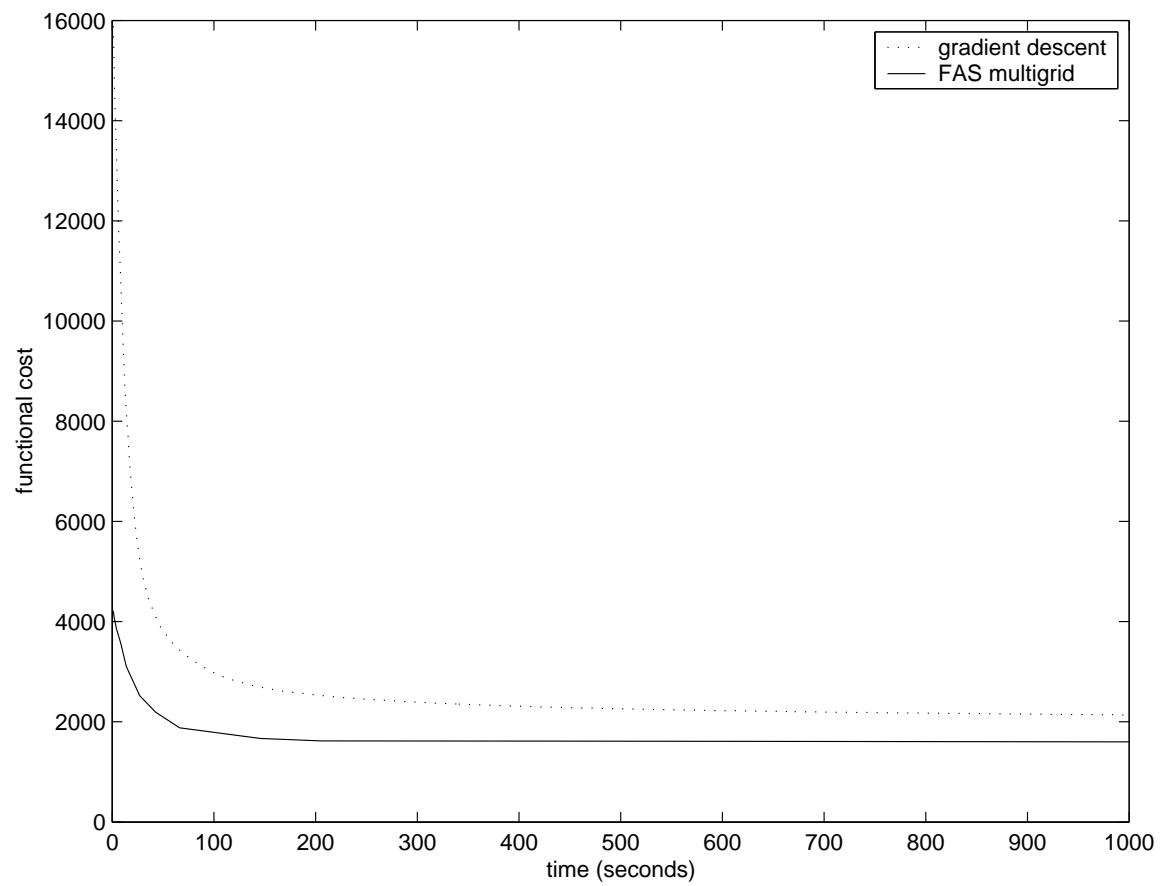


Figure 32: Functional cost versus computation time for the gradient descent and FAS multigrid algorithms.

CHAPTER VII

MINIMAL SURFACES FOR CONTOUR REGISTRATION

7.1 *Introduction*

7.1.1 Proposed Work

Contour registration has been established as a fundamental problem in computer vision and medical imaging; the literature is rich in techniques for registration [13, 43]. In a previously submitted paper, we have introduced a method for registration of contours using minimal surfaces, that arises from developing a natural geometric generalization of the set symmetric difference between the interiors of two sets. In this paper, we will elaborate more of the scale space properties of the proposed method.

We develop a novel method for rigid registration of two dimensional contours that relies on connecting two contours, separated in three dimensional, by a minimal surface. The novel contribution of this paper is not the computation of the minimal surfaces, but rather in the evolution of the registration as prescribed by the family of minimal surfaces. The problem of computing minimal surfaces we performed in a way described in [20]. Once the minimal surface is computed between the two contours, the registration of the top contour is evolved rigidly in an attempt to minimize the surface area of the connecting surface. We will explain the method and its implementation as well as stating its connection to set symmetric difference. We show an example that presents the method as a scale space, with the separation distance between the two contours the scale space parameter.

7.1.2 Past Work in Registration

A significant amount of work has been done in image registration. Reviews of registration methods can be found in [13, 43]. Point based techniques, such as those used by West et. al. and by Zhang [71, 79] attempt to match feature points by finding optimal registrations.

Sebastian, Klein and Kimia have developed a method based on shape outlines, that aligns based on a measure of similarity between the intrinsic properties of the curve such as curvature and length [60]. Feldmar and Ayache discuss rigid and affine registration of free-form surfaces [25]. Hansen and Morse develop a method for multiscale registration using scale trace correlation [29].

The technique proposed in this chapter can be considered a transformation technique, where a one contour is gradually transformed to match the second via a connecting minimal surface.

7.2 Method

7.2.1 Initialization

In the initialization, we will embed two, two dimensional contours into three dimensional space and create a surface between them. We will also define some terminology in this section that will be used throughout the paper.

Let vector functions $\mathbf{C} : [0, 1] \rightarrow \mathbb{R}^2$ and $\hat{\mathbf{C}} : [0, 1] \rightarrow \mathbb{R}^2$ explicitly represent two simple closed contours which we wish to register. We represent these contours implicitly as the 0 level sets of the functions $I : \mathbb{R}^2 \rightarrow \mathbb{R}$ and $\hat{I} : \mathbb{R}^2 \rightarrow \mathbb{R}$, such that $I(\mathbf{C}(s)) = \hat{I}(\hat{\mathbf{C}}(s)) = 0$ for $s \in [0, 1]$. We will construct the level set functions such that $I < 0$ and $\hat{I} < 0$ in the interior of the contours and $I > 0$ and $\hat{I} > 0$ in the exterior of the contours.

We then embed the functions I and \hat{I} into three dimensional space, and thus embed the contours. Let $\Phi_0 : \mathbb{R}^2 \times [0, z_M] \rightarrow \mathbb{R}$ be the three dimensional level set function in which the contours will be embedded. The embedding is represented by setting $\Phi_0(x, y, 0) = I(x, y)$ and $\Phi_0(x, y, z_M) = \hat{I}(x, y)$. The variable $z_M > 0$ has special significance, representing the distance between the embedded contours, will be called the *separation distance*.

To create a surface between the two contours, with the contours as its boundary, we will interpolate between the contours, initializing the function so that,

$$\Phi_0(x, y, z) = (1 - z)I(x, y) + z\hat{I}(x, y), \quad (86)$$

for $z \in (0, z_M)$. The surface, S , is now said to exist at the zero level set of this function Φ_0 , that is, $S = \Phi_0^{-1}(0)$.

We will call the planes in which the contours are embedded the *top* and *bottom contour planes* at $z = z_M$ and at $z = 0$ respectively. Similarly, the *middle contour plane* will refer to the slice of the function Φ_0 when $z = z_M/2$ and the *middle contour* will refer to the zero level set of Φ_0 when $z = z_M/2$.

7.2.2 Evolution to Minimal Surface

Finding the minimal surface between these two contours is equivalent to finding the surface with constant zero mean curvature that has the contours as its boundary. The minimal surface that connects the two contours exists when z_M is sufficiently small. A more detailed discussion of the sufficient condition for existence is discussed in the appendix of this paper.

The initial surfaces described in Section 7.2.1 will not, except in certain degenerate cases, have the property of being minimal surfaces. Throughout this paper it will be necessary to obtain the surface of minimal area that connects the two fixed contours. For this purpose we have chosen to employ an evolution technique explained in [20]. The idea is that by evolving the surface in the direction of the surface normal with speed proportional to the mean curvature, H , we ensure the fastest possible decrease in surface area. Therefore, we wish to evolve the level sets of the function that represents the surface in this fashion.

Consider the function $\Phi : \mathbb{R}^2 \times [0, z_M] \times [0, \infty) \rightarrow \mathbb{R}$, whose first three arguments are the spatial variables x , y , and z and whose last argument is the temporal variable t . The temporal variable t will be used to state the surface area minimization evolution as a partial differential equation. The initial value of Φ is set to the initialization in Section 7.2.1, that is, $\Phi(x, y, z, 0) = \Phi_0(x, y, z)$.

When we evolve $\Phi(x, y, z, t)$ according to the non-linear partial differential equation,

$$\frac{\partial \Phi}{\partial t} = H \|\nabla \Phi\|, \quad (87)$$

to guarantee that the surface area of each level set is decreasing as quickly as possible. In order to not keep the contours fixed throughout this evolution, we do not evolve the function Φ at $z = 0$ or at $z = z_M$, but we do evolve at every value of z in between.

The mean curvature, H , of the surface is given by a function of derivatives of Φ ,

$$H = \frac{\left(\Phi_{xx}(\Phi_y^2 + \Phi_z^2) + \Phi_{yy}(\Phi_x^2 + \Phi_z^2) + \Phi_{zz}(\Phi_x^2 + \Phi_y^2) - 2\Phi_{xy}\Phi_x\Phi_y - 2\Phi_{yz}\Phi_y\Phi_z - 2\Phi_{xz}\Phi_x\Phi_z \right)}{2(\Phi_x^2 + \Phi_y^2 + \Phi_z^2)^{3/2}}, \quad (88)$$

as explained in [20].

Evolving the function Φ by Eq. 87 ensures that the zero level set approaches the minimal surface that connects the two contours. Fig. 33 shows the surface connecting two stationary circular contours at various stages of this area minimizing evolution.

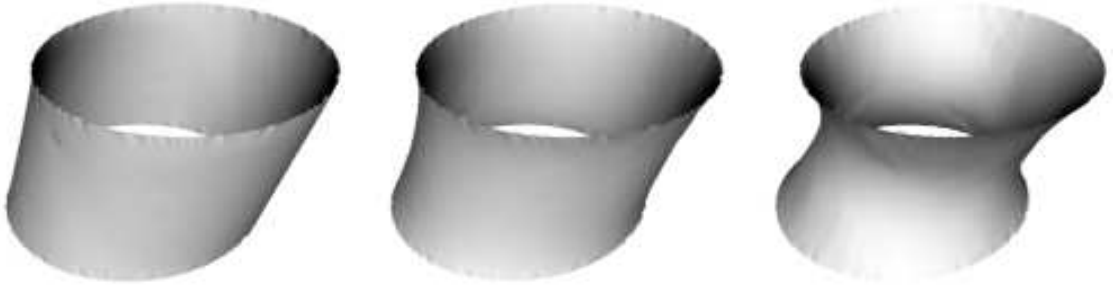


Figure 33: Oblique view of three different stages of surface minimizing evolution for stationary shifted circular contours. Surface created from initialization (left), surface after 125 seconds of surface evolution (center), and final minimal surface (right).

7.2.3 Gradient Evolution for Rigid Registration

The previous two sections described the initialization of a minimal surface evolution between two contours. This section describes how we rigidly register the top contour using the minimal surface. Since we wish to obtain a two dimensional rigid registration, we evolve the top contour rigidly within the top contour plane to ensure the most rapid decrease in area of the minimal surface, while holding the bottom contour fixed. This will be done using two projections.

7.2.3.1 Unconstrained Top Contour Evolution

Assume, temporarily, that the top contour were not constrained to remain within the top contour plane. If the goal of the evolution was to minimize surface area, the contour at a

given point would move in a direction downward along the surface, orthogonal to both the tangent vector to the contour, \mathbf{T} , and the surface normal, \mathbf{N} .

More explicitly, the outward unit normal to the surface is $\mathbf{N} = (N_x, N_y, N_z) = \nabla\Phi$. The normal vector to the contour \mathbf{n} is the projection of \mathbf{N} onto the plane of the top contour. It is given by $\mathbf{n} = (N_x, N_y)/\sqrt{N_x^2 + N_y^2}$. The tangent vector to the contour is in the plane of the top contour and is a 90 degree rotation of \mathbf{n} , i.e., $\mathbf{T} = (N_y, -N_x)/\sqrt{N_x^2 + N_y^2}$. The vector, \mathbf{V} , that is orthogonal to both \mathbf{T} and \mathbf{N} , and points along the direction of the surface, can be found by computing the vector cross-product of \mathbf{T} with \mathbf{N} ,

$$\mathbf{V} = \frac{(-N_x N_z, -N_y N_z, N_x^2 + N_y^2)}{\sqrt{N_x^2 + N_y^2}}. \quad (89)$$

7.2.3.2 Projection to Top Contour Plane

The vector \mathbf{V} shows the direction in which the contour would move in 3D space to minimize surface area most rapidly. Constraining the contour to remain in the top contour plane yields the explicit contour evolution, $(\mathbf{V} \cdot \mathbf{n})\mathbf{n} = -N_z \mathbf{n}$.

7.2.3.3 Projection to Rigid Registration

By projecting the explicit contour evolution $-N_z \mathbf{n}$, onto the group of rigid motions, we ensure that the top contour's shape and size do not change.

Define a rigid motion, $g : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, by,

$$g(\hat{\mathbf{C}}) = R(\hat{\mathbf{C}} - \mathbf{m}) + T + \mathbf{m}, \quad (90)$$

where R is the rotation matrix that characterizes rotation around the point \mathbf{m} . R is parameterized by rotation angle θ . T is the translation vector parameterized by T^x and T^y , the x and y components of the translation.

We can convert the explicit evolution of the contour, \mathbf{C}_t , into an evolution of the rigid motion parameters, θ , T^x , and T^y . This is done by evolving the rigid group parameters to maximize the inner product

$$J = \int_0^L (-N_z \mathbf{n} \cdot \mathbf{n}) (\mathbf{g} \cdot \mathbf{n}) ds, \quad (91)$$

where ds is the arc length element and L is the length of the contour.

In this fashion we obtain a gradient evolution for the rigid group parameters that ensures motion of the contour in the direction that minimizes surface area. They are,

$$\theta_t = J_\theta = \int_0^L (-N_z \mathbf{n} \cdot \mathbf{n}) (\mathbf{g}_\theta \cdot \mathbf{n}) ds = - \int_0^L N_z (\mathbf{g}_\theta \cdot \mathbf{n}) ds \quad (92)$$

$$(T^x)_t = J_{T^x} = \int_0^L (-N_z \mathbf{n} \cdot \mathbf{n}) (\mathbf{g}_{T^x} \cdot \mathbf{n}) ds = - \int_0^L N_z \frac{N_x}{\sqrt{N_x^2 + N_y^2}} ds \quad (93)$$

$$(T^y)_t = J_{T^y} = \int_0^L (-N_z \mathbf{n} \cdot \mathbf{n}) (\mathbf{g}_{T^y} \cdot \mathbf{n}) ds = - \int_0^L N_z \frac{N_y}{\sqrt{N_x^2 + N_y^2}} ds, \quad (94)$$

where,

$$(\mathbf{g}_\theta \cdot \mathbf{n}) = \frac{1}{\sqrt{N_x^2 + N_y^2}} \begin{bmatrix} -\sin \theta(x - m_x) - \cos \theta(y - m_y) \\ \cos \theta(x - m_x) - \sin \theta(y - m_y) \end{bmatrix} \cdot \begin{bmatrix} N_x \\ N_y \end{bmatrix}. \quad (95)$$

Evolution of the rigid registration parameters in this fashion will change the top contour. In the proposed registration method, we evolve the registration of the top contour by equations (92), (93), and (94) while ensuring that the surface continues to stay minimal by equation (87). Continued evolution of the top contour in this fashion, while ensuring that the connecting surface remains minimal, produces a rigid registration of the contour $\hat{\mathbf{C}}$, summarizing our proposed method.

In Fig. 34 we show an example of the alignment of two misaligned rectangles while constraining the registration to consist of only rotations and translations, as described in this section. The correct registration of these two shapes is a 30 degree rotation followed by a 10 pixel translation in the x direction and a 20 pixel translation in the y direction. The proposed minimal surface method produces a registration of 29.3 degrees followed by an x translation of 9.8 pixels and a y translation of 19.6 pixels, accurate to within half of a pixel.

7.2.4 Motivation

In this section we will motivate the proposed method by explaining its connection to set symmetric difference registration. In fact, we will show that set symmetric difference registration is a special case of the proposed method. The cost functional in set symmetric

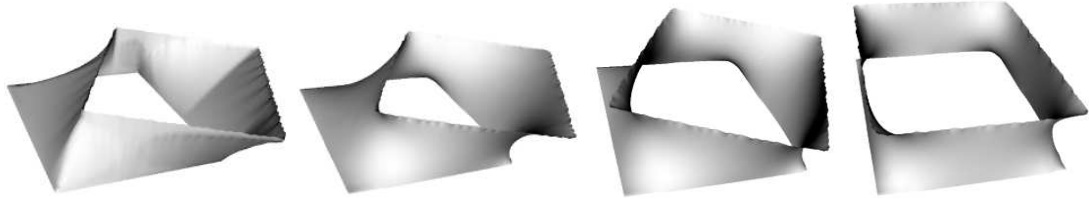


Figure 34: An example of rigid registration. Initial surface between two unregistered rectangles (left), minimal surface between unregistered rectangles (second from left), surface during rigid registration (second from right), and surface after rigid registration (right).

difference registration is, as the name suggests, the mismatch between the interiors of the two contours. We will explain how this mismatch generalizes to surface area, thus motivating surface area as the new cost functional. In addition, we will explain how the proposed method eliminates local minima that occur when registering by minimizing SSD, thus justifying the proposed method as a useful generalization. We show an example of the proposed method avoiding a registration local minimum.

7.2.4.1 Set Symmetric Difference

Let A and B be sets that are the interiors of two contours. The SSD of A and B is, $A \cup B - A \cap B$, that is, all points in the union that are not in the intersection. The area of the SSD gives a measure for how poorly two contours are aligned. Perfect alignment of two contours yields an area of 0. Very poorly aligned contours yield very high SSD areas. These properties make it a natural cost functional to measure the alignment between two contours.

The minimal surface method proposed in this paper is a generalization of SSD registration. This can be seen by imagining the set symmetric difference to comprise the area of a surface as the two contours are pulled apart in space.

Therefore, SSD registration can be viewed as a special case of the proposed method, when the separation distance, $z_M = 0$, in which the area of the connecting surface simplifies to the area of the SSD.

7.2.4.2 Elimination of Local Minima

Evolving to minimize the area of the SSD will produce usable results when the interiors of the contours are convex. However, certain contours with non-convex interiors develop local minima when set symmetric difference registration is used. This happens when the registration must pass through a region of higher SSD area in order to reach the minimum SSD area. Since the evolution method will not pass through this region of higher SSD area, the resulting registration is a suboptimal solution where SSD area is locally minimized but not globally minimized. We will refer to this as a local minimum.

Minimal surfaces, having constant zero mean curvature, are smooth. In addition, as the separation distance z_M increases, the intermediate contours between $z = 0$ and $z = z_M$, i.e., cross sections of the surface in planes parallel to the contour planes, become smoother. The surface thus acts as an interface between the two contours that eliminates some of the fine scale structure of the original contours. Fig. 35 shows this smoothing effect. On the left is the original contour that has six protrusions of varying length surrounding a circular center. In the middle is the minimal surface connecting this contour with an identical replica of itself, each of the two contours embedded in their respective contour planes. The right contour is the middle contour for this surface, that is, the zero level set of $\Phi(x, y, z_M/2)$. Notice that, in the middle contour, much of the fine scale structure was eliminated by the smoothing of the mean curvature flow.

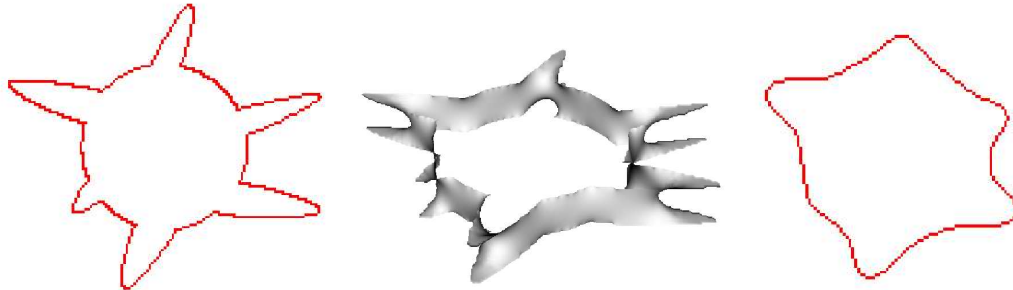


Figure 35: An illustration of the smoothing effect of minimal surface on the intermediate contours. Original contour (left), minimal surface connecting this contour with replica of itself (center), middle contour of minimal surface (right).

As the separation distance, z_M , is increased, the middle contour of the surface becomes more circular and the interface between the two contours becomes more coarse, i.e., the fine scale structure that can produce local minima is eliminated. As a result, the registration evolution may pass through regions in the registration space that would have produced higher surface area for low values of the separation distance. Figs. 36 and 37 illustrate this concept. Fig. 36 shows an “E”-shaped contour which is of dimension 140 voxels by 90 voxels. In the center, it shows the minimal surface connecting this contour with a translated replica of itself, when the separation distance is $z_M = 12$ voxels. Note that the surface connects the legs of the “E”. On the right is the final registration of this contour, which results in an incorrect alignment. The contour does not get to the correct misaligning the legs of the “E” would produce a surface with much higher area, thus making the incorrect alignment on the right a local minimum. Note that SSD registration behaves in a similar fashion, reaching a local minimum in the registration space.

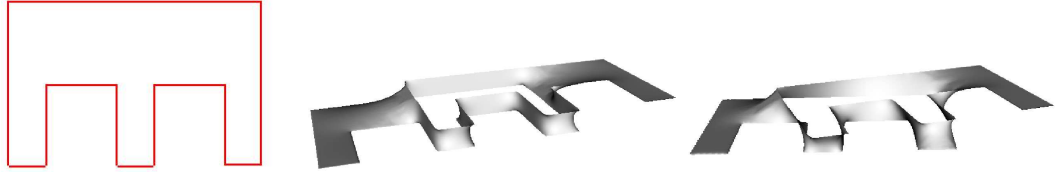


Figure 36: Contour and registration local minimum. “E”-shaped contour (left), minimal surface connecting contour and translated replica of itself (center), and registration local minimum (right).

Fig. 37 shows the proposed method when the separation between the two contours is $z_M = 25$. On the left is the minimal surface between the contour and its translated replica. Notice in this case how the minimal surface does not directly connect the legs of the “E”, but instead smoothly connects by curving inward, removing the fine scale legs for contours in between the top and bottom. This smooth, inward curving connection allows the registration to pass through the position that was unattainable when the separation distance was too low due to the existence of the fine scale features. The registration passing through this intermediate region is shown in the center of Fig. 37. On the right is the final

correct registration, showing how the method overcame the local minimum.

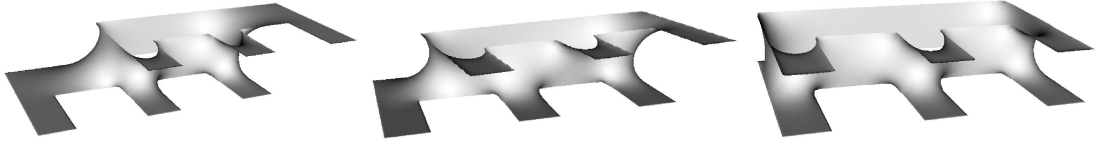


Figure 37: Registration overcoming local minimum by increasing separation distance to 25. Minimal surface connecting contour and translated replica (left), registration overcoming local minimum (center), and final correct registration of top contour (right).

7.3 *Scale Space*

7.3.1 Scale Space Parameter

The natural scale space parameter for this registration method is the separation distance between the two contour planes. As this height becomes smaller, the proposed method behaves more like set symmetric difference registration, and thus the finest scale information in the contour is represented. As this height becomes larger, less fine scale information about the contour is considered. This method acts as a viable scale space because the connection between the contours, namely the surface, becomes smoother as the distance between the contour planes increases.

7.3.2 Extreme Cases

The two extreme cases of the scale parameter help us understand this method more clearly. As the separation distance approaches zero, this method behaves more closely to set-symmetric difference registration. When the separation distance is too small, the minimal surface closely resembles the shape of the original contours, and the method behaves similarly to SSD registration.

When the separation distance is very large¹, the minimal surface that connects the two contours will fail to exist. Each pair of contours has a critical distance at which the connecting surface exists but won't exist for higher separation distances. This critical distance is the coarsest possible scale at which registration using the proposed method can

¹larger than 1.325 times the circle exscribing the contour

be performed. When the scale parameter is at its critical value, we expect a low number of registration local minima. In the next section we experimentally support these claims using an example with structure at varying scales. When the separation distance, z_M is greater than $1.325r_{\text{ex}}$, where r_{ex} is the radius of the exscribed circle, the minimal surface will not exist for any contours. When the separation distance is less than $1.325r_{\text{in}}$, where r_{in} is the radius of the inscribed circle, a connecting minimal surface is guaranteed to exist by the maximum principle of minimal surfaces. This will be proved and further discussed in Appendix C. Thus the region of interest for the separation distance is in the range from 0 to $1.325r_{\text{in}}$; in this range, registration can be performed with the proposed method.

7.3.3 Local Minima Decreasing

In this section we will show an example that has multiple local minima at different scale levels that will serve as experimental support that this method is a scale space in the separation distance parameter.

Fig. 38 shows the example contour, that has a multiscale structure. The finest scale features of the contours are the 8 long extended thin protrusions. These thin protrusions create local minima while registering with set symmetric difference or when the separation distance is very low. The intermediate scale features of the contour are the four stems from which the thin protrusions extend. They will create local minima in the presented method when the separation distance is small or medium, but not always when the separation distance is large. The coarsest scale feature of the contour is the circular middle that would remain after several iterations of curvature flow on the contour. A few example of the local minima produced by this method are displayed in Fig. 39.

Fig. 40 shows a plot of the number of local minima as a function of the separation distance between the two contours for selected separation distances. At the highest separation distances, there are the fewest number of local minima and at the lowest separation distances there are the highest number of local minima. Thus is it experimentally verified that the presented method is a reasonable scale space method.

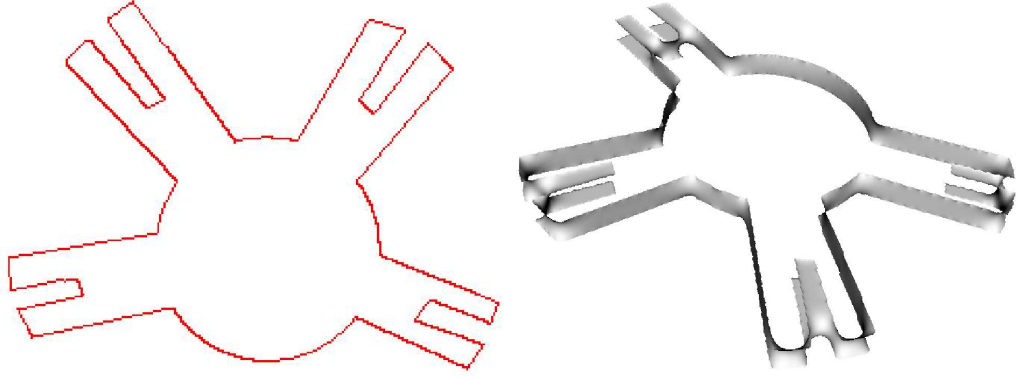


Figure 38: Contour with features at multiple scales (left) and minimal surface resulting from correct registration of contour with a separation distance of 10 (right).

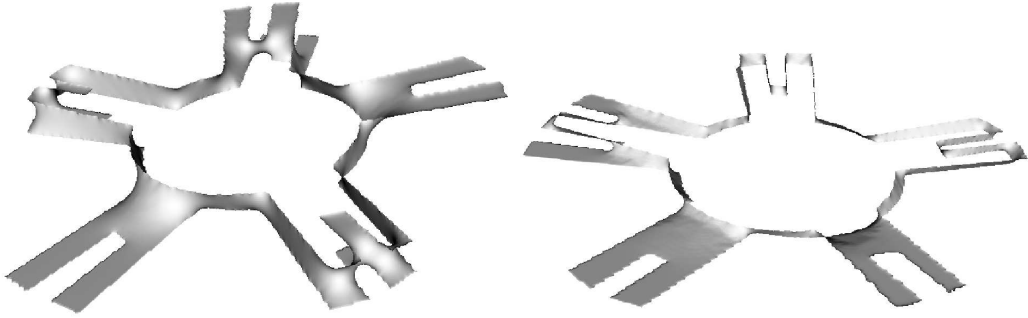


Figure 39: Two local minima for contour with multiple scale features. The method can not correct for incorrect initial alignment due to local minima. The separation distances for the surfaces 5 and 10 for the left and right surfaces respectively.

7.4 *Implementation*

In this section we explain appropriate implementation details of the proposed method. First, we will discuss the level set representation of the surface and its details. The level set representation and mean curvature flow of the surface is nearly identical to the method of Chopp in [20], with noted exceptions. Then we will discuss the implementation of the partial differential equations. Finally, we will explain the simultaneous evolution of the registration and minimal surface.

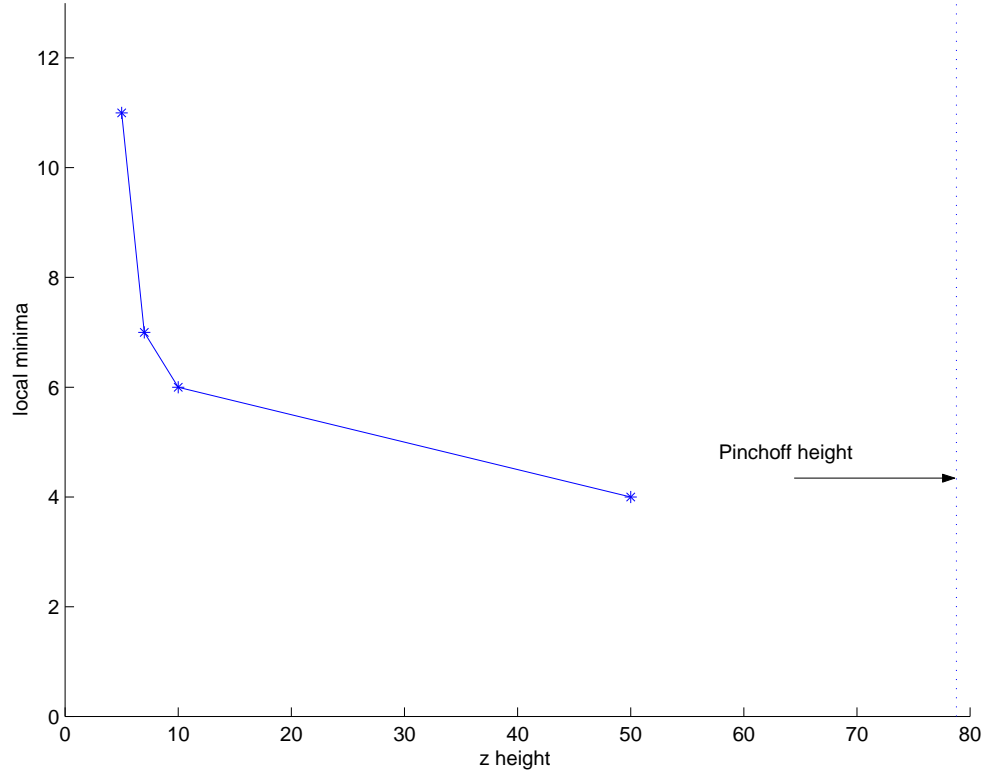


Figure 40: Number of local minima as a function of the separation distance between the two contours for the contour in Fig. 38. On the right is shown the pinch-off distance, i.e., the separation distance at which the minimal surface fails to exist.

7.4.1 Level Set Representation of Surface

As explained in Section 7.2, we implicitly represent the contours and the surface that connects the contours as the level set of the function $\Phi(x, y, z, t)$. To create the minimal surface, we evolve the Φ function with mean curvature flow as stated in Equation (87). Using level sets to represent the curve and surface implicitly allows us to handle natural topology changes of the surface that occur during mean curvature flow [20].

We represent the interior of the surface as negative level set values and the exterior of the surface as positive level set values. This enables us to get the outward surface normal vector when we compute the gradient, $\nabla\Phi$, at any point on the surface.

Numerical problems can develop if we simply evolve the function Φ . The reason for these numerical problems is that surrounding level sets to the surface can bunch up or spread out.

We are primarily interested in what happens to the surface, i.e, the zero level set. Therefore, after each evolution step of the function Φ , in order to avoid these numerical problems, we recalculate the values surrounding the surface in order to maintain equidistant level sets. To do this, we replace the values of the function at points that surround the surface by a value representing the distance to the nearest surface point. This keeps the level sets equally spaced and ensures a numerically well-behaved flow. We then multiply all points interior to the surface by -1, so that the sign convention of the level set function is preserved. This technique, as seen in [20], is known as the signed distance transform.

Although we use the function Φ to represent the surface, we do not store Φ on the entire grid. Since we are only interested in the evolution of the surface and not of the entire Φ function, we only store the values of Φ on a narrow band near the zero level set. Only storing values on this narrow band reduces computational cost significantly [62].

7.4.2 Freezing of Top and Bottom Contours

When we evolve the surface with Equation (87) in order to minimize surface area, we want to keep the top and bottom contours constant. Unlike the method proposed in [20], we freeze the top and bottom contours by simply not evolving the Φ function when $z = 0$ or when $z = z_M$. We still recompute the level set functions on the top and bottom contours with the signed distance transform, as mentioned above, to keep the level sets regularly spaced. Although Chopp’s method, enforcing chains where the fixed contours lie, is more general, we do not need such generality since the contours are always in the top and bottom planes.

7.4.3 PDE Implementation

7.4.3.1 Spatial Derivatives

Implementation of this work relies on calculation of spatial derivatives on the discrete grids. On the interior of the three dimensional region, we estimate spatial derivatives using the central difference approximation. On the boundaries of the three dimensional region we use the one-sided difference approximation.

7.4.3.2 Mean Curvature Flow

The temporal derivatives in Equation (87) is implemented with Euler's method. Euler's method is equivalent to estimating the time derivative by its one-sided finite difference approximation, that is,

$$\Phi_t = (\Phi(x, y, z, t + \Delta t) - \Phi(x, y, z, t)) / \Delta t. \quad (96)$$

Substituting Φ_t with the evolution in Equation (87) and rearranging, we obtain the update equation for Φ ,

$$\Phi(x, y, z, t + \Delta t) = \Phi(x, y, z, t) + \Delta t H(x, y, z, t) \|\nabla \Phi(x, y, z, t)\|. \quad (97)$$

Note that the Δt parameter is known as the time step, and is chosen as large as possible, as long as the flow will remain stable and well behaved. Also note that the implementation of Equation (97) is spatially discrete, that is, the mean curvature is computed with discrete spatial derivatives as mentioned above.

We evolve the surface with mean curvature flow, i.e., the flow that minimizes surface area, until the surface area decreases slowly enough to be considered nearly minimal. We stop the mean curvature flow when the difference in surface area between two successive time steps of Equation (97) is less than 0.01. In this manner, we ensure that the surface connecting the two contours is nearly minimal.

7.4.3.3 Registration Flow

The proposed rigid registration method relies on the surface already being minimal. Once the surface is minimal, as described in the previous section, we evolve the registration parameters by Equations (92), (93), and (94). We evolve these parameters with the forward Euler method, obtaining,

$$\theta(n+1) = \theta(n) + \eta J_\theta \quad (98)$$

$$T^x(n+1) = T^x(n) + \eta J_{T^x} \quad (99)$$

$$T^y(n+1) = T^y(n) + \eta J_{T^y}. \quad (100)$$

Each time we update the rigid registration parameters, we are moving the top contour to reflect this movement. If we move the top contour too much, the contour will no longer be connected to the surface. This is undesirable in our proposed method. Therefore, η is often scaled to prevent the contour becoming disconnected from the surface.

Note that here we use the letter n to denote the time step and not t as in Equation (97). This is because of the nesting of the algorithm as will be described in the next section.

7.4.3.4 *Simultaneous Evolution Algorithm*

The algorithm we used in our proposed method is, to first initialize by embedding the contours within their respective contour planes, then to initialize and minimize the connecting surface, then to modify the registration parameters while ensuring that the surface remains minimal. The structure of this algorithm is,

1. Embed contours in respective contour planes
2. Initialize surface
3. Evolve surface with mean curvature flow until minimized
4. Evolve registration parameters once
5. Go to step 3 if registration is not converged.

Step 3 of this algorithm involves evolving with mean curvature flow as per Equation (97). This evolution happens separately from the evolutions of the registration parameters, and thus they are denoted with time variable t . Each instance of step 3 requires multiple iterations of Equation (97). Step 4 of the algorithm happens when the surface is minimal and only occurs once per surface minimization. By nesting the surface minimization within the evolution of the registration parameters, we achieve the desired effect of always having a minimal surface guiding the motion of the registration parameters.

7.5 *Conclusion*

We have presented a novel method for registration of two dimensional images that not only a geometric extension of set symmetric difference registration, but is also a scale space. The method rigidly registers contours by evolving both the registration of a contour and

the surface that connects them by minimizing a unified energy functional, i.e., surface area. We have explained the method in theory and its implementation and shown a number of examples. One particular example showed how the number of local minima decreased with increasing scale parameter. We have also identified a sufficient condition for existence of the minimal surface that connects two contours.

CHAPTER VIII

FAST MUMFORD-SHAH SEGMENTATION USING IMAGE SCALE SPACE BASES

8.1 *Introduction*

Image segmentation involves extracting the boundaries of one or more objects in an image. Several active contour methods for image segmentation have been proposed beginning with the seminal work of Kass et al. [39]. Many works have appeared based on geometric models for active contours [15, 16, 44, 73] including edge-based models [39, 44] and region-based models.

Mumford and Shah presented the variational problem for piecewise smooth segmentation [47, 48]. Several implementations of Mumford-Shah segmentation have been presented [17, 70]. Tsai et al. report applications of solving the Mumford-Shah functional for image denoising, interpolation and magnification [70]. The robustness of the Mumford-Shah model is well-known. However, so is its high computational cost.

In this chapter we present a fast method for approximate Mumford-Shah segmentation that relies on representing the image regions via a reduced image basis. The idea of using a reduced image basis is that it can capture most of the accuracy of a full Mumford-Shah segmentation, at a fraction of the computational cost. In the implementation we compare between two different bases that both come from scale spaces of smoothed version of the image. One scale space is the space of images smoothed by the linear heat equation. One scale space is the space of anisotropic diffusions presented by Perona and Malik [52].

This method can also be used as a very effective method of finding the initial contours for full basis Mumford-Shah segmentation, thus yielding an algorithm that converges quickly and very accurately obtains the piecewise smooth image functions.

8.2 Theory

In this section, we will first discuss the image segmentation model proposed by Mumford and Shah [48]. We will discuss how this model has typically been implemented assuming a basis whose dimension is equal to the number of pixels in the image. We will then discuss the proposed fast approximation to computing Mumford-Shah segmentation using a reduced basis. Finally we will discuss the generalization of the proposed method to arbitrary multiple regions.

8.2.1 Mumford-Shah Model

The variational segmentation model proposed by Mumford and Shah assumes that the image is a piecewise smooth function, i.e., a function that is smooth within regions but not necessarily across the boundaries of these regions. The proposed cost functional simultaneously penalizes the function for deviating from the image, penalizes the function for deviating from smoothness, and penalizes the length of the boundary contour. The appropriate energy functional is,

$$\begin{aligned}
 E = & \sum_{i=1}^N \int_{R_i} (f_{R_i} - I)^2 d\bar{x} \\
 & + \alpha \sum_{i=1}^N \int_{R_i} \|\nabla f_{R_i}\|^2 d\bar{x} \\
 & + \beta L,
 \end{aligned} \tag{101}$$

where I is the image, R_i is the i th open subset of the image domain, N is the total number of such regions, f_{R_i} is the restriction of the model function, f , to the region R_i , $d\bar{x}$ is the image domain area element, and $\|\nabla g\|$ is the Euclidean norm of the gradient of the function g . Note that the opens regions, R_i , are always such that the R_i 's are disjoint from one another, i.e., $R_i \cap R_j = \emptyset$ for $i \neq j$, and the set formed by the boundaries of these regions is a contour, C with total length, L . Finally, α and β are weighting parameters designed to control the penalty tradeoff between the three terms.

Note that the first two terms in this functional involve both the function, f , and, implicitly, the contour, C . Proper derivation of the first variation for these two terms produces

terms that depend on the evolution of the contour since the regions, and hence the boundary contour, are allowed to deform. The third term does not depend on the function, f , but only the contour, C .

Evolution in the direction opposite to the gradient ensures that the cost functional is decreasing as quickly as possible. The evolution that does this involves the image functions and the contour as follows,

$$\frac{\partial f_{R_i}}{\partial t} = -2(f_{R_i} - I) + \Delta f_{R_i} \quad (102)$$

$$\begin{aligned} \frac{\partial C}{\partial t} = & ((I - f_{R_a})^2 - (I - f_{R_b})^2) \vec{\mathcal{N}} \\ & + \alpha (\|\nabla f_{R_a}\|^2 - \|\nabla f_{R_b}\|^2) \vec{\mathcal{N}} \\ & + \beta \kappa \vec{\mathcal{N}} \end{aligned} \quad (103)$$

where t is an artificial evolution time parameter, R_a is locally the region on one side of the contour and R_b the local region on the other side of the contour, $\vec{\mathcal{N}}$ is the local unit normal vector to the contour, and κ is the local curvature of the contour.

In theory, evolution in this manner that is continuous both spatially and in evolution time, would reach a local minimum that depends on the initialization of the contour. In practice however, some finite basis must be chosen to represent the image functions, f_{R_1} through f_{R_N} . Most implementations of this technique allow each image pixel to assume its own value, i.e., they assume that the function basis has dimension equal to the number of pixels in the image and that the basis itself consists of functions,

$$\delta_{(u,v)}(x,y) = \begin{cases} 1 & \text{if } x = u \text{ and } y = v \\ 0 & \text{otherwise} \end{cases}, \quad (104)$$

for all image points, (x,y) . In this manner, the function f can be seen as,

$$f_{R_i}(x,y) = \sum_{(u,v) \in R_i} c_{(u,v)} \delta_{(u,v)}(x,y), \quad (105)$$

and evolution of the function f_{R_i} corresponds to evolution of each of the basis weights, $c_{(u,v)}$, for each region.

Unfortunately, implementation of these evolution functions is very computationally expensive, even when the functions, f_{R_i} for $1 \leq i \leq N$ are only evolved partially before each

contour evolution step. This technique is similar to the implementation technique used in [70]. However, it is typically superfluous to allow each pixel to vary independently since the image regions typically have a certain smooth structure.

8.2.2 Proposed Function Model

The proposed technique is to choose a basis for the functions, f_{R_i} , that is significantly smaller than the pixel-by-pixel basis mentioned in the previous section but also contains enough resemblance to the functions f_{R_i} that are obtained when the pixel-by-pixel basis is used and the evolution of the function reaches its final solution. The intention is that the evolution of the contour will precede much in the same fashion as when the pixel-by-pixel basis is used but the problem of finding a local minimizer of the cost functional when the contour remains fixed will become much less computationally expensive. The choice of which basis to use is not obvious since the image regions are changing during the evolution. We will defer discussion of the specific choice of basis and will now discuss computation of the basis weights for arbitrary bases.

In general, let the image basis be Φ_b for integer $1 \leq b \leq B$. Then the function f has a restriction to region R_i given by,

$$f_{R_i} = \sum_{b=1}^B c_{b,i} \Phi_b, \quad (106)$$

where $c_{b,i} \in \mathbb{R}$ is the weight for basis vector Φ_b and for region R_i . Thus, the portion of the cost functional that involves the function f_{R_i} is,

$$\begin{aligned} E(c_{1,i}, \dots, c_{B,i}) &= \int_{R_i} \left(\sum_{b=1}^B c_{b,i} \Phi_b - I \right)^2 \\ &\quad + \alpha \left\| \nabla \left(\sum_{b=1}^B c_{b,i} \Phi_b \right) \right\|^2 d\bar{x}. \end{aligned} \quad (107)$$

For each region, R_i , a necessary condition for the $c_{j,i}$'s to be a minimizer of E is that for all j ,

$$\begin{aligned} \frac{\partial E}{\partial c_{j,i}} &= \int_{R_i} 2 \left(\sum_{b=1}^B c_{b,i} \Phi_b - I \right) \Phi_j \\ &\quad + 2\alpha \left\langle \nabla \Phi_j, \sum_{b=1}^B c_{b,i} \nabla \Phi_b \right\rangle d\bar{x} = 0, \end{aligned} \quad (108)$$

which can be rearranged into the linear system,

$$\{\Gamma_i + \alpha\Lambda_i\} \mathbf{c}_i = \Xi_i, \quad (109)$$

where,

$$\Gamma_i = \begin{bmatrix} \int_{R_i} \Phi_1 \Phi_1 & \cdots & \int_{R_i} \Phi_B \Phi_1 \\ \int_{R_i} \Phi_1 \Phi_2 & \cdots & \int_{R_i} \Phi_B \Phi_2 \\ \vdots & & \vdots \\ \int_{R_i} \Phi_1 \Phi_B & \cdots & \int_{R_i} \Phi_B \Phi_B \end{bmatrix}, \quad (110)$$

and,

$$\Lambda_i = \begin{bmatrix} \int_{R_i} \langle \nabla \Phi_1, \nabla \Phi_1 \rangle & \cdots & \int_{R_i} \langle \nabla \Phi_B, \nabla \Phi_1 \rangle \\ \int_{R_i} \langle \nabla \Phi_1, \nabla \Phi_2 \rangle & \cdots & \int_{R_i} \langle \nabla \Phi_B, \nabla \Phi_2 \rangle \\ \vdots & & \vdots \\ \int_{R_i} \langle \nabla \Phi_1, \nabla \Phi_B \rangle & \cdots & \int_{R_i} \langle \nabla \Phi_B, \nabla \Phi_B \rangle \end{bmatrix}, \quad (111)$$

are each $B \times B$ matrices that must be computed for each region. Also,

$$\mathbf{c}_i = \begin{bmatrix} c_{1,i} \\ c_{2,i} \\ \vdots \\ c_{B,i} \end{bmatrix}, \text{ and } \Xi_i = \begin{bmatrix} \int_{R_i} \Phi_1 I \\ \vdots \\ \int_{R_i} \Phi_B I \end{bmatrix}, \quad (112)$$

are each $B \times 1$ vectors. Note that Ξ_i must be computed for each region. Note that all integrals in Eqs. (110)-(112) are with respect to $d\bar{x}$.

Thus, by solving the linear system in Eq. (109), we obtain, for each region, R_i , a vector of coefficients, \mathbf{c}_i , that yields the function f_{R_i} that minimizes Eq. (107) for a fixed region.

8.2.3 Choice of Basis

It is important to choose a basis that yields a desirable tradeoff between computational cost and segmentation performance. For this reason, we note that when during evolution via the Mumford-Shah evolution using the pixel-by-pixel basis, that the image regions resemble linear heat equation blurred version of the original image. For this reason, it makes sense to choose a basis that consists of various levels of linear heat equation (LHE) blurring. However, when the contour reaches the edges of the objects within an image, the regions,

f_{R_i} , resemble regions that are smoothed via the geometric heat equation, since the geometric heat equation (GHE) smooths more within the edges of an object than across the edges [52].

Thus, in this chapter, we will illustrate examples of this proposed method using two different bases. The first is a basis that consists of a various scales from the LHE-based scale space on the image. This is the well known Gaussian scale space. The second is a basis that consists of various scales from the GHE-based scale space on the image. Using too many scale levels in the basis has the effect of making the matrix to be inverted, $\Gamma_i + \alpha\Lambda_i$, ill-conditioned. It also increases computational cost of the algorithm.

8.3 *Implementation*

In this section, we discuss various implementations of Mumford-Shah segmentation that we will compare against the proposed method for computational performance. In each of these implementations, Mumford-Shah evolution is set up as a two-step iterative algorithm. The first step is one in which the function evolution is performed while holding the contour, and thus the regions R_i , fixed. This evolution is performed by some predetermined amount until some stopping criterion is met. Typically, it is not iterated until full convergence since this would be extremely time consuming. The second step is where the contour evolution is performed. The more accurate the function evolution that occurs during the first step, the faster the evolution of the contour will be during the second step [70]. It is in this step that the regions R_i change, thereby rendering the information obtained during the function evolution step inaccurate. This necessitates then going back to the first step again and repeating again until the contour ceases to move.

There are various choices of how to evolve for the functions, f_{R_i} during the first step of the algorithm. We will describe some of the methods below. Note that the first two, Jacobi Mumford-Shah and Multigrid Mumford-Shah both use the full pixel-by-pixel basis shown in Eq. (104).

8.3.1 Jacobi Mumford-Shah

We will denote solving for the functions f_{R_i} by way of Jacobi relaxation iterations as *Jacobi Mumford-Shah*. Jacobi iterations are a relaxation technique for solving discrete boundary value problems. More detail about the Jacobi relaxation methods can be found in [53].

The two most natural stopping criteria for Jacobi iterations are either to iterate some fixed amount of iterations, K , or to iterate until convergence, i.e., iterate until the error is below some predetermined threshold. As we will see, iterating until convergence is too computationally expensive to be practical. Thus, we shall report Jacobi Mumford-Shah for various fixed iteration amounts, denoted by K .

8.3.2 Multigrid Mumford-Shah

We will denote solving for the function f_{R_i} by way of Multigrid relaxation technique as *Multigrid Mumford-Shah*. Multigrid techniques solve for the solution of boundary value problems by computing the residual error on the original grid, coarsifying this residual error and solving for the coarse grid correction to the original function on the coarse grid where the computational cost is low, and then upsampling the correction to the original grid and correcting the function. Multigrid methods are known to be particularly efficient at solving elliptic or elliptic-like partial differential equations. For more detail about Multigrid relaxation methods, we refer the reader to [12, 53].

We specifically use a single Multigrid V-cycle for each function evolution, solving exactly for the coarsest scale correction. In addition, and use zeros steps of Jacobi pre-relaxation and one step of Jacobi post relaxation per grid level. According to the notation in [53] this corresponds to the parameters, $\nu_1 = 0$ and $\nu_2 = 1$.

8.3.3 Reduced Basis Implementation

We will denote representing the functions f_{R_i} by way of the reduced basis shown in Eq. (106) as *Proposed LHE* or *Proposed GHE* depending on whether the basis is obtained from the linear heat equation or the geometric heat equation.

The specific basis used for each evolution method always included the image itself, I

and the unit image 1 that has value 1 at each pixel. In addition, some levels of smoothing in between I and 1 are also included. We denote the number of basis images used in this method as $B \geq 2$ which will always imply $\Phi_1 = I$, $\Phi_B = 1$ and Φ_2 through Φ_{B-1} will be images within the scale space for increasing levels of smoothing.

8.3.4 Contour Evolution

In order to present a fair comparison, the same contour evolution was used for every method in this chapter. Therefore, the speed of the contour evolution was determined primarily by the function accuracy. Specifically, we use level set methods in order to allow for topological changes [51]. We compute the level set function only on a narrow band of pixels around the contour for speed. The contour evolves by,

$$\begin{aligned} \frac{\partial C}{\partial t} = & ((I - f_{R_a})^2 - (I - f_{R_b})^2) \vec{\mathcal{N}} \\ & + \alpha (\|\nabla f_{R_a}\|^2 - \|\nabla f_{R_b}\|^2) \vec{\mathcal{N}} \\ & + \beta \kappa \vec{\mathcal{N}}, \end{aligned} \tag{113}$$

which corresponds to evolving the level set function, Ψ , by

$$\Psi_t = -\frac{\partial C}{\partial t} \cdot \nabla \Psi \tag{114}$$

We used the fastest possible stable time step for contour evolution that ensures stability of the level set function when updating it with an explicit Forward Euler update scheme.

Note that level set methods naturally define a foreground and background region by the sign of the level set function. That is, the foreground region can be considered all regions in which the level set function, Ψ is positive and the background region is that in which Ψ is negative. Thus, this method naturally lends itself to a two-region method where the foreground function is defined by one coefficient vector, \mathbf{c}_1 and the background is defined by another coefficient vector, \mathbf{c}_2 . We will call this method the *two region* method. Unfortunately, disconnected foreground regions should not necessarily have the same coefficient vector, \mathbf{c}_1 . In the next section, we generalize this method by labeling disconnected foreground and background regions separately.

8.3.5 Multiregion Labeling

It is important to note that since the first two methods use the pixel-by-pixel basis, that it is not necessary to keep track of which other pixels are within the same region at a given time. This is because the pixel itself is its own basis function and therefore the pixel's intensity value is the value of the coefficient corresponding to that basis function. Therefore, it is possible to determine the value of a given pixel by only using information of the pixel itself and neighboring pixels that are within the same region. It is trivial to determine whether a neighboring pixel is in the same region when using signed distance level set functions for the representation of the contour. If the sign of the level set function is the same at the point of interest as it is at the neighboring point, then the two points are in the same region.

However, this is where the proposed method differs. Since each basis function involves all of the pixels within the image and the computation of Γ_i , Λ_i , and Ξ_i require knowledge of all of the pixels that are in the specific region and not just the neighboring pixels, it is necessary to label each pixel as belonging to a certain region. Unfortunately, two different non-neighboring pixels on the level set function having the same sign does not guarantee that the pixels are in the same region.

Thus, it is necessary to compute which pixels are in which regions. While this may seem prohibitive at first, this is handled by a very efficient two queue flood-fill labeling scheme that we will summarize in the Appendix. The output of this algorithm is a labeling that tells us which pixels are in which regions and allows us to compute Γ_i , Λ_i , and Ξ_i for each region. In this manner, we are able to label each pixel as belonging to a certain region, and thus, assign a different coefficient vector, \mathbf{c}_i to region i . We will call this method the *multiregion* method.

8.4 Experiments

In this section, we present experimental results for the proposed method as compared with various other segmentation techniques. In addition we stress the computational improvement of this technique over Mumford-Shah segmentations using Jacobi Mumford-Shah and Multigrid Mumford-Shah.

8.4.1 LHE vs. GHE Comparison

Figure 41 shows an example segmentation of a peregrine falcon image. While this image is nearly trivial to segment, we include it for the purpose of reporting computational performance, and to display a specific property of the Proposed LHE method as compared with the Proposed GHE method.

The top row in this image shows the evolution of Multigrid Mumford-Shah segmentation. The leftmost image is the original image and the initialization of the contour. The three images to the right show the evolution of the algorithm to convergence. The middle row shows the segmentation result of the Proposed LHE method using a basis of dimension $B = 3$, where $\Phi_1 = I$, $\Phi_2 = l(I, 10.0)$, and $\Phi_3 = 1$. Note that $l(I, \tau)$ denotes τ time units of Gaussian blurring on image I . Note the similar looking function regions when compared with Multigrid Mumford-Shah, especially around the boundary of the peregrine falcon. The exact evolution of the contour is slightly different but produces a nearly identical resulting segmentation. The bottom row illustrates the segmentation result of the Proposed GHE method using a basis of dimension $B = 3$, where $\Phi_1 = I$, $\Phi_2 = g(I, 10.0)$, and $\Phi_3 = 1$. Again, $g(I, \tau)$ denotes τ time units of geometric heat equation blurring on image I . Note that there is more accurate edge information on the wing of the peregrine falcon in the second image from the left.

8.4.2 Two Region vs. Multi-Region Results

Figure 42 compares Multigrid Mumford-Shah with the Proposed GHE method assuming two regions and the Proposed GHE method allowing for multiple regions. The top row shows the evolution of Multigrid Mumford-Shah on an image containing various types of fruit having different intensity levels. Since the Multigrid Method uses the pixel-by-pixel basis, each piece of fruit assumes its own intensity. The middle row shows the evolution of the two region Proposed GHE method. The two region Proposed GHE method has a limitation in that it only uses two sets of coefficients to represent the image, i.e., one for the foreground and one for the background. Therefore, this method is not able to represent the different fruit intensities. The bottom row shows the evolution of the multiregion Proposed GHE

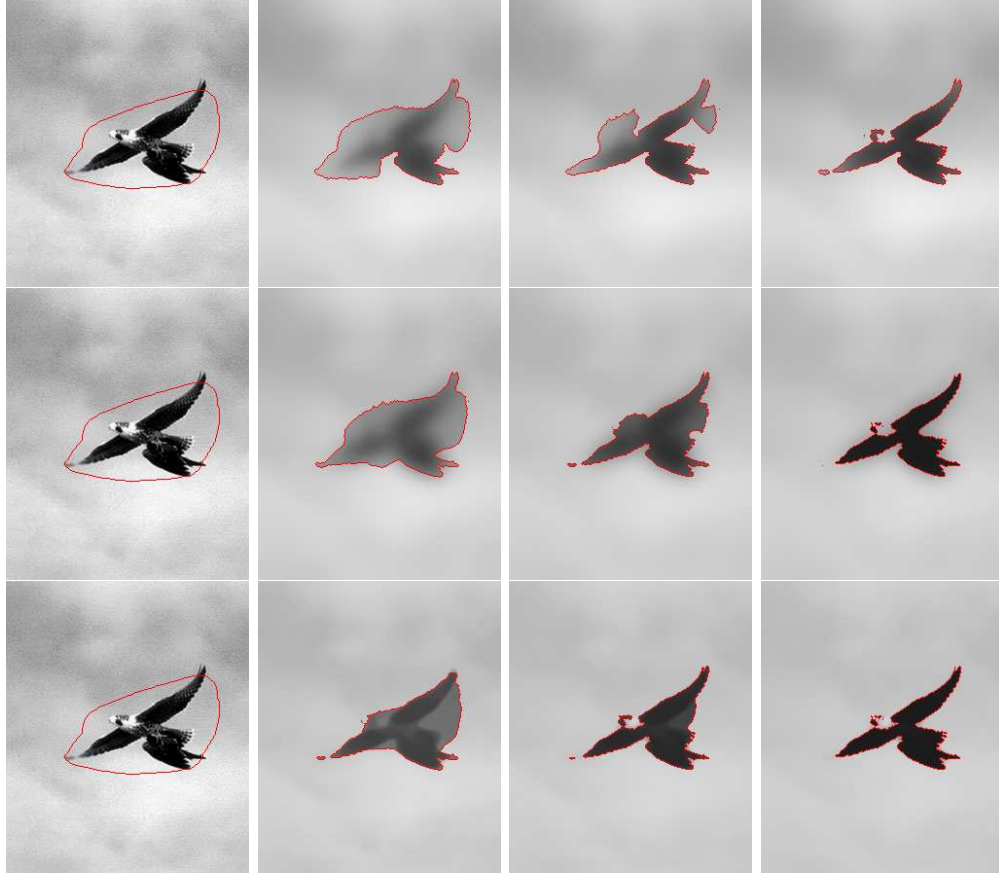


Figure 41: Peregrine falcon segmentation: For each row, progression from left to right shows segmentation evolution from initial contour to final segmentation. (Top row) Multigrid Mumford-Shah segmentation with pixel-by-pixel basis. (Middle row) Proposed LHE has very similar function regions when using drastically reduced basis. (Bottom row) Proposed GHE produces similar segmentation but function regions preserve image edges better than in top two rows.

method. Note that in the third image from the left, when the strawberry on the bottom right of the image becomes disconnected from the remaining foreground region, it becomes darker since this region then obtains its own coefficient vector, \mathbf{c}_i . The final segmentation image on the right has seven distinct regions, each with its own coefficient vector. The result matches very closely with the pixel-by-pixel basis Mumford-Shah segmentation technique but at a fraction of the computational cost.

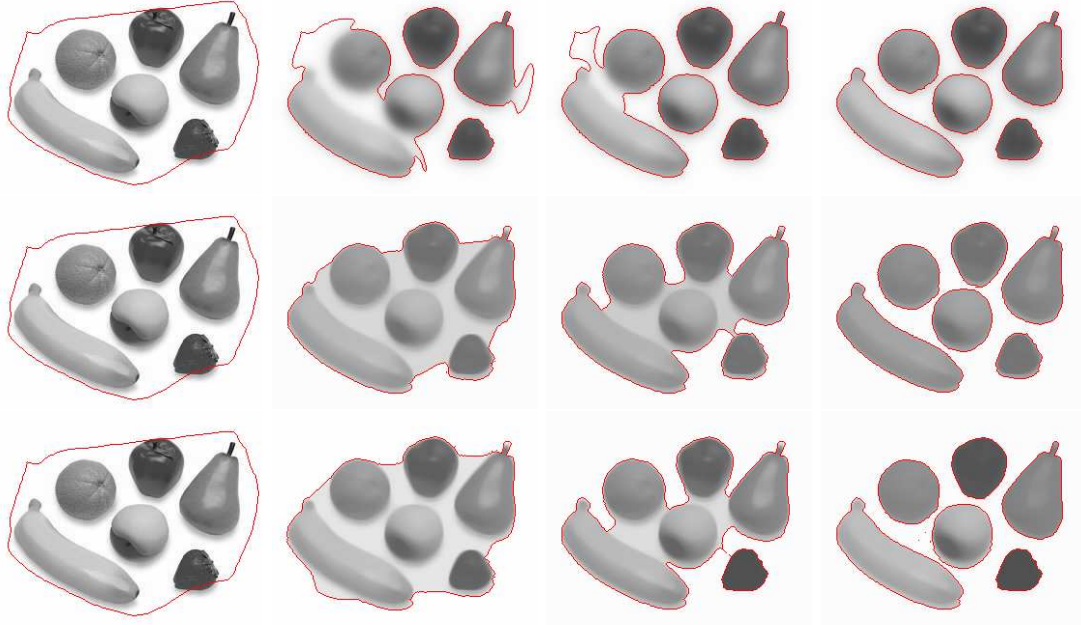


Figure 42: Fruits image: (Top row) Segmented using full pixel-by-pixel basis Mumford-Shah segmentation. (Middle row) Segmented using Proposed GHE method using two distinct regions, foreground and background. Note the lack of accuracy in mean intensity for each function region. (Bottom row) Segmented using Proposed GHE method using multiple regions. Note the ability to represent each different fruit as a separate region with distinct basis weight coefficients.

8.4.3 Segmentation in Noise

Figure 43 shows an example of a segmentation of the fruit image in noise. The top row shows Multigrid Mumford-Shah segmentation, the left of which is the initialization and the right of which is the final segmentation. The middle row shows the initial contour and final segmentation using the two-region proposed GHE method. The bottom row shows the initial contour and final segmentation using the multiregion proposed GHE method.

8.4.4 Computational Improvement

Table 3 shows the computational speed improvement of the proposed algorithms. In this table, “Proposed 2R LHE” represents the two region reduced basis technique proposed in this chapter where there are only two different coefficient vectors, one for the foreground and one for the background. As stated before, LHE represents linear heat equation basis.

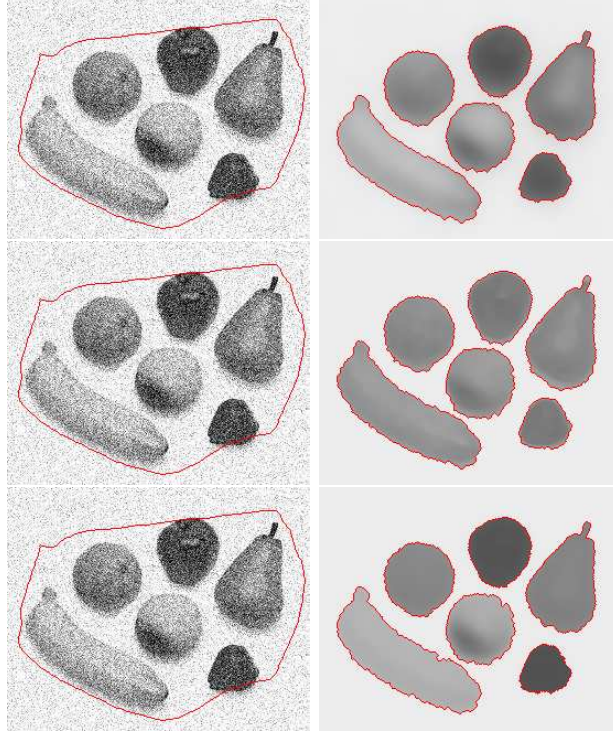


Figure 43: Noisy fruit image: (Top row) Initial contour overlayed on noisy image along with final Multigrid Mumford-Shah segmentation of image. (Middle row) Initial contour and final segmentation using two-region proposed GHE method. (Bottom row) Initial contour and final segmentation using multiregion proposed GHE method.

“Proposed MR LHE” represents the multiregion version of the proposed method. The parameter K/B shows the number of Jacobi iterations, K for the Jacobi Mumford-Shah method or then number of basis vectors, B for the proposed techniques.

The peregrine image is of size 250×300 pixels and the fruits image is of size 275×221 pixels. The main reason for the much increased computational cost of the fruits image is that initial contour was chosen further from the final segmentation in the fruits image than in the peregrine image.

Note that the proposed method is substantially faster than Multigrid Mumford-Shah. Among the variants of the proposed segmentation method, the multiregion (MR) methods are the slowest because they are required to label each pixel as being in a specific region every iteration. While this algorithm is not slow, it brings with it some computational cost. The Jacobi Mumford-Shah method is the slowest of all.

Table 3: Computational performance of proposed method compared with Multigrid Mumford Shah (MG Mum-Shah), Jacobi Mumford Shah with various number of function evolution iterations, K , per contour evolution, and proposed technique.

image	method	K/B	iterations	time
peregrine	MG Mum-Shah		450	59s
peregrine	Jacobi Mum-Shah	1	3400	243s
peregrine	Jacobi Mum-Shah	2	2000	232s
peregrine	Jacobi Mum-Shah	5	1200	300s
peregrine	Proposed 2R LHE	3	200	6s
peregrine	Proposed 2R GHE	3	250	5s
peregrine	Proposed MR LHE	3	250	9s
peregrine	Proposed MR GHE	3	250	8s
fruits	MG Mum-Shah		3350	385s
fruits	Jacobi Mum-Shah	1	10800	670s
fruits	Jacobi Mum-Shah	2	6400	641s
fruits	Jacobi Mum-Shah	5	3800	810s
fruits	Proposed 2R LHE	3	600	15s
fruits	Proposed 2R GHE	3	400	9s
fruits	Proposed MR LHE	3	1100	47s
fruits	Proposed MR GHE	3	550	23s

8.4.5 Brain CT Segmentation

Figure 44 shows a multiregion segmentation of a 145×145 pixel brain CT image using the Proposed GHE method with a three-image basis. Note that the method is able to segment a majority of the important features in this image including the white objects in the center of the brain as well as the ears.

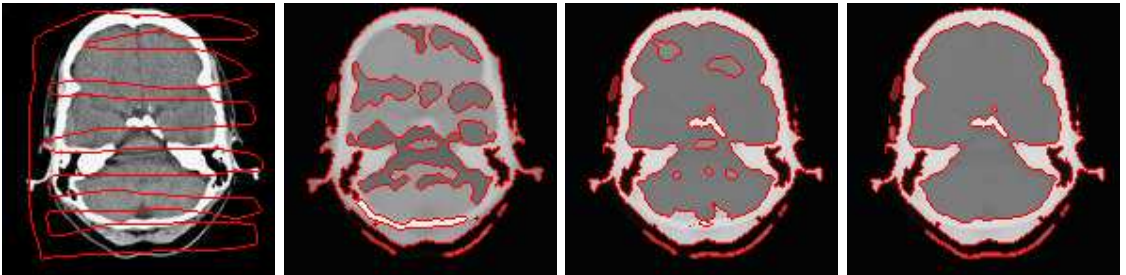


Figure 44: Segmentation of brain CT image using Multiregion Proposed GHE method.

8.5 *Conclusion*

In this chapter, we have presented a reduced basis method for fast robust approximation to Mumford-Shah segmentation. We explain how the specific choice of basis allows for accurate approximation to the full, standard “pixel-by-pixel” basis Mumford-Shah evolution, while reducing computational cost significantly. We presented a method for labelling like regions, thus enabling the use of regions with independent coefficient vectors. We showed the performance of this method for real images that were both noiseless and noisy. Future work includes determining optimal bases for Mumford-Shah segmentation.

CHAPTER IX

CONCLUSION

The central application in this thesis has been in reconstruction of tomography images using a cost functional inspired by the Mumford-Shah functional. This tomography reconstruction technique relies on implementing a coupled gradient flow, i.e., the simultaneous evolution of a contour and an image function. In this implementation, the contour evolution relies on the value of the function and the function evolution relies on the accuracy of the contour. The most accurate contour evolutions are typically obtained when the function is solved as accurately as possible.

However, unlike in traditional Mumford-Shah segmentation, the image function is not known *a priori*. In tomography, the image function must be obtained from the projections. For this reason, the function evolution is exceedingly slow when compared with Mumford-Shah segmentation. In this thesis, we have presented a novel multiscale computation technique, inspired by multigrid methods, for the estimation of these image functions. We have shown that the multiscale computation technique lowers the overall computational cost of the algorithm by quickly obtaining accurate functions values, thereby allowing an accurate contour evolution.

In addition, we have proposed a novel method to obtain the initial contour for the proposed tomography reconstruction technique. This method produces a fuzzy binary segmentation of the image density function and thereby yields a class of initial surfaces. These initial contours are shown to reduce the number of iterations that it takes for the surface to evolve. In addition, these contours are more robust and produce less local minima than contours produced by thresholding minimum norm segmentation obtained by algebraic reconstruction technique (ART) or of those obtained by thresholding filtered back projection. In addition, the choice of threshold is no longer arbitrary.

For validation of the proposed technique, we compared the proposed technique with

various other reconstruction algorithms. In addition we also illustrated the ability of the proposed technique to accurately reconstruct phantom images of heart perfusion abnormalities using the modified NCAT phantom.

The remaining chapters, while all being novel contributions to computer vision independently, support the claims and implementation made in the tomography chapters. The chapter involving rigid contour registration using minimal surfaces contains a novel scale space for contour registration that, while not computationally feasible, highlights the ability of multiscale techniques to avoid registration local minima. This chapter also expounds some of the details of level set surface evolution techniques. In supplement, we have shown a sufficient condition for existence of these minimal surfaces using the maximum principle of minimal surfaces.

We have presented results on multigrid computation for optical flow of non-linear partial differential equations (PDE's). This chapter highlights the computational power of using multigrid techniques for solving PDE's whose solution functions are spatially smooth. In addition, we have shown that a previously proposed cost functional for optical flow is not rotationally invariant, and we have proposed a slight modification to make this optical flow rotationally invariant, while still preserving the edges of objects in the optical flow.

Finally, we have presented a multiscale method for computational of Mumford-Shah segmentation on images where the image function basis is reduced. This method has extremely fast convergence and produces very similar segmentations to the Mumford-Shah functional where each pixel is allowed to vary independently. We have presented a labeling scheme and method where an arbitrary number of regions can be modeled by this reduced basis technique.

APPENDIX A

EVOLUTION EQUATION DERIVATION

A.1 Evolution derivation

In order to determine the evolution equations, i.e., Eq. (5), (6) and (7), of the image function and the contour connecting the two image regions, we use calculus of variations.

Taking the derivative of the functional J_1 with respect to time we obtain,

$$\frac{\partial J_1}{\partial t} = -2 \sum_{\theta \in \Theta} \int_0^{L_\theta} (p(s, \theta) - \hat{p}(s, \theta)) \frac{\partial \hat{p}}{\partial t} ds.$$

Note, that while p is not a function of time, \hat{p} , which depends on the image function and the contour, is a function of time. Therefore, where $\vec{\nu}_{R_i}$ is the unit outward normal vector with respect to region R_i , we see that,

$$\frac{\partial \hat{p}}{\partial t} = \frac{\partial}{\partial t} \int_{\Omega} \hat{f}(x, y) \delta(x \cos \theta + y \sin \theta - s) d\bar{x}, \quad (115)$$

$$= \frac{\partial}{\partial t} \int_{R_1} \hat{f}_{R_1}(x, y) \delta(P_\theta(x, y) - s) d\bar{x} \quad (116)$$

$$+ \frac{\partial}{\partial t} \int_{R_2} \hat{f}_{R_2}(x, y) \delta(P_\theta(x, y) - s) d\bar{x}$$

$$= \int_0^L \left\langle \vec{C}_t, \hat{f}_{R_1}(x, y) \delta(P_\theta(x, y) - s) \vec{\nu}_{R_1} \right\rangle d\ell \quad (117)$$

$$+ \int_{R_1} \hat{f}_{R_1,t} \delta(P_\theta(x, y) - s) d\bar{x}$$

$$+ \int_0^L \left\langle \vec{C}_t, \hat{f}_{R_2}(x, y) \delta(P_\theta(x, y) - s) \vec{\nu}_{R_2} \right\rangle d\ell$$

$$+ \int_{R_2} \hat{f}_{R_2,t} \delta(P_\theta(x, y) - s) d\bar{x}$$

$$= \int_0^L \left\langle \vec{C}_t, \left(\hat{f}_{R_1}(x, y) - \hat{f}_{R_2}(x, y) \right) \delta(P_\theta(x, y) - s) \vec{\nu}_{R_1} \right\rangle d\ell \quad (118)$$

$$+ \int_{R_1} \hat{f}_{R_1,t} \delta(P_\theta(x, y) - s) d\bar{x}$$

$$+ \int_{R_2} \hat{f}_{R_2,t} \delta(P_\theta(x, y) - s) d\bar{x}.$$

Note, that Eq. (117) follows from Eq. (116) is a result of the divergence theorem.

Substituting Eq. (118) into Eq. (A.1), and abbreviating $\delta(P_\theta(x, y) - s)$ as δ , we obtain,

$$\frac{\partial J_1}{\partial t} = -2 \sum_{\theta \in \Theta} \int_0^{L_\theta} (p - \hat{p})(s, \theta) \int_0^L \left\langle \vec{C}_t, (\hat{f}_{R_1} - \hat{f}_{R_2}) \delta \vec{\nu}_{R_1} \right\rangle d\ell ds \quad (119)$$

$$\begin{aligned} & -2 \sum_{\theta \in \Theta} \int_0^{L_\theta} (p - \hat{p})(s, \theta) \int_{R_1} \hat{f}_{R_1, t} \delta d\bar{x} ds \\ & -2 \sum_{\theta \in \Theta} \int_0^{L_\theta} (p - \hat{p})(s, \theta) \int_{R_2} \hat{f}_{R_2, t} \delta d\bar{x} ds \\ = & -2 \int_0^L \sum_{\theta \in \Theta} (p - \hat{p})(P_\theta(x, y), \theta) \left\langle \vec{C}_t, (\hat{f}_{R_1} - \hat{f}_{R_2}) \vec{\nu}_{R_1} \right\rangle d\ell \end{aligned} \quad (120)$$

$$\begin{aligned} & -2 \int_{R_1} \sum_{\theta \in \Theta} (p - \hat{p})(P_\theta(x, y), \theta) \hat{f}_{R_1, t} d\bar{x} \\ & -2 \int_{R_2} \sum_{\theta \in \Theta} (p - \hat{p})(P_\theta(x, y), \theta) \hat{f}_{R_2, t} d\bar{x} \\ = & \int_0^L \left\langle \vec{C}_t, 2 \sum_{\theta \in \Theta} (\hat{p} - p)(P_\theta(x, y), \theta) (\hat{f}_{R_1} - \hat{f}_{R_2}) \vec{\nu}_{R_1} \right\rangle d\ell \\ & + \int_{R_1} \hat{f}_{R_1, t} 2 \sum_{\theta \in \Theta} (\hat{p} - p)(P_\theta(x, y), \theta) d\bar{x} \\ & + \int_{R_2} \hat{f}_{R_2, t} 2 \sum_{\theta \in \Theta} (\hat{p} - p)(P_\theta(x, y), \theta) d\bar{x}. \end{aligned} \quad (121)$$

Taking the derivative of the functional J_2 with respect to time similarly yields,

$$\frac{\partial J_2}{\partial t} = \int_{R_1} \frac{\partial}{\partial t} \|\nabla \hat{f}_{R_1}(x, y)\|^2 d\bar{x} + \int_{R_2} \frac{\partial}{\partial t} \|\nabla \hat{f}_{R_2}(x, y)\|^2 d\bar{x} \quad (122)$$

$$= 2 \int_{R_1} \hat{f}_{R_1, x} \hat{f}_{R_1, xt} + \hat{f}_{R_1, y} \hat{f}_{R_1, yt} d\bar{x} \quad (123)$$

$$\begin{aligned} & + 2 \int_{R_2} \hat{f}_{R_2, x} \hat{f}_{R_2, xt} + \hat{f}_{R_2, y} \hat{f}_{R_2, yt} d\bar{x} \\ & + 2 \int_0^L \left\langle \vec{C}_t, (\|\nabla \hat{f}_{R_1}(x, y)\|^2 - \|\nabla \hat{f}_{R_2}(x, y)\|^2) \vec{\nu}_{R_1} \right\rangle d\ell \\ = & 2 \int_{R_1} \hat{f}_{R_1, t} \Delta \hat{f}_{R_1} d\bar{x} \\ & + 2 \int_{R_2} \hat{f}_{R_2, t} \Delta \hat{f}_{R_2} d\bar{x} \\ & + 2 \int_0^L \left\langle \vec{C}_t, (\|\nabla \hat{f}_{R_1}(x, y)\|^2 - \|\nabla \hat{f}_{R_2}(x, y)\|^2) \vec{\nu}_{R_1} \right\rangle d\ell. \end{aligned} \quad (124)$$

Finally, taking the derivate with respect to time of the functional J_3 , which corresponds to the arc length of the contour is,

$$\frac{\partial J_3}{\partial t} = \frac{\partial}{\partial t} \int_0^L d\ell, \quad (125)$$

however the partial derivative operator can not be interchanged with the integral because the contour length, L , is a function of the contour that is changing in time. Thus, the contour must be written in terms of a specific parameterization. Therefore, writing the contour as a function $\vec{C}' : [0, 1] \rightarrow \mathbb{R}^2$, it follows that,

$$\frac{\partial J_3}{\partial t} = \frac{\partial}{\partial t} \int_0^L d\ell = \frac{\partial}{\partial t} \int_0^1 \|\vec{C}'_p(p, t)\| dp \quad (126)$$

$$= \int_0^1 \frac{\partial}{\partial t} \sqrt{\langle \vec{C}'_p, \vec{C}'_p \rangle} dp \quad (127)$$

$$= \int_0^1 \frac{\langle \vec{C}'_{pt}, \vec{C}'_p \rangle}{\sqrt{\langle \vec{C}'_p, \vec{C}'_p \rangle}} dp = \int_0^1 \left\langle \vec{C}'_{pt}, \frac{\vec{C}'_p}{\|\vec{C}'_p\|} \right\rangle dp \quad (128)$$

$$= \int_0^1 \langle \vec{C}'_{pt}, \vec{T} \rangle dp \quad (129)$$

$$= \int_0^1 \langle \vec{C}'_{tp}, \vec{T} \rangle dp \quad (130)$$

$$= - \int_0^1 \langle \vec{C}'_t, \vec{T}_p \rangle dp + \langle \vec{C}'_t, \vec{T} \rangle \Big|_0^1 \quad (131)$$

$$= - \int_0^1 \langle \vec{C}'_t, \vec{T}_p \rangle dp \quad (132)$$

$$= - \int_0^1 \left\langle \vec{C}'_t, \vec{T}_\ell \frac{d\ell}{dp} \right\rangle dp \quad (133)$$

$$= - \int_0^1 \langle \vec{C}'_t, \vec{T}_\ell \rangle \|\vec{C}'_p\| dp \quad (134)$$

$$= - \int_0^L \langle \vec{C}_t, \vec{T}_\ell \rangle d\ell, \quad (135)$$

where \vec{T} is the unit tangent vector at a point on the contour, Eq. 130 follows from Eq. 129 when the second order partial derivatives of the contour are continuous, Eq. 131 follows from Eq. 130 by integration by parts, and Eq. 132 follows from Eq. 131 since values $p = 0$ and $p = 1$ correspond to the same points on the contour. Also note that equation Eq. 133 follows from Eq. 132 by the chain rule. By definition, the derivative of the unit tangent vector with respect to arclength is curvature, κ , times the unit normal, and thus,

$$\frac{\partial J_3}{\partial t} = \int \langle \vec{C}_t, \kappa \nu_{R_1} \rangle d\ell. \quad (136)$$

In the three dimensional case, i.e., when taking the derivative with respect to surface area, the term κ is replaced by the local mean curvature, H .

APPENDIX B

SUFFICIENCY PROOF OF CONVERGENT TIME STEP

B.1 Existence of Convergent Time Step

The discrete implementation of the forward Euler iterations on the estimated density functions as in Chapter 2 can be seen as matrix multiplication,

$$f^{n+1} = (I + \Delta t(-\alpha P + \beta S)) f^n + g, \quad (137)$$

where f^n is a vector representing the image function at iteration n , I is the identity matrix, Δt is the time step, and α and β are the weighting factors for the projection and smoothness terms respectively as in Chapter 2. Also, note that P is the projection matrix that operates on an image, producing, for each pixel, an image that contains the sum of all projections at the points on the projections to which that pixel projects. The matrix S represents the discrete Laplacian operator on the image function with Neumann boundaries conditions enforcing that the normal derivative on the image boundary is zero. The vector g is a constant vector that does not affect whether the iterative system converges.

The null space of S is the set of vectors corresponding to constant images. This follows naturally since the only solutions of the Laplace equation with zero normal derivative on the boundary are constant images.

According to the above description, P is a mapping that, for each pixel on the image, sums all of the image pixels that project to the same projection locations as the pixel itself. Formally, we can see that P is of the form,

$$P = U^T U, \quad (138)$$

where U is the matrix that corresponds to projecting the image pixels onto the projection pixels, and U^T corresponds to mapping the projection pixels back onto the image pixels that projected to it. The matrix U depends on the number and angles of the projections and the selected interpolation scheme used in projection.

Note that U is not a square matrix. It is a matrix with number of columns corresponding to the number of image pixels and number of rows corresponding to the number of projection pixels. Also note that U only contains real values. Also, we will assume that the sum of all the elements of P is strictly positive. This is true for any set of projection angles and interpolation scheme due to the additive nature of tomography as modeled by the Radon transform.

Lemma 1 *The matrix, $P = U^T U$, corresponding to the projection matrix described above is positive semidefinite.*

For $x \neq 0$,

$$x^T P x = x^T U^T U x = (Ux)^T (Ux) = \|Ux\|^2 \geq 0, \quad (139)$$

where $(Ux)^T (Ux) = \|Ux\|^2$ is true since Ux is real. \square

Lemma 2 *The matrix, S , corresponding to the discrete Laplacian operator with Neumann boundary conditions where the normal derivative is zero along the boundary, is negative semidefinite.*

The structure of the matrix S is that it is a real matrix whose diagonal elements, s_{jj} , are negative and the off-diagonal elements are all positive values who sum for each row, j , is such that,

$$s_{jj} = - \sum_{k \neq j} s_{jk}. \quad (140)$$

Then, this lemma follows naturally from Gerschgorin's theorem [41], from which we have that if λ is an eigenvalue of S , then for some integer, $1 \leq j \leq n$, that,

$$|a_{jj} - \lambda| \leq |a_{j1}| + \cdots + |a_{j,j-1}| + |a_{j,j+1}| + \cdots + |a_{jn}|. \quad (141)$$

But, from Eq. (140), and since $s_{jk} > 0$ for $j \neq k$ and $s_{jj} < 0$ for all $1 \leq j \leq n$, we have that,

$$|s_{jj} - \lambda| \leq |s_{j1}| + \cdots + |s_{j,j-1}| + |s_{j,j+1}| + \cdots + |s_{jn}| = \sum_{k \neq j} s_{jk} = -s_{jj} = |s_{jj}|. \quad (142)$$

Thus, all eigenvalues must be, for some $1 \leq j \leq n$, within $|s_{jj}|$ of s_{jj} , a strictly negative number. In addition, since S is a real, symmetric matrix, all eigenvalues are real. Therefore, all of the eigenvalues of S are non-positive real numbers, thus proving the lemma. \square

This is shown in Fig. 45 which illustrates the eigenvalues of S lying within the Gerschgorin disks centered at the negative real values, s_{jj} .

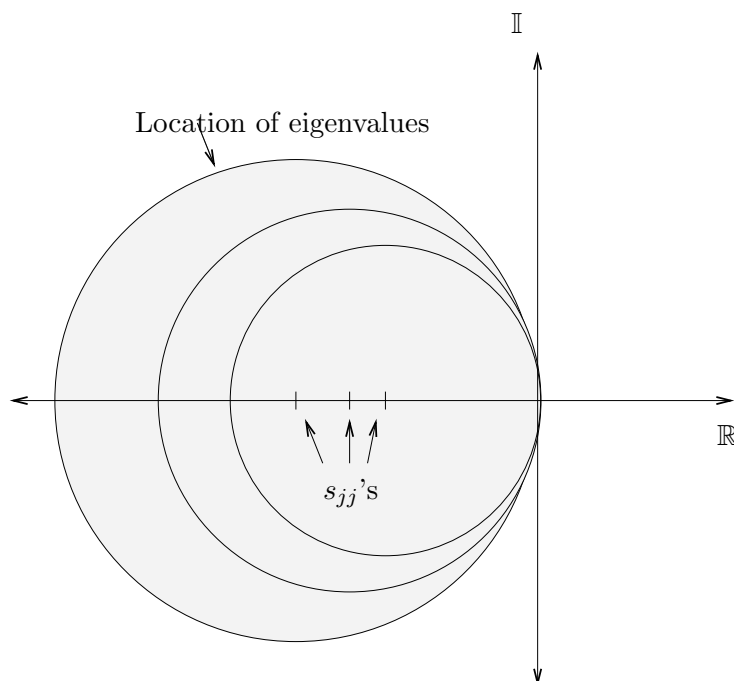


Figure 45: Gerschgorin disks showing location of eigenvalues of Laplacian operator matrix, S .

From these lemmas, it is trivially true that $-\alpha P + \beta S$ is negative semidefinite since α and β are positive constants. However, it is also true that the matrix $-\alpha P + \beta S$ is, in fact, negative definite. This occurs due to the structure of the null space of matrix S . This is proven below.

Lemma 3 *The matrix, $-\alpha P + \beta S$, is negative definite.*

The null space of S is the space of constant image functions. Thus, $x = a \neq 0$ is in the

null space of S . Then,

$$x^T(-\alpha P + \beta S)x = -\alpha x^T P x = -a^2 \sum_{i,j} P_{ij} < 0, \quad (143)$$

since a is real, and the sum of all the components of P is strictly positive.

If $x \neq 0$ is not a constant vector, and thus not in the null space of S , then

$$x^T(-\alpha P + \beta S)x \leq x^T S x < 0, \quad (144)$$

where the first inequality is true since P is positive semidefinite and the second is clear by spectral decomposition of x with respect to the eigenvectors of S . \square

Theorem 1 *There exists a $\Delta t > 0$ such that $(I + \Delta t(-\alpha P + \beta S))$ has all of its eigenvalues in the interval $(-1, 1)$.*

Since $(-\alpha P + \beta S)$ is negative definite, all of its eigenvalues are strictly less than zero. Choose $\Delta t < \frac{2}{|\lambda_*|}$ where λ_* is the most negative eigenvalue of $(-\alpha P + \beta S)$. Therefore, $\Delta t(-\alpha P + \beta S)$ has all of its eigenvalues in the interval $(-2, 0)$. The theorem follows from the spectral shift theorem [41]. \square

From this theorem, it follows that repeated multiplication by the matrix $(I + \Delta t(-\alpha P + \beta S))$ with the properly chosen Δt converges.

B.2 Sufficient Time Step for Convergence

In order to sketch a proof for a time step sufficient for convergence, we need to look at the structure of the matrices S and P . We will show a sufficient time step for arbitrary dimension of the image density function, D . The matrix S has a very specific structure for most of its rows. Most rows have the value $-2D$ along its diagonal and have $2D$ 1's in the off diagonals, and the rest of the terms in each row are 0. The only rows that do not have this structure are rows that correspond to image pixels that are on the boundary. These rows, that correspond to boundaries of the image, have negative integer values, q , in the diagonal with magnitude less than $2D$ and q 1's along the off diagonal. The rest of the terms are 0.

The structure of P is such that the value along each diagonal has value equal to the number of projections, p . The remaining entries in each row are positive and sum to a number less than or equal to $(L-1)p$ where L is the length of the longest dimension of the image.

Therefore, by application of the Gerschgorin theorem, the matrix $-\alpha P + \beta S$ will have its most negative eigenvalue λ_* such that,

$$|\lambda_* - (-\alpha p - \beta 2D)| \leq \alpha(L-1)p + \beta 2D. \quad (145)$$

Therefore, it follows that,

$$-\alpha(L-1)p - \beta 2D \leq \lambda_* - (-\alpha p - \beta 2D) \leq \alpha(L-1)p + \beta 2D \quad (146)$$

$$-\alpha Lp - \beta 4D \leq \lambda_* \leq \alpha(L-2)p. \quad (147)$$

But since we proved above that $-\alpha P + \beta S$ is negative definite, we know that,

$$-\alpha(L)p - \beta 4D \leq \lambda_* < 0. \quad (148)$$

Therefore, as shown in the proof of Theorem 1, a sufficient time step for convergence of the iterative equation is,

$$\Delta t < \frac{2}{\alpha Lp + \beta 4D}. \quad (149)$$

APPENDIX C

EXISTENCE CRITERIA FOR MINIMAL SURFACES

We will explain a sufficient condition for existence of the minimal surface that connects two contours. The first section will describe the condition for existence of the catenoid. The second section will explain the condition for existence of the minimal surface connecting two contours using the maximum principle of minimal surfaces.

C.1 Existence of Catenoid

C.1.1 Parameterization

The catenoid is the only surface of revolution with the topology of a cylinder that is a minimal surface [33]. The parametric equation of the catenoid is,

$$(x(u, v), y(u, v), z(u, v)) = (p \cosh(v/p) \cos(u), p \cosh(v/p) \sin(u), v) , \quad (150)$$

where $u \in [0, 2\pi)$ and $v \in [-z_H/2, z_H/2]$ are the surfaces parameters and p is the radius of the middle contour. Note that in the appendix we use the convention that the top contour of the catenoid is at $z = z_H/2$ and the bottom contour of the catenoid is at $z = -z_H/2$. Although this does not follow the convention of the paper, it will prove useful in the analysis. It can be shown that this surface has constant zero mean curvature, although we will not show it here.

By setting $u = 0$ and $v = z_H/2$, which represents the top of the catenoid, we get that, $(r, 0, 0) = (p \cosh(z/2p), 0, 0)$, where r is the radius of the circular contour and the top and bottom of the catenoid. This shows that $r = p \cosh(z/2p)$, giving a relation between the radius of the top and bottom circular contours, r , the height of the catenoid, z , and the inner radius of the catenoid which occurs at $z = 0$, p .

Solving this function for positive z results in, $z = 2p \ln \left(\frac{r + \sqrt{r^2 - p^2}}{p} \right)$.

C.1.2 When does it collapse?

By finding out, for what value of z the current catenoid with top radius r fails to exist, we are essentially answering our title question. The exact z where the catenoids fail to exist are when $\frac{\partial r(p,z)}{\partial p} = 0$ for a given r , for this is where the bottom of the constant z curves intersect with the horizontal fixed r lines. This analysis proceeds as follows,

$$r(p, z) = p \cosh\left(\frac{z}{2p}\right) \text{ and } \frac{\partial r}{\partial p} = \cosh\left(\frac{z}{2p}\right) - \frac{z}{2p} \sinh\left(\frac{z}{2p}\right). \quad (151)$$

Setting this partial derivative equal to zero will give us the appropriate value of p where z is maximum and the catenoid still exists. I will call these critical values p' and z' . Substituting for z and $\frac{r}{p'}$ for $\cosh(z'/2p')$ yields,

$$0 = r - \ln\left(\frac{r + \sqrt{r^2 - p'^2}}{p'}\right) \sqrt{r^2 - p'^2}. \quad (152)$$

By finding, for a given r , the p' that satisfies Eq. (152), we will have found the critical p' below which catenoids fail to exist. That p' will produce the largest possible catenoid height, z' .

An explicit function to get p' from r is, $p'(r) = \alpha r$, where α is the zero of the function,

$$f(x) = 1 - \ln\left(\frac{1 + \sqrt{1 - x^2}}{x}\right) \sqrt{1 - x^2}. \quad (153)$$

It can be seen that $p'(r)$ satisfies Eq. (152). by substitution into Equation (152) if α is a zero of $f(x)$. The zero of $f(x)$ is found numerically to be, $\alpha \approx 0.5524$.

It follows that, $z'(p'(r), r) = 2\alpha \ln\left(\frac{1 + \sqrt{1 - \alpha^2}}{\alpha}\right) r = \beta r$ where,

$$\beta = 2\alpha \ln\left(\frac{1 + \sqrt{1 - \alpha^2}}{\alpha}\right) \approx 1.325. \quad (154)$$

These results show that the critical p' and z' for a given r are given by, $p' = \alpha r$ and $z' = \beta r$. That is, the critical inner radius, p' , and critical height, z' , of a catenoid of fixed top radius are each directly proportional to the fixed top radius. Therefore, a catenoid with radius r will exist for heights less than or equal to $z = 1.325r$

C.2 Existence of Minimal Surface for Arbitrary Closed Contours

Arbitrary contours that are not concentric circles yield surfaces that are not catenoids and typically have no closed form parameterization. Consider an arbitrary closed contour, C , and its identical, aligned replica, C' , lying directly below in, separated in respective contour planes by a distance z_M . In each plane, the contour lies between its inscribed circle and its exscribed circle with radii r_{in} and r_{ex} respectively. If we choose the separation distance so that $z_M < 1.325r_{\text{in}}$, a catenoid exists that connects the inscribed circles, as proven in Section C.1. Since $r_{\text{in}} \leq r_{\text{ex}}$, and thus, $z_M < 1.325r_{\text{in}} \leq 1.325r_{\text{ex}}$. It follows that the catenoid of radius r_{ex} exists for this chosen z_M . It then follows from the maximum principle of minimal surfaces, that the minimal surface with contours C and C' as its boundary will exist and will completely contain the catenoid connecting the inscribed circles, that is, the minimal surface will exist for the chosen separation distance $z_M < 1.325r_{\text{in}}$.

This argument can be extended to show existence for the minimal surface connecting the top contour, D , and bottom contour E . Consider the inscribed circle lying within the intersection of the interior of the two contours. A catenoid with this radius, r_b , will be completely contained in the minimal surface connecting D and E and therefore by a similar argument as above, the minimal surface connecting D and E will exist when the separation distance is less than $1.325r_b$.

APPENDIX D

FLOOD FILL LABELING ALGORITHM

Here we explain the algorithm for two queue flood-fill labeling. This algorithm gives each pixel in the region a label to identify it only with points that are within the same region. Even though this algorithm is run every iteration, it is a manageable part of the computational cost of the proposed technique. The proposed multigrid technique still outperforms other comparable methods with respect to computational cost.

The algorithm operates on the discretized image domain and assumes that the region also contains a level set function, Ψ , that exists on each pixel of the domain. The algorithm takes in a signed level set function, Ψ , and after completion, has labeled each pixel in the grid with a unique label identifying which connected region it is in. Figure 46 shows an example of a labeling given a 6×6 pixel grid.

The two queue flood fill labelling algorithm is as follows:

1. $L \leftarrow 1$
2. Enqueue upper leftmost pixel into queue #1
3. While queue #1 is not empty:
 - (a) Dequeue pixel p from queue #1 and label p to be in region L
 - (b) For all neighbors, q , of p that have not been labeled
 - i. If the sign of Ψ is the same at q and p , and q has never been in queue #1, enqueue q into queue #1
 - ii. If the sign of Ψ is different at q and p , and q has never been in queue #2, enqueue q into queue #2
4. If queue #2 is not empty, continue to dequeue from queue #2 until an unlabeled pixel is found or until queue #2 is empty. If an unlabeled pixel, r , is found, enqueue it into

$L = 1$ $\Psi > 0$	$L = 2$ $\Psi \leq 0$	$L = 2$ $\Psi \leq 0$	$L = 2$ $\Psi \leq 0$	$L = 4$ $\Psi > 0$	$L = 6$ $\Psi \leq 0$
$L = 1$ $\Psi > 0$	$L = 1$ $\Psi > 0$	$L = 2$ $\Psi \leq 0$	$L = 2$ $\Psi \leq 0$	$L = 4$ $\Psi > 0$	$L = 4$ $\Psi > 0$
$L = 1$ $\Psi > 0$	$L = 1$ $\Psi > 0$	$L = 2$ $\Psi \leq 0$	$L = 4$ $\Psi > 0$	$L = 4$ $\Psi > 0$	$L = 4$ $\Psi > 0$
$L = 1$ $\Psi > 0$	$L = 1$ $\Psi > 0$	$L = 2$ $\Psi \leq 0$	$L = 2$ $\Psi \leq 0$	$L = 2$ $\Psi \leq 0$	$L = 2$ $\Psi \leq 0$
$L = 3$ $\Psi \leq 0$	$L = 1$ $\Psi > 0$	$L = 2$ $\Psi \leq 0$	$L = 2$ $\Psi \leq 0$	$L = 5$ $\Psi > 0$	$L = 5$ $\Psi > 0$
$L = 3$ $\Psi \leq 0$	$L = 1$ $\Psi > 0$	$L = 2$ $\Psi \leq 0$	$L = 2$ $\Psi \leq 0$	$L = 5$ $\Psi > 0$	$L = 5$ $\Psi > 0$

Figure 46: Example labeling for 6×6 pixel grid. Only input to the algorithm is the level set function, Ψ , at each pixel. Output of algorithm is a labeling, L , for each pixel into its appropriate connected region.

queue #1.

5. Increment L .

6. If queue #1 is empty terminate algorithm, otherwise go to step 3.

It is clear that the algorithm eventually terminates since points are only enqueued when they have not been already queued into a specific queue, and since eventually both queues will be emptied by the iterated dequeuing operations. It is also clear that the algorithm will eventually label all points since every point is the neighbor of the upper leftmost pixel through zero or more other pixels, and since every neighboring point is eventually either

labeled, placed into queue #1, or placed into queue #2 to be later transferred to queue #1. That the labels identify connected regions is not trivial, but can be seen from the fact that eventually, for a given label, L , all points that are neighbors and have the same sign of the level set function, will eventually be placed into queue #1, only to be dequeued and labeled before L is incremented. Thus, queue #1 is only emptied when a region is fully labeled. At this point, a single point is transferred from queue #2 into queue #1, seeding a new region. Then L is incremented and the procedure begins again, eventually labelling all points in this newly seeded region with the new value of L .

REFERENCES

- [1] ACTON, S., “Multigrid anisotropic diffusion,” *IEEE Trans. Image Proc.*, vol. 7, pp. 280–291, March 1998.
- [2] ALVINO, C. V., TANNENBAUM, A., YEZZI, A. J., and CURRY, C. W., “Multigrid computation of rotationally invariant non-linear optical flow,” in *International Conference on Image Processing*, 2005. In Review.
- [3] ALVINO, C. V. and YEZZI, A. J., “A scale space for contour registration using minimal surfaces,” in *Scale Space Theories in Computer Vision*, pp. 164–179, 2003.
- [4] ALVINO, C. V. and YEZZI, A. J., “Tomographic reconstruction of piecewise smooth images,” in *Computer Vision and Pattern Recognition*, vol. I, pp. 576–581, 2004.
- [5] ALVINO, C. V. and YEZZI, A. J., “Fast Mumford-Shah segmentation using image scale space bases,” in *IEEE International Conference on Computer Vision*, 2005. In Review.
- [6] ALVINO, C. V. and YEZZI, A. J., “Simultaneous segmentation and reconstruction of tomographic images using multiscale methods,” in *MICCAI*, 2005. In Preparation.
- [7] ASANO, T., KATOH, D. Z., and TOKUYAMA, T., “Polynomial-time solutions to image segmentation,” in *7th Ann. SIAM-ACM Conference on Discrete Algorithms*, pp. 104–113, Jan. 1996.
- [8] BATTITI, R., AMALDI, E., and KOCH, C., “Computing optical flow across multiple scales: An adaptive coarse-to-fine strategy,” *International Journal of Computer Vision*, vol. 6, no. 2, pp. 133–145, 1991.
- [9] BEAUCHEMIN, S. S. and BARRON, J. L., “The computation of optical flow,” *ACM Computing Surveys*, vol. 27, no. 3, pp. 433–467, 1995.
- [10] BRAKKE, K. A., *The Motion of a Surface by its Mean Curvature*. Princeton University Press, 1978.
- [11] BRANDT, A., “Multi-level adaptive solutions to boundary value problems,” *Math. Comput.*, vol. 31, pp. 333–390, 1977.
- [12] BRIGGS, W. L., HENSON, V. E., and MCCORMICK, S. F., *A Multigrid Tutorial*. SIAM, second ed., 2000.
- [13] BROWN, L., “A survey of image registration techniques,” *ACM Computing Surveys*, vol. 24, pp. 325–376, 1992.
- [14] BRUANDET, J.-P., PEYRIN, F., DINTEN, J.-M., and BARLAUD, M., “3D tomographic reconstruction of binary images from cone beam projections: A fast level set approach,” in *IEEE Symposium on Biomedical Imaging*, (Washington, D.C.), pp. 677–80, July 2002.

- [15] CASELLES, V., CATTE, F., COLL, T., and DIBOS, F., "A geometric model for active contours in image processing," *Numerische Mathematik*, vol. 66, pp. 1–31, 1993.
- [16] CASELLES, V., KIMMEL, R., and SAPIRO, G., "Geodesic active contours," *Int. J. Comput. Vis.*, vol. 22, no. 1, pp. 61–79, 1997.
- [17] CHAN, T. F. and VESE, L. A., "A level set algorithm for minimizing the Mumford-Shah functional in image processing," Tech. Rep. CAM 00-13, UCLA, Department of Mathematics, 2000.
- [18] CHEN, J., GALT, J. R., VALENTINE, J. D., FABER, T. L., and GARCIA, E. V., "Modelling SPECT acquisition and processing of changing radiopharmaceutical distribution," in *IEEE Nucl Sci Symp and Med Imag Conf*, pp. 1336–1370, 2001.
- [19] CHIUSO, A., BROCKETT, R., and SOATTO, S., "Optimal structure from motion: Local ambiguities and global estimates," *International Journal on Computer Vision*, vol. 39, no. 3, pp. 195–228, 2000.
- [20] CHOPP, D. L. *Journal of Computational Physics*, vol. 106, pp. 77–91, 1993.
- [21] CHOW, C. K. and KANEKO, T., "Automatic boundary detection of the left ventricle from cineangiograms," *Computers and Biomedical Research*, vol. 5, pp. 388–410, 1972.
- [22] ENKELMANN, W., "Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences," *Computer Vision, Graphics, and Image Processing*, vol. 43, pp. 150–177, 1988.
- [23] EVANS, L., *Partial Differential Equations*, vol. 19 of *Graduate Studies in Mathematics*. American Mathematical Society, 1998.
- [24] FABER, T. L., GARCIA, E. V., and LALUSH, D. S., "Simulating patient-specific heart shape and motion using spect perfusion images with the ncat phantom. visualization, display, and image guided procedures," in *Proceedings SPIE* (MUN, S. K., ed.), vol. 4319, pp. 22–26, 2001.
- [25] FELDMAR, J. and AYACHE, N., "Rigid, affine and local affine registration of free-form surfaces," *IJCV*, vol. 18, no. 2, pp. 99–119, 1996.
- [26] FOX, C., *An Introduction to the Calculus of Variations*. Dover Publications, Inc., 1987.
- [27] GLAZER, F., *Multi-Resolution Image Processing and Analysis*, ch. Multilevel relaxation in low-level computer vision, pp. 312–330. New York: Springer-Verlag, 1984.
- [28] HA, J., ALVINO, C. V., PRYOR, G., NEITHAMMER, M., JOHNSON, E. N., and TANENBAUM, A., "Active contours and optical flow for automatic tracking of flying vehicles," in *American Control Conference*, 2004.
- [29] HANSEN, B. and MORSE, B., "Multiscale registration using scale trace correlation," in *CVPR*, 1999.
- [30] HELGASON, S., *The Radon Transform*. Springer Verlag, 1999.
- [31] HERMAN, G. T., *Image Reconstruction from Projections: The Fundamentals of Computerized Tomography*. New York, NY: Academic Press, 1980.

- [32] HILDRETH, E. C., “Computations underlying the measurement of visual motion,” *Artificial Intelligence*, vol. 23, pp. 309–354, 1984.
- [33] HOFFMAN, D. and MEEKS, W. H., “Minimal surfaces based on the catenoid,” *Amer. Math. Monthly*, vol. 97, no. 8, pp. 702–730, 1990.
- [34] HORN, B. K. P. and SCHUNCK, B. G., “Determining optical flow,” *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.
- [35] INOUE, T., “Image reconstruction with limited angle projection data,” *IEEE Transactions on Nuclear Science*, vol. 26, pp. 2666–2684, 1979.
- [36] ISENBERG, C., *The Science of Soap Films and Soap Bubbles*. Dover Publications Inc., 1978.
- [37] JACKSON, J. D., YEZZI, A., WALLACE, W., and BEAR, M. F., “Segmentation of coarse and fine scale features using multi-scale diffusion and mumford-shah,” in *Scale Space Theories on Computer Vision*, pp. 615–624, 2003.
- [38] JIN, H., YEZZI, A., and SOATTO, S., “Region-based segmentation on evolving surfaces with application to 3D reconstruction of shape and piecewise constant radiance,” in *European Conference on Computer Vision*, vol. 2, pp. 114–125, 2004.
- [39] KASS, M., WITKIN, A., and TERZOPOULOS, D., “Snakes: Active contour models,” *Int. Journal of Computer Vision*, vol. 1, pp. 321–331, 1987.
- [40] KICHENASSAMY, S., KUMAR, A., OLVER, P., TANNENBAUM, A., and YEZZI, A., “Gradient flows and geometric active contour models,” in *ICCV*, pp. 810–815, 1995.
- [41] KREYSZIG, E., *Advanced Engineering Mathematics*. John Wiley and Sons, Inc., 8th ed., 1999.
- [42] KUMAR, A., TANNENBAUM, A. R., and BALAS, G. J., “Optical flow: A curve evolution approach,” *IEEE Transactions on Image Processing*, vol. 5, no. 4, pp. 598–610, 1996.
- [43] LESTER, H. and ARRIDGE, S. R., “A survey of hierarchical non-linear medical image registration,” *Pattern Recognition*, vol. 32, pp. 129–149, 1999.
- [44] MALLADI, R., SETHIAN, J., and VEMURI, B., “Shape modeling with front propagation: A level set approach,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 158–175, 1995.
- [45] MARR, D. and HILDRETH, E. C., *Theory of edge detection*. Proc. Roy. Soc. Lond., 1980.
- [46] MATEJ, S., HERMAN, G. T., and VARDI, A., “Binary tomography on the hexagonal grid using Gibbs priors,” *Journal of Imaging Systems and Technology*, vol. 9, pp. 126–131, 1998.
- [47] MUMFORD, D. and SHAH, J., “Boundary detection by minimizing functionals,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, (San Francisco, CA), pp. 22–26, June 1985.

- [48] MUMFORD, D. and SHAH, J., "Optimal approximation by piecewise smooth functions and associated variational problems," *Comm. Pure Appl. Math.*, vol. 17, pp. 577–685, 1989.
- [49] NAGEL, H. H., "Constraints for the estimation of displacement vector fields from image sequences," in *Int. Joint Conference on Artificial Intelligence*, (Karlsruhe, FRG), pp. 945–951, August 1983.
- [50] OLSON, T. and JAFFE, J. S., "An explanation of the effects of squashing in limited angle tomography," *IEEE Trans. on Med. Imag.*, vol. 9, pp. 242–246, Sept. 1990.
- [51] OSHER, S. and SETHIAN, J., "Fronts propagating with curvature-dependent speed: algorithms based on the Hamilton-Jacobi equations," *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.
- [52] PERONA, P. and MALIK, J., "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 629–639, July 1990.
- [53] PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., and FLANNERY, B. P., *Numerical Recipes in C*. Cambridge University Press, second ed., 1992.
- [54] PRINCE, J. L. and WILLSKY, A. S., "Hierarchical reconstruction using geometry and sinogram restoration," *IEEE Trans. on Image Proc.*, vol. 2, pp. 401–416, July 1993.
- [55] RAMM, A. G. and KATSEVICH, A. I., *The Radon Transform and Local Tomography*. Boca Raton, FL: CRC Press Inc., 1996.
- [56] REEDS, J. A. and SHEPP, L. A., "Limited angle reconstruction in tomography via squashing," *IEEE Trans. on Med. Imaging*, vol. 6, pp. 89–97, June 1987.
- [57] RIDLER, T. W. and CALVARD, S., "Picture thresholding using an iterative selection method," *IEEE Trans. Syst., Man., Cybern.*, vol. 8, pp. 630–632, Aug. 1978.
- [58] ROBERT, N., PEYRIN, F., and YAFFE, M., "Binary reconstruction from a limited number of cone beam projections," *Medical Physics*, vol. 21, no. 12, pp. 1839–1851, 1994.
- [59] RONFARD, R., "Region-based strategies for active contour models," *Int. J. Computer Vision*, vol. 13, pp. 229–251, 1994.
- [60] SEBASTIAN, T. B., KLEIN, P. N., and KIMIA, B. B., "Alignment-based recognition of shape outlines," in *IWVF*, pp. 606–618, 2001.
- [61] SEGARS, W. P., LALUSH, D. S., and TSUI, B. M. W., "A realistic spline-based dynamic heart phantom," *IEEE Trans Nucl Sci*, vol. 46, pp. 503–506, 1999.
- [62] SETHIAN, J. A., *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
- [63] SEZAN, M. I. and STARK, H., "Tomographic image reconstruction from incomplete view data by convex projections and direct fourier inversion," *IEEE Trans. on Med. Imaging*, vol. 3, pp. 91–98, 1984.

- [64] STRANG, G., *Linear Algebra and Its Applications*. Harcourt Brace Jovanovich, 3rd ed., 1988.
- [65] STRIKWERDA, J. C., *Finite Differences and Partial Differential Equations*. Society for Industrial and Applied Mathematics, 2 ed., 2004.
- [66] TAGARE, H. D., O'SHEA, D., and RANGARAJAN, A., "A geometric criterion for shape based non-rigid correspondence," in *Fifth Intl. Conf. on Computer Vision (ICCV)*, pp. 434–439, 1995.
- [67] TERRY, J. A., TSUI, B. M. W., PERRY, J. R., HENDRICKS, J. L., and GULLBERG, G. T., *Biomedical Engineering: Opening New Doors*, ch. The design of a mathematical phantom of the upper human torso for use in 3-D SPECT imaging research, pp. 185–190. New York: New York University Press, 1990.
- [68] TERZOPOULOS, D., "Image analysis using multigrid relaxation methods," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 129–139, 1986.
- [69] TERZOPOULOS, D., WITKIN, A., and KASS, M., "Constraints on deformable models: Recovering 3D shape and nonrigid motions," *Artif. Intell.*, vol. 36, pp. 91–123, 1988.
- [70] TSAI, A., YEZZI, A. J., and WILLSKY, A. S., "Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification," *IEEE Trans. Image Proc.*, vol. 10, pp. 1169–1186, August 2001.
- [71] WEST, J. B., FITZPATRICK, J. M., and BATCHELOR, P. G., "Point-based registration under a similarity transform," in *Proceedings SPIE*, 2001.
- [72] WHITAKER, R. and ELANGOVA, V., "A direct approach to estimating surfaces in tomographic data," *Journal of Medical Image Analysis*, vol. 6, no. 3, pp. 235–249, 2002.
- [73] YEZZI, A., KICHENASSAMY, S., KUMAR, A., OLVER, P. J., and TANNENBAUM, A., "A geometric snake model for segmentation of medical imagery," *IEEE Trans. Med. Imaging*, vol. 16, no. 2, pp. 199–209, 1997.
- [74] YEZZI, A. and SOATTO, S., "Deformation: Deforming motion, shape average and the joint registration and approximation of structures in images," *International Journal of Computer Vision*, vol. 53, no. 2, pp. 153–167, 2003.
- [75] YEZZI, A. and SOATTO, S., "Stereoscopic segmentation," *International Journal of Computer Vision*, vol. 53, no. 1, pp. 31–43, 2003.
- [76] YEZZI, A., 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.
- [77] YU, D. and FESSLER, J., "Edge-preserving tomographic reconstruction with nonlocal regularization," in *Proceedings of IEEE Int. Conf. on Image Proc.*, vol. 1, (Chicago, IL, USA), pp. 29–33, Oct. 1998.
- [78] ZACHMANOGLU, E. C. and THOE, D. W., *Introduction to Partial Differential Equations with Applications*. Dover Publications, Inc., 1986.

- [79] ZHANG, Z., “Iterative point matching for registration of free-form curves and surfaces,” *IJCV*, vol. 13, no. 2, pp. 147–176, 1994.
- [80] ZHU, S. and YUILLE, A., “Region competition: Unifying snakes, region growing, and bayes/MDL for multiband image segmentation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 884–900, 1996.

VITA

Christopher V. Alvino was born in Long Branch, New Jersey in 1976, and received the B.S. (highest honors) and M.S. degrees in Electrical and Computer Engineering (ECE) from Rutgers University, New Brunswick, NJ in 1998 and 2001, respectively. He has held several industry internships including Lucent Technologies, Lutron Electronics, and Sarnoff Corporation. He was a full-time signal processing consultant at Sarnoff Corporation, Princeton, NJ from May 2000 through August 2001. In August of 2001, he began at the Georgia Institute of Technology under the direction of Professor Anthony Yezzi and received the Ph.D. degree in ECE in May 2005 with a minor in Mathematics.

He was a Garden State Scholar throughout his undergraduate studies. In 1998, he was selected by the Rutgers University, Electrical and Computer Engineering faculty to receive the James Leroy Potter Award for original independent investigation by an undergraduate. He received the Georgia Institute of Technology President's Fellowship throughout the duration of his Ph.D. studies. He has published and submitted four journal papers and eight refereed conference papers, holds one U.S. patent, and has given two talks at major computer vision conferences.