Abstract

Estimation of 3D Left Ventricular Deformation

from Medical Images Using Biomechanical Models.

Xenophon Papademetris

2000

The non-invasive quantitative estimation of regional cardiac deformation has important clinical implications for the assessment of viability in the heart wall. In this work we describe a general framework for estimating soft tissue deformation from sequences of three-dimensional medical images. We also explore some of their theoretical constraints which can be used to guide the selection of an appropriate model for the displacement field. We then apply this framework to the problem of estimating left ventricular deformations from sequences of 3D image sequences. The images are segmented interactively to extract the endocardial and epicardial surfaces. Then, initial frame-to-frame correspondences are established between points on the surfaces using a shape-tracking approach. The myocardium is modeled using a transversely isotropic linear elastic model, which accounts for the preferential stiffness of the left ventricular myocardium along its fiber directions. The measurements and the model are integrated within a Bayesian estimation framework. The resulting equations are solved using the finite element method, to produce a dense displacement field for the whole of the left ventricle. The dense displacement field is, in turn, used to calculate the deformation of the heart wall in terms of the strains. This method was tested on over 40 image sequences, and the strains produced using this non-invasive technique exhibit high correlation with strains simultaneously obtained from invasive measurements using implanted markers and sonomicrometers. We also demonstrate that these strains are useful as predictors of the viability of the underlying tissue and can be used to distinguish between classes of subjects in which there was moderate or severe injury. This proposed method provides quantitative regional 3D estimates of left ventricular deformation from three-dimensional sequences of Magnetic Resonance, Ultrasound, and X-Ray CT images.

Estimation of 3D Left Ventricular Deformation from Medical Images Using Biomechanical Models

A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by
Xenophon Papademetris

Dissertation Director: James Scott Duncan

May 2000

Acknowledgments

I would first like to acknowledge the support, guidance and help from my advisor, James Duncan. I am grateful for the guidance from Gary Povirk and Turan Onat from the Mechanical Engineering department for allowing an electrical engineer into their circles, and having the patience to spend the time needed to understand problems outside their usual line of work. I would also like to acknowledge the support and help from Peter Belhumeur and David Kriegman, especially during my first two years at Yale. Thanks also to Peter Schultheis, Arie Nehorai and Roman Kuc for at different times serving on my committee. I would like to acknowledge the help of Larry Staib for, and I quote from another thesis that came out of this laboratory, "his infinite patience and willingness to talk about any subject at any time." I would like to thank Dimitris Metaxas for serving as my external reader.

The primary focus of the experimental results in this thesis is the estimation, validation and applicability of image-derived 3D strains from left ventricular image sequences. This would not have been possible without the cooperation of a number of individuals which the author would like to acknowledge at this point. The surgical preparation, implantation of markers/sonomicrometers and the processing of the results from these, as well as the postmortem information and regional blood flow measurements and 3DE images acquisition, was done in the Nuclear Cardiology Laboratory, Yale School of Medicine under the direction of Dr Albert Sinusas with the help of Don Dione, Jason Soares, David Meoli, Jennifer Hu and their collaborators. They also helped with the analysis of the strain results. The MR images were acquired under the supervision of Dr Todd Constable of the MR Physics Group, also at Yale. The DSR Images were acquired under the supervision of Dr Eric Ritman at Mayo Clinic. Farah Janzad, Donna Dione, Peter Borek, Don Dione and others also helped with the segmentation of the various image datasets.

This thesis directly builds on the thesis work of Amit Chakraborty and Pengcheng Shi, whom I would also like thank. Pengcheng, in the beginning stages of my dissertation, was almost like a second advisor. Amit provided me with his implementation of some the texture segmentation algorithms used in this work. I would also like to acknowledge the help of Hemant Tagare, who generously allowed me access to his implementation of the B-snake algorithm which formed the basis of my own implementation. His differential geometry seminars where also extremely helpful. Thanks also to Francois Meyer and Anand Rangaranjan for many useful discussions.

I would also like to thank all the current members and the alumni of the Image Processing and Analysis Group at Yale, for making this a pleasant place to work (in alphabetical order with apologies to anyone whose name I have accidentally left out) Amir Amini, Sudha Chelikani, Haili Chui, Melissa Koudelka, John McEachen, Reshma Munbodh, Suguna Pappu, Jim Rambo, Glynn Robinson, Oskar Skrinjar, Rik Stokking, Colin Studholme, Yongmei Wang, Lawrence Win, Xiaolan Zeng, George Zubal. Thanks also to Carolyn Meloling for all her help.

Thanks also to Genzo and Ellen Yamamoto and David Lumsdaine for their help and advice on numerous topics.

Finally I would like to thank the members of Holy Transfiguration Orthodox Church, here in New Haven, for their support, my family especially my parents Panayiotis and Maria and my wife Susan to whom this thesis is dedicated.

Contents

Acknowledgements **i**

Table of Contents **iii**

List of Figures **vii**

List of Tables **x**

1 Introduction **1**

1.1 Structure of the Thesis 1

1.2 Introduction to the Problem 2

1.3 Contributions of this Work 3

1.4 A Personal Note on Methodology 4

2 Cardiac Image Analysis **7**

2.1 Introduction 7

2.2 Invasive Approaches to Measuring Myocardial Deformation. 10

2.3 Approaches to Obtaining Estimates of Cardiac Deformation from 4D Images 11

2.3.1 Methods Relying on Magnetic Resonance Tagging 12

2.3.2 Methods Relying on Phase Contrast MRI 16

2.3.3 Computer Vision Based Methods 18

2.4 Modeling used for Interpolation and Smoothing 20

2.5 Validation of Results 25

2.6 Conclusions and Further Research Directions 28

3 An Interactive Approach to Left Ventricular Segmentation 30

3.1 Introduction 30

3.2 Parameterizing Closed Curves Using B-splines 31

3.3 A B-spline Snake Implementation 37

3.3.1 The Internal Energy Functional 37

3.3.2 The External Energy Functional 38

3.4 An Interactive Surface Segmentation Platform 46

4 Geometrical Background and Techniques 50

4.1 Geometrical Methods in Two-dimensions 50

4.1.1 Shape-Based Interpolation of Contours 50

4.1.2 Symmetric Nearest Neighbor Correspondences in Curves 53

4.2 Geometrical Methods in Three-dimensions 57

4.2.1 Delaunay Triangulation Between Planar Contours 57

4.2.2 Non-Shrinking Surface Smoothing 61

4.2.3 Curvature Computation 62

4.2.4 Symmetric Nearest Neighbor Correspondences in Surfaces 66

4.3 Generating Hexahedral Meshes 71

4.4 A Shape-Based Tracking Algorithm 74

5 Continuum Mechanics Models and the Finite Element Method 77

5.1 Deformations 78

5.1.1 The Deformation Gradient Matrix 78

5.1.2 Small Deformations and Rotations 81

5.1.3 Finite Deformations 82

5.1.4 Some Further Properties of the Strain Tensor 82

5.2 Material Models 83

5.2.1 Linear Elastic Energy Functions 84

5.2.2 Non-Linear Energy Functions 86

5.3 The Finite Element Method 87
5.3.1 An Example Problem 87
5.3.2 Outline of the Solution Procedure 88

6 Modeling the Displacement Field 94

6.1 The General Regularization Framework 94
6.1.1 The Energy Minimization Framework 94
6.1.2 A Probabilistic Interpretation 95
6.1.3 Advantages of the Probabilistic Interpretation 97
6.1.4 The Problem of Different Units 98
6.1.5 Soft Tissue Objects as Markov Random Fields 99
6.2 A First-Order Regularization Functional 101
6.2.1 The Two Dimensional Dense Case 101
6.2.2 Relationship with Infinitesimal Linear Elasticity 103
6.2.3 The Discrete Spring Model 104
6.2.4 A Second-Order Regularization Functional and the Affine Model . 105
6.3 The Use and Abuse of Linear Elasticity 106
6.3.1 Zero Stiffness 107
6.3.2 Bias Correction 108
6.3.3 The History-Free Approach 109
6.3.4 Fluid Model 109
6.4 Active Elastic Models 111
6.4.1 Problems With Passive Models 111
6.4.2 A Proposed Extension 112
6.4.3 A Hierarchal Estimation Scheme for Finding the Active Component 112

7 Estimating Left Ventricular Deformation 114

7.1 Image Acquisition 114
7.1.1 Canine MR-images 114
7.1.2 3D Echocardiography (3DE) 115

7.1.3 Human MR-images . . . 116

7.1.4 Dynamic Spatial Reconstructor Data . . . 116

7.2 Image Analysis . . . 116

7.2.1 Segmentation and Shape-Based Tracking . . . 116

7.2.2 Modeling the myocardium . . . 117

7.2.3 Integrating Model and Data . . . 119

7.2.4 Numerical Solution . . . 120

7.2.5 Strain Analysis . . . 120

7.2.6 Measures of Myocardial Viability . . . 121

7.3 *In-Vivo* Validation . . . 123

7.3.1 Implanted Image-Opaque Markers: . . . 123

7.3.2 Sonomicrometers . . . 126

7.4 Results . . . 128

7.4.1 Canine MRI . . . 128

7.4.2 3D Echocardiography . . . 133

7.4.3 Human MRI . . . 137

7.4.4 DSR . . . 137

7.5 Conclusions . . . 137

Bibliography **139**

List of Figures

2.1 Geometry of the Mammalian Heart. 8

2.2 Short-axis magnetic resonance images from two 3D acquisitions of a canine heart. 9

2.3 Typical placement of arrays of sonomicrometer crystal (or implanted bead) arrays in the left ventricle. 10

2.4 Magnetic Resonance Tagging Images 12

2.5 Reconstruction of 3 perpendicular tagging planes acquired in different acquisitions. 13

2.6 An example of a low-frequency tagged MRI image. 15

2.7 Three-slice thick volumetric dataset obtained using magnetic resonance phase contrast images. 17

2.8 MR image of gel phantom with SPAMM (tag) stripes. 26

2.9 Example of the use of the cardiac simulator used to validate methods based on MR tagging. 27

2.10 2D MR image slice of left ventricle with implanted markers. 27

3.1 The elements of a B-spline. 32

3.2 Approximation of a function with B-splines. 34

3.3 External Energy Functions for intensity and intensity+texture snakes. . . 41

3.4 Ultrasound images and surfaces. 41

3.5 Use of multiple external energy functions. 42

3.6 Temporal contour interpolation. 47

3.7 3D Viewer. 48

3.8 The Editor Module. 49

3.9 A long axis view. 49

4.1 Chamfer transformation templates. 51

4.2 Extracting Zero Crossings. 52

4.3 Illustration of problems with asymmetric nearest neighbor matches. . . . 54

4.4 An example of the 2D implementation of the symmetric nearest neighbor algorithm. 56

4.5 Steps involved in moving from slice by slice contours to full surface representation. 58

4.6 Schematic for the proof of the optimality of the triangulation procedure. . 60

4.7 Delaunay Triangulation. 61

4.8 Local parameterization for a surface. 64

4.9 Symmetric Nearest Neighbor Algorithm in 3D. 67

4.10 Symmetric 3D Nearest Neighbor Algorithm. 69

4.11 A schematic of the mesh generation process. 72

4.12 A further illustration of the mesh generation process. 73

4.13 The shape-tracking algorithm. 75

5.1 Definition of displacement 78

5.2 Fiber direction in the left ventricle 85

5.3 3D Hexahedral Mesh. 88

5.4 2D and 3D Isoparametric Elements. 89

6.1 Illustration of a Markov Random Field. 100

6.2 A spring connected object. 104

6.3 Example of bias on the maximum a-posteriori approach. 108

7.1 Image acquisition geometry for the 3DE images. 115

7.2 Coordinate system used to define fiber orientation. 118

7.3 Division of the left ventricle(LV) into slices and sectors. 121
7.4 Labeling of sectors based on TTC staining. 122
7.5 Implantation of Image-Opaque Markers. 124
7.6 Localization of implanted markers. 124
7.7 Algorithm-derived Strains vs. Implanted Marker-derived Strains. 125
7.8 3DE Algorithm-Derived Strains vs. Sonomicrometer-derived Strains. 127
7.9 Average strain information at baseline for N=8 dogs. 128
7.10 Baseline (normal) canine LV strains derived from cine-MRI using the algorithm 129
7.11 Strain Development in Post-Infarction (and vs. Baseline) Canine LV derived from cine-MRI. 130
7.12 Average radial and circumferential strains for different *postmortem*-classified regions, ... 131
7.13 Results from Nontransmural ($N = 6$) vs. Transmural ($N = 6$) Acute Canine Studies. 132
7.14 "3DE-tissue-tagging" 133
7.15 3D Strain development from Ultrasound. 134
7.16 Human cine-MRI-derived results. 136
7.17 Canine DSR-derived results. 136

List of Tables

7.1 Results from 4 animal studies using 3D Echocardiography. 135

Chapter 1

Introduction

1.1 Structure of the Thesis

The major goal of this thesis is the development of an approach for the estimation of three-dimensional left ventricular deformation from medical images derived from different modalities. A secondary goal is the development of a more general framework for the estimation of soft-tissue deformation from medical images.

The thesis reads as follows: chapter 2 is an extended literature review for the area of cardiac image analysis with a special emphasis on the techniques used which would be of more general application in the area of soft-tissue deformation. In chapter 3 we present material relating to the problem of left ventricular segmentation. The segmented endocardial and epicardial surfaces are the inputs to the geometrical techniques of chapter 4. Chapter 4 itself provides some geometrical background and describes two key applications of geometrical ideas in this work, namely, 3D mesh generation and shape-based tracking. The background material concludes with chapter 5. Here we present material relating to continuum mechanics and a brief description of the finite element method.

In chapter 6 we discuss issues related to the development of a general framework for the estimation of soft-tissue deformation from sequences of 3D medical images. Finally, in chapter 7 we present experimental results and validation for the application of the overall methodology to the problem of left ventricular estimation.

1.2 Introduction to the Problem

The estimation of soft tissue deformation is related to the general non-rigid motion problem in Computer Vision and especially the problem of optical flow estimation [46]. Since deformation measures are calculated as combinations of the derivatives of displacement fields, the key problem in this line of work is the estimation of a dense and noise-free displacement field for the region of interest. Once this displacement field has been estimated, the deformation can be calculated.

In areas such as surgical training and image guided surgery, the displacement field is what is actually needed. The deformation measures themselves become important as measures of function of actively deforming organs such as the left ventricle. It is the general consensus that the analysis of heart wall deformation provides quantitative estimates of the location and extent of ischemic myocardial injury.

The major problem faced here is that is in general difficult to obtain dense displacement fields from medical images. In practice the displacement field can be measured only at sparse locations in the region of interest and these measurements are often corrupted by noise. The key to solving this deformation estimation problem is the techniques used to smooth and interpolate these sparse displacements in order to obtain a dense displacement field for the whole object. The selection of an appropriate model is constrained by many factors such as lack of knowledge about the underlying material properties and computational cost.

In this work we describe a methodology for estimating soft-tissue deformation from image derived information. We review a number of approaches proposed in the literature and propose our own extensions to account for some of the problems. We use this methodology to estimate left ventricular deformations from 3D medical images obtained using different modalities, primarily Magnetic Resonance and Echocardiography. The images are segmented interactively and then initial correspondence is established using a shape-tracking approach. A dense motion field is then estimated using a transversely anisotropic linear elastic model, which accounts for the fiber directions in the left-ventricle. The dense motion field is in turn used to calculate the deformation of the heart wall in terms of strain in cardiac specific directions. The strains obtained using this approach in open-chest dogs before and after coronary occlusion, show good agreement with previously published results in the literature. They also exhibit a high correlation with strains produced in the same animals using invasive techniques such as implanted markers and sonomicrometers. This proposed method provides quantitative regional 3D estimates of heart deformation from 3D Images.

1.3 Contributions of this Work

There are two major contributions of this work:

- The in-detail analysis and comparisons of various approaches to modeling the displacement field as used in many medical image analysis problems. We also identify similarities and problems with these approaches and propose a new approach to deal with many of these deficiencies. We call this new model the *active elastic model*.

- The development of a framework for accurate and reliable 3D left ventricular deformation estimation from medical images, including techniques for image segmentation. Of paramount importance here was the proper integration of biomechanics with image analysis techniques. This framework has been tested on a large number of studies and the results are shown to correlate well with invasive measures of deformation as well as other indicators of myocardial function.

We also note that there are some less substantial contributions in the area of interactive segmentation. We also developed some interesting geometric techniques to solve problems such as mesh generation and nearest neighbor estimation in three-dimensions.

1.4 A Personal Note on Methodology

Alexander Solzhenitsyn in this Nobel Lecture [1] tries to capture two possible attitudes to art. He writes:

> “One artist sees himself as the creator of an independent spiritual world; he hoists onto his shoulders the task of creating this world, of peopling it and of bearing the all-embracing responsibility for it; but he crumples beneath it, for a mortal genius is not capable of bearing such a burden. Just as man in general, having declared himself the center of existence, has not succeeded in creating a balanced spiritual system. And if misfortune overtakes him, he casts the blame upon the age-long disharmony of the world, upon the complexity of today’s ruptured soul, or upon the stupidity of the public.
>
> Another artist, recognizing a higher power above, gladly works as a humble apprentice beneath God’s heaven; then, however, his responsibility for everything that is written or drawn, for the souls which perceive his work, is more exacting than ever. But, in return, it is not he who has created this world, not he who directs it, there is no doubt as to its foundations; the artist has merely to be more keenly aware than others of the harmony of the world, of the beauty and ugliness of the human contribution to it, and to communicate this acutely to his fellow-men. And in misfortune, and even at the depths of existence–in destitution, in prison, in sickness–his sense of stable harmony never deserts him.[2]”

[1] This lecture was delivered only to the Swedish Academy and was not actually given as a lecture, as Solzhenitsyn could not leave the Soviet Union at the time (1970).

[2] Often at the end of some of my many discussions with Prof Turan Onat, I could see the contrast between the two approaches. Where I would see problem after problem and tried to force a solution and move on, he would often, to my frustration, be in a state of wonder and curiosity at the intricacy and almost ‘perfection’ of the left ventricle. Much of the work on the active models in this thesis is directly derived from this sense of wonder, and an attempt to understand it.

In many respects one finds analogues to the above expressions in the attempt to devise solutions to complicated engineering problems. In the case of the estimation of left ventricular deformation (and soft tissue deformation in general) a number of choices need to be made which place the engineer in one of two categories above. For example, consider the problem of modeling the displacement field itself. Does one try to use a method that tries to approximate in some way the real properties of the material (Solzhenitsyn's second category) or does one try to find a model which is driven more by convenience and computational requirements, such as a more generic smoothness model? What is the next step, if the approach appears to not work satisfactorily? How much is the methodology driven by the data itself or how much are we trying force existing approaches on to the problem?

Looking through the contents of the thesis, in retrospect (after the work was completed) one finds a mixed bag.[3] The segmentation work is clearly in the first category where we try to force our own convenience and models onto the problem by segmenting a 3D (if not 4D) object in a slice-by-slice basis. The ability to see 3D surface reconstructions in almost real time tries to mitigate this deficiency somewhat. In the geometry work, the 'symmetric' nearest-neighbor is a step towards letting the problem dictate, but the shape-based tracking work is still very much asymmetric (unlike the bimorphism work [98].) In the review of the various techniques for modeling displacement fields we point out the pitfalls of trying to force seemingly innocent ideas such as smoothing onto the real world. The blind use of linear elasticity is also seen to be problematic. The active elastic model which tries to capture the reality of an actively deforming tissue offers the promise of solving such problems in the future. For the same reason, while using continuum mechanics models to model the tissue, we avoid terms such as 'stress' and 'force' because these would be referring to simulated data 'forces' and not their physical analogues. Attempts to calculate the stress on the myocardial wall without accounting for the wall pressure are doomed to fail even though a quantity labeled 'stress' is available after the deformation analysis.

[3] Clearly for Solzhenitsyn, and for this author as well, the second category is the preferable one.

Perhaps the most telling single experiment was the attempt to see whether the methodology of this thesis could be used to distinguish between animals where there was *post mortem*-confirmed globally transmural as opposed to nontransmural injury (see section 7.4.1). In this case, the *cardiac specific* strains, which amount to forcing the left ventricular deformation to measured in a cylindrical coordinate frame, failed to produce a significant difference. Using the *principal strains* instead which are the major directions of deformation of the material irrespective of the external coordinate system, led to the desired outcome. This is a clear case when letting the data dictate led to a better answer than our preconceived notions of how things ought to work.

Chapter 2

Cardiac Image Analysis

In this chapter we describe research in the area of estimation of cardiac motion and deformation from medical images. We focus primarily on the use of 3D magnetic resonance image sequences, but we will also discuss the application of some methods to ultrafast CT and 3D echo.

2.1 Introduction

The estimation of cardiac motion and deformation from 3D images has been an area of major concentration in medical image analysis. In these problems, the image data utilized are typically acquired in 16–20 frames consisting of 10–16 slices each in the case of Magnetic Resonance. One such image slice through a canine heart acquired using magnetic resonance imaging is shown in figure 2.1 (as well as a reconstructed long-axis slice). In the figure, we label major areas such as the left and right ventricles and the two ventricular walls which bound the left ventricular myocardium (the endocardium and the epicardium). Most researchers have focused almost exclusively on the motion and deformation of the left ventricle. More recently, however, some preliminary work on right ventricular deformation has also appeared in the literature [42].

The estimation of regional 3D cardiac deformation is an important issue as ischemic heart disease is a major clinical problem. Myocardial injury caused by ischemic

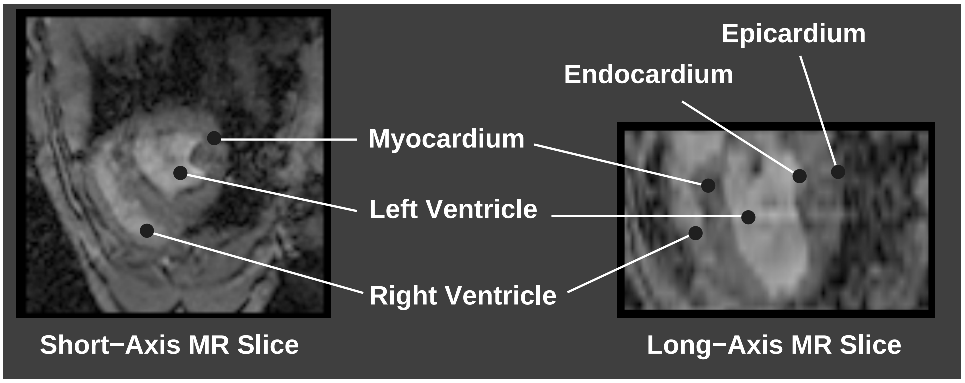
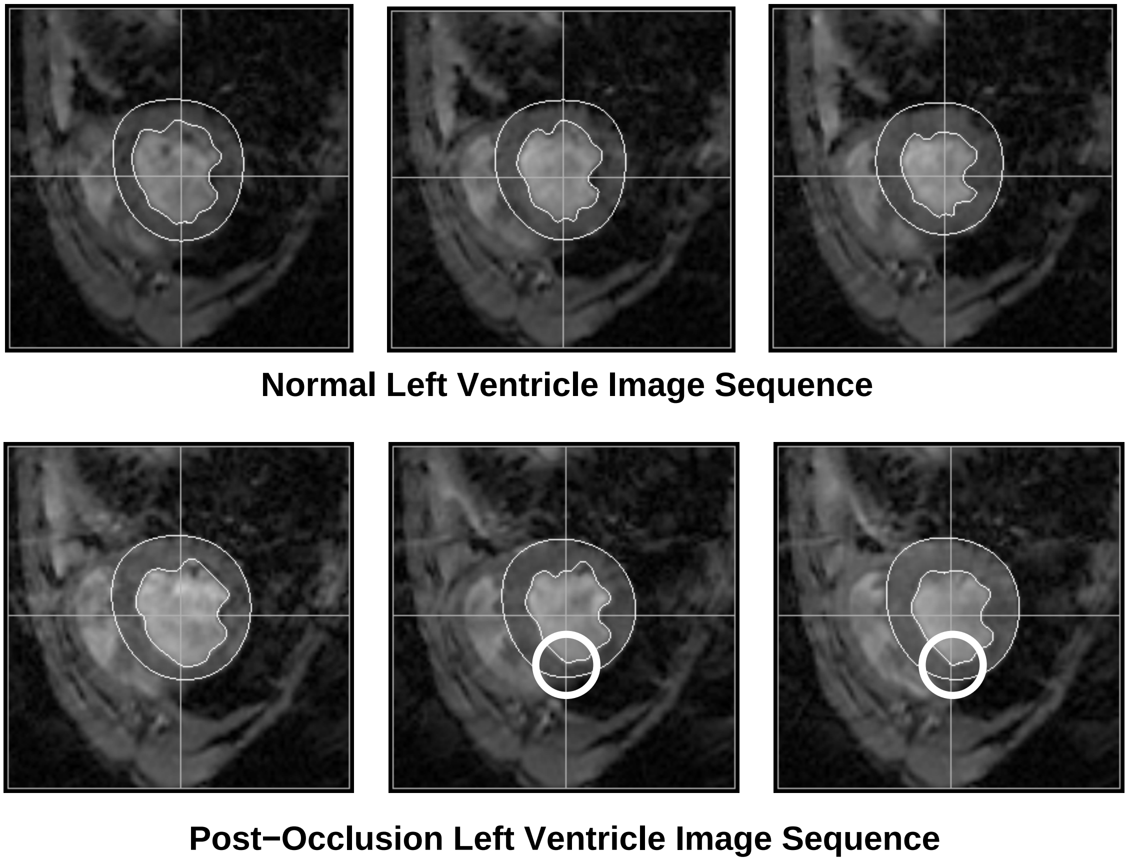


Figure 2.1: Geometry of the Mammalian Heart. In the discussion to follow the terms endocardium and epicardium will be used to refer to the bounding surfaces of the left ventricular myocardium.

heart disease is often regional. It is the fundamental goal of many forms of cardiac imaging and image analysis to measure the regional function of the left ventricle (LV) in an effort to isolate the location and extent of ischemic or infarcted myocardium. Figure 2.2 illustrates the effect of a blocked artery; in this case the left-anterior descending artery (LAD) has been occluded. There is a change in the deformation in a local region which is supplied by the LAD, which instead of the normal thickening behavior, actually thins on contraction. Quantitative estimation of these changes is a major goal of cardiac image analysis, as it will hopefully allow for the measurement of both the location and the extent of the affected region.

In addition, the current management of acute ischemic heart disease is directed at establishing coronary reperfusion and, in turn, myocardial salvage. Also, understanding the physiology of the heart is an important research problem in cardiology, for the evaluation of various surgical procedures such as Transmyocardial Revascularisation [36].

The rest of this chapter reads as follows: In section 2.2, we briefly describe alternative invasive techniques to estimating cardiac deformation, involving surgically implanted beads or ultrasound transducers. Then in sections 2.3 and 2.4, we turn our attention to



Normal Left Ventricle Image Sequence

Post−Occlusion Left Ventricle Image Sequence

Figure 2.2: Short-axis magnetic resonance images from two 3D acquisitions of a canine heart. The top sequence was acquired before left coronary anterior artery occlusion and the bottom sequence post-occlusion. The occlusion generates a disruption of the normal thickening behavior of the myocardium in contraction in the highlighted region. The quantification of such parameters from 3D image sequences is the focus of methods reviewed in this chapter.

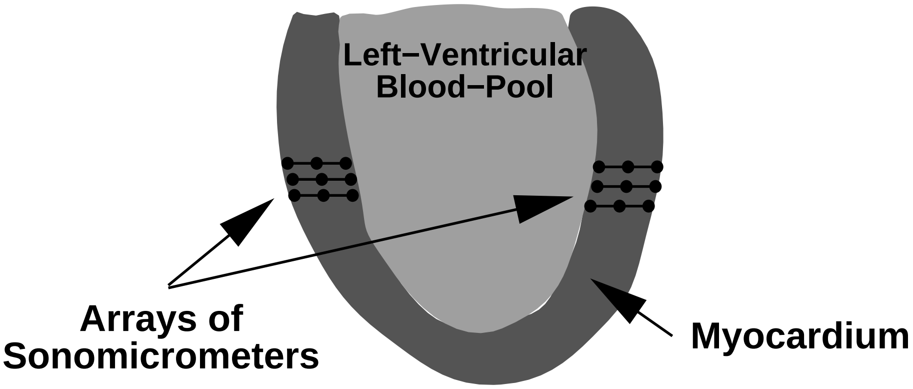


Figure 2.3: Typical placement of arrays of sonomicrometer crystal (or implanted bead) arrays in the left ventricle. These can produce highly accurate estimates of the deformation at a small number of locations in the left ventricle.

describing current and previous research efforts in the medical imaging community with respect to estimating Cardiac Motion and Deformation. Typically, any given method will combine a set of sparse, noisy, image derived and sometimes partial set of displacement estimates (the ‘data’) with a model which is used to simultaneously smooth and interpolate these estimates as necessary (the ‘model’). This combination of ‘data’ and ‘model’ produces the resulting displacement field. We will first analyze the ‘data’-component of the presented methods in section 2.3 and the ‘model’-component in section 2.4. Next in section 2.5 we turn to the all important topic of validation. Finally, in section 2.6 we present some possible future research directions in this area.

2.2 Invasive Approaches to Measuring Myocardial Deformation.

A variety of work is evident in the cardiac physiology literature attempting to quantitatively measure transmural myocardial strain. Several noteworthy efforts in particular have used sonomicrometers [35, 34, 27] and arrays of implanted markers (see, for example, [104, 68]). Figure 2.3 shows a schematic of a typical implantation of sonomicrometers in the left ventricle. While accepted as being accurate, in both cases only a sparse number of specific sites on the LV can be measured, due to the difficulty in implanting the sonomicrometers and the markers. It would be quite difficult to measure a large number of sites simultaneously.

Also, it is possible that these implanted devices can alter myocardial perfusion and function, although there is little published evidence of this. While many of these measurements are performed in animals, we note that some interesting measurements of strain using markers have been produced even in humans [52]. Finally, we also note that some researchers have looked at measuring *in vivo* strain using attached strain gauges [26] (as noted in Azhari [7]), although little has been pursued along these lines.

2.3 Approaches to Obtaining Estimates of Cardiac Deformation from 4D Images

There are two aspects to this problem; the first relates to the manipulation of the acquisition parameters to obtain the most useful images and the second to the post-processing of these images to estimate cardiac deformation. Regarding the first aspect, a significant level of activity has been performed within the magnetic resonance imaging (MRI) community regarding the development of MR tagging, and to a lesser extent, MR phase velocity imaging. The underlying physics of these techniques is beyond the scope of this chapter; the interested reader is referred to a review article by Leon Axel [6].

The second aspect of this problem, the analysis of the images, relates to work traditionally done in the computer vision community, especially in the areas of non-rigid motion estimation, including the case of variable illumination, segmentation and surface mapping. A general, although somewhat dated, coverage of the field can be found in Horn [46].

In this section, we focus on the image-derived characteristics used to obtain the initial somewhat sparse, often noisy and partial displacements and/or velocities which

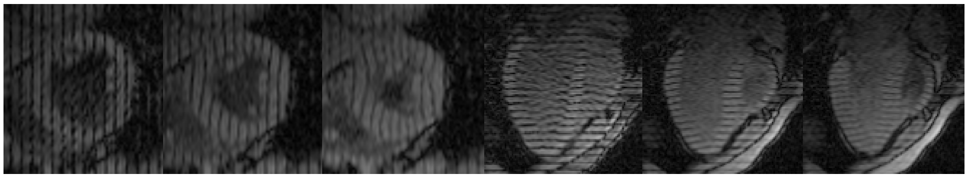


Figure 2.4: Samples of short-axis and long-axis magnetic resonance images illustrating magnetic resonance tagging at 3 time points in the cardiac cycle. Courtesy of Dr Jerry L. Prince, John Hopkins University.

are combined with a model to produce complete and dense displacement and deformation estimates.

2.3.1 Methods Relying on Magnetic Resonance Tagging

In this approach, grid lines at certain positions can be generated at one point in the cardiac cycle and their deformation tracked over a portion of the cycle, primarily using gated acquisition techniques. The development of the grid tagging approach to the measurement of myocardial strain has been vigorously pursued by two groups in particular, at the University of Pennsylvania [6] and Johns Hopkins [67], who are the original developers of the tagging ideas. Figure 2.4 shows an example of such an acquisition. Three frames are shown. In frame 1 the original tags are laid out parallel to the vertical axis and are shown to deform with the material in the subsequent frames.

Much of these groups' current efforts are focused on how to create dense fields of measurements in 3D by putting together several orthogonal tagging grid acquisitions. Their approaches certainly show promise, because of the inherent capability of including discernible patterns that deform with the tissue, but currently have the following limitations: i.) it is difficult to track the tags over the complete LV cycle due to decay of the tags with time, ii.) multiple acquisitions are required to assemble 3D information and iii) it is still quite difficult to assemble the detected tags into a robust 3D analysis/display. All of these problems are being aggressively pursued by the two primary groups mentioned

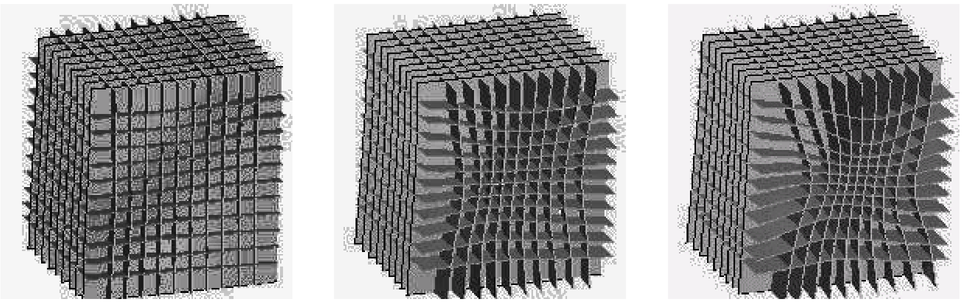


Figure 2.5: Reconstruction of 3 perpendicular tagging planes acquired in different acquisitions. From Kerwin et al.[55] Courtesy of Dr Jerry L. Prince, John Hopkins University.

above, as well as at a few other institutions (e.g. Amini [1]).

In general, there seem to be three different approaches to estimating initial displacement data from magnetic resonance tagging as follows:

- Tagging in multiple intersecting planes and using the tag intersections as tokens for tracking [1, 55, 109].
- Tagging in multiple intersecting planes and then for each tagging plane estimating the magnitude of the motion perpendicular to the plane. This generates a sense of partial displacements (i.e. the component parallel to the tag lines is missing) to be combined later [42, 24].
- Attempting to model the tag fading over time using a model for the Bloch equations and using a variable brightness optical flow approach to extract the displacements [86, 40].

Using intersections: The multiple intersecting planes are either generated by imposing a tag-grid pattern in a single acquisition, which can only be done for two-dimensional grid patterns, or by tagging along different planes in separate acquisitions and superimposing the tagged-planes to create the grid later (see work by Kerwin and Prince [55], Amini [1], Young and Axel [109], etc.) An example of the later approach is shown in figure 2.5, from the work of Kerwin and Prince [55]. The underlying idea here is to try to generate 'material'-markers at the intersection points and then use these as the features for the overall motion-estimation scheme.

Using the whole tag lines: The second approach instead of using just the intersections tries to use the whole of the tag lines (planes). (See work by Haber and Metaxas [42], or Denney and Prince [24].) This has the advantage of being more robust to noise than the first approach, as it uses more of the tag-line and also can provide partial information in regions where there are few intersections. This becomes especially useful in the case of the right ventricle [42], where the thickness of the heart wall is much smaller and the likelihood of having regularly spaced intersections is very low. The penalty paid for this technique is that, at this stage, one can only generate displacement estimates perpendicular to the tag-plane which need to be processed later to generate a full displacement field.

In both of the above approaches, in the pre-processing stage, there is also a need to identify which of the intersections or parts of the tag lines lie within the myocardium and to discard all the others. This results in the need for at least a crude segmentation of the myocardium. The segmentation is commonly done interactively such as in the work of Guttman et al, [41], Young et al, [109] or Kumar et al, [57]. (It is worth noting, however, that Denney [23] proposes a new method which bypasses this segmentation step.)

Both the tag detection step and the pre-segmentation work, in general use methods based on deformable models, following the original work by Kass [54]. (See also the review article by McInerney and Terzopoulos [66].) A deformable model tries to find the curve which minimizes an energy functional which consists of an image based term (typically the gradient) and an internal energy or smoothness term. In the formulation of Kass [54], the snake equation had the form:

$$\int_s |\nabla I(x,y)|^2 + \alpha[(\frac{dx}{ds})^2 + (\frac{dy}{ds})^2] + \beta[(\frac{d^2x}{ds^2})^2 + (\frac{d^2y}{ds^2})^2]ds \tag{2.1}$$

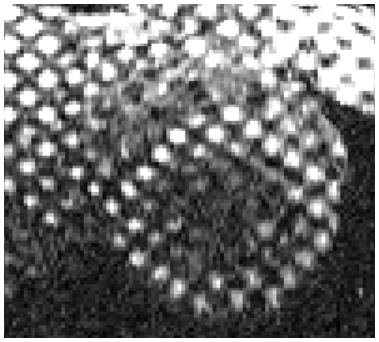


Figure 2.6: An example of a low-frequency tagged MRI image. From Thetokis and Prince[4]. Courtesy of Dr Jerry L. Prince, John Hopkins University.

where $I(x, y)$ is the image as a function of the coordinates x, y, s is the arclength which parameterizes the curve $c(s) = (x(s), y(s))$ and α and β are the smoothing parameters. The gradient term ensures adherence to the image data, whereas the second term tries to keep the curve smooth. This approach is modified to allow for different deformable model geometries, such as grids [57] and for better image adherence terms using some knowledge of the underlying physics such as in the case of Amini [2].

Variable Brightness Optical Flow Methods: In the third case, the whole image is used and data are extracted using a variable brightness optical flow approach on the image intensity. Sinusoidal tagging patterns are primarily used in this case which provide for the smooth intensity fields needed for efficient optical flow computation. See figure 2.6 for an example of this.

The variable brightness part of the algorithm is based on modeling the fading of the tag intensity over time using a model of the imaging process as generated by the Bloch equations [86, 40]. For example, in the work of Gupta [40], the signal (brightness) at time t is modeled as:

$$\psi(t) = D_0 e^{-T_E/T_2}(1 - e^{-T_R/T_1}) + D_0 e^{-T_E/T_2}(\xi - 1)(e^{-t/T1} - e^{T_R - T_1}) \tag{2.2}$$

where D_0 is the proton density, T_1 and T_2 are the relation time constants, T_R is the repetition time, T_E is the echo time, and ξ is the tag modulation coefficient. The first three parameters (D_0, T_1, T_2) are properties of the underlying tissue where as the last three (T_R, T_E, ξ) are the acquisition parameters. In Gupta [40] a composite of the tissue parameters is estimated as part of the displacement estimation algorithm.

As with all intensity based-methods, the original estimates of the displacement field consist of the component of the displacements perpendicular to the isophotes, (this limitation is known as the aperture problem, see Horn [46] for details) which are later regularized to produce a full displacement estimate. The quality of these estimates are highest in the middle of the wall and can be very noisy near the myocardial boundaries. This method has the advantage of not having to detect tags explicitly, but here the brightness variation parameters must be either known or estimated. A rough pre-segmentation of the ventricle is also needed here to avoid smoothing across the boundaries. These methods have, so far, been applied only in 2D.

2.3.2 Methods Relying on Phase Contrast MRI

Several investigators have employed changes in phase due to motion of tissue within a fixed voxel or volume of interest to assist in estimating instantaneous, localized velocities, and ultimately cardiac motion and deformation. While the basic ideas were first suggested by van Dijk [102] and Nayler[72], it was Pelc and his team [82, 83, 81] that first bridged the technique to conventional cine MR imaging and permitted the tracking of myocardial motion throughout the cardiac cycle. This technique basically relies on the fact that a uniform motion of tissue in the presence of a magnetic field gradient produces a change in the MR signal phase that is proportional to velocity. In principle, these instantaneous Eulerian velocities can be derived from each pixel in an image acquisition. An example

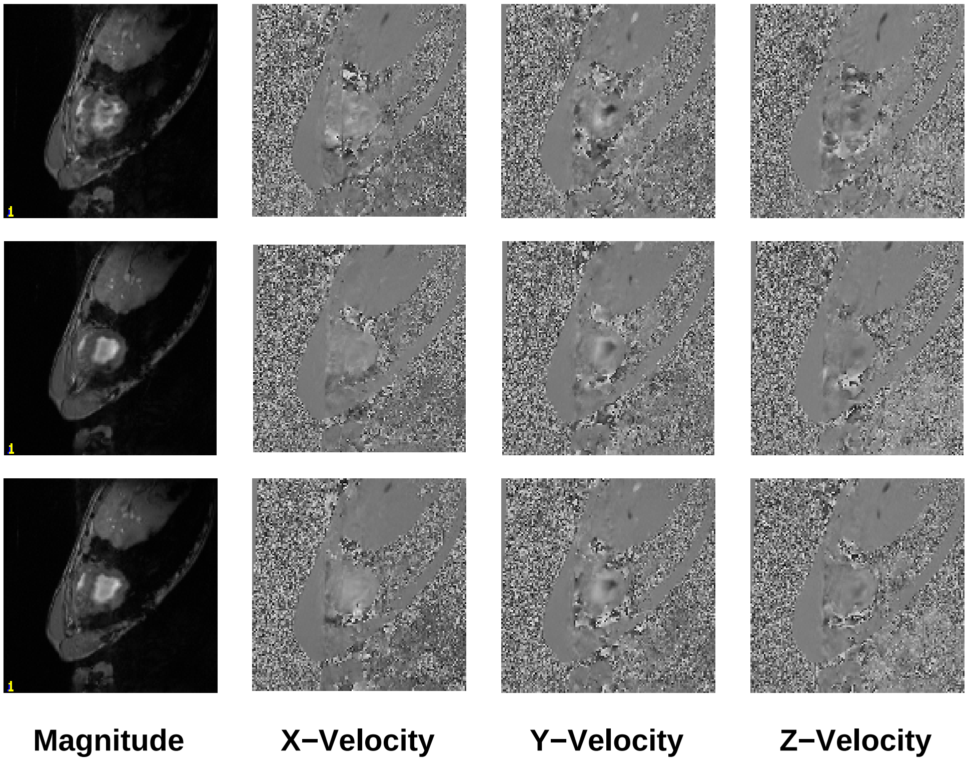


Figure 2.7: Three-slice thick volumetric dataset obtained using magnetic resonance phase contrast images. The left column shows the magnitude images for the three slices and the other columns show the magnitudes of the velocity in the X, Y and Z directions respectively. From Shi et al[91]

of such an acquisition is shown in figure 2.7.

However, clusters of pixels within regions of interest (ROI's) are typically analyzed when predicting point-wise motion, primarily due to signal-to-noise issues. It is worth noting that, as with MR tagging, accurately tracking myocardial motion in 3D requires additional image processing, and little has been reported in the literature about this problem. Assembling the dense field phase velocity information into a complete and accurate 3D myocardial deformation map is currently a limiting problem for this technology. Furthermore, current phase contrast velocity estimates near the endocardial and epicardial boundaries are less accurate. This is due to the fact that the required size of an ROI, for signal-to-noise purposes, is typically large and can include information from outside the myocardial wall. Thus, as with MR tagging, the most accurate LV function information is obtained from the middle of the myocardial wall, and the least accurate information is usually near the endocardial and epicardial wall boundaries. In general there seem to be the following two common approaches to extracting useful information from phase contrast images:

- Processing the data directly to estimate strain rate tensors [105, 82].
- Integrating the velocities over time, via some form of tracking mechanism to estimate displacements [70, 20, 111, 44].

We also note that Shi [91] combined the phase-contrast velocities with shape-based displacements [90] within an integrated framework based on continuum mechanics.

2.3.3 Computer Vision Based Methods

Quantifying the deformation of the LV could be seen as a two-step process: first establishing correspondence between certain points on the LV at time t and time $t + 1$ and second, using these correspondences as a guide, solving for a complete mapping (embedding) of the LV between any two time frames. This problem could be posed for the entire myocardium or just portions of it, such as the endocardial surface alone. There has been considerable effort in general on these two topics, although rarely have they been addressed together.

One common approach to establishing correspondence is to track shape- related features on the LV over time as reported by Duncan [29], Amini[3], Goldgof [53], Ayache [19], McEachen [64] and Shi [90]. The preliminary displacement estimates here are, in general, generated using the following steps:

- First extract the endocardial and epicardial surfaces from the images.
- Then calculate the quantity that is used as the shape feature from these surfaces. These tend to be the curvatures; either the principal curvatures [90] or the Gaussian curvature [53].
- Track points on the surfaces from one frame to the next by minimizing a metric such as bending energy or difference in curvature.

Then the displacement field is smoothed (as was the case with previous methods) to produce the final output displacements. A validation study of shape-based tracking by comparing trajectories with implanted markers was reported by Shi [90], which found that the accuracy of tracking was within the resolution of the image voxel sizes. Another interesting approach by Tagare [99] poses the mapping problem in 2D as a bimorphism between two curves, thus eliminating the basic asymmetry in the tracking process. This has not been extended to 3D yet.

In general all of the methods here depend on an accurate segmentation of the LV walls, but have the advantage of being imaging modality independent. They have been used on MR, CT [90] and 3D ultrasound [78]. The dependency on obtaining an accurate segmentation, however, remains a significant issue, as there still are no fully automated robust and efficient LV surface segmentation methods. The accuracy of the LV segmentation needed for these methods to be successful is obviously greater than in the case of methods using MR Tagging. This is because the surfaces themselves provide the features as opposed to being bounding surfaces within which to search for intersections.

There has been some work done on using the intensity of the images directly to track the LV. Song and Leahy [93] used the intensity in ultrafast CT images to calculate the displacement fields for a beating heart. This is similar in scope to some of the work done with MR tagging (e.g. Gupta [40]) but does not have the advantage of a specially modulated image.

2.4 Modeling used for Interpolation and Smoothing

In general, the initial displacement fields produced by the methods discussed in the previous section have the following characteristics:

- They are *sparse*. Displacements and/or velocities are only available at certain points and not the whole of the myocardium.
- They are *noise-corrupted*. This is an inherent problem in all medical image analysis methods, although the level of noise is very method dependent.
- They may be *partial*. Even where displacements and/or velocities are available, only a certain component of the displacement vector may be known.

The estimation of accurate myocardial deformation requires a dense, smooth and complete displacement field. This is because the deformation is typically captured in terms of the strain which is a function of the derivatives of the displacement field. The process of taking derivatives is very noise-sensitive and this is what makes this problem so challenging as compared to simply estimating the volume of the LV which is an integral measure and hence relatively less sensitive to noise.

The interpolation and smoothing of the displacement field has been attacked in a number of ways. This step essentially constitutes the modeling-step and it is data-independent. The models contain implicitly or explicitly the assumptions made about the displacement field. All of the ‘models’ currently used in this area are passive; they ignore the fact that the heart is an actively contracting organ and not a passive lump of tissue. Some of the modeling strategies are:

- Impose a regularization constraint which penalizes the spatial derivatives, either explicitly [24, 107, 40] combined in some cases with an isochoric constraint[1] [24, 93]. This is further developed in the use of explicit continuum mechanics models, which behave as regularizers [90, 42, 77].
- Model the displacement field by using a smooth spatial parameterization such as affine [70, 73] or splines [55, 1]. This method is used most often when displacement field modeling and tag-extraction are combined in a single step, and is driven by the ease of parameterizing the geometry.
- Use of temporal smoothness or damping [80, 42, 99, 91] and temporal periodicity constraints [64].

In a sense, all of the above methods try to penalize the derivatives of the displacement either in space, or in time, or both. We note that imposing a polynomial distribution such as an affine model is equivalent to setting all derivatives higher than a certain order to zero. This is a limiting case of penalizing spatial derivatives and will be explored in more detail in chapter 6.

Spatial Smoothness Constraints: The application of spatial smoothness constraints relies on the intuition that given that the myocardium is a single object, its displacement field can be expected to be smooth. If this is violated then the tissue would tear apart. Therefore, high values of derivatives in the displacement field (or equivalently high frequency components of its Fourier Transform in the spatial sense) are likely to be the result of noise. This results in methods that penalize the spatial derivatives as in the optical flow method proposed by Horn and Schunk [47]. In this case the optimal displacement field is found as a trade-off between satisfying the *gradient constraint equation* and a

[1] The myocardium is considered to be nearly incompressible and the isochoric constraint tries to enforce this incompressibility.

regularization term as follows:

$$\hat{u} = \begin{matrix}\arg\min \\ u\end{matrix} \int_{\mathbf{x}} (\frac{dI}{dt} + u.\nabla I)^2 + \lambda(\sum_{ij}(\frac{du_i}{dx_j})^2)d\mathbf{x} \tag{2.3}$$

where the u is the displacement vector field over a space x which can be two or three-dimensional, t is time and I represents the image.

The gradient constraint term $(I_t + u.\nabla I)^2$ essentially tries to match points of equal intensity and is the *data* term, whereas the sum of squared derivatives multiplied by the smoothness factor λ constitutes the regularizing term. The regularizing term can be thought of as a *model* term as it contains *no* image related information. It captures the authors' prior belief in the properties of the displacement field.

This framework is used in many of the approaches described earlier, although it is adapted to either match the data or the prior information. For example, in the case of the variable brightness optical flow method [40, 86], the gradient constraint term is replaced by a different measure which allows for the fading in the tag pattern. In a more general case, the gradient constraint term can be replaced by an image-data adherence term. This term tries to find a displacement field which stays close to some pre-existing displacement estimates obtained using approaches described in section 2.3. For example if an estimate u^m of the displacement field exists, we could modify the Horn and Schunk framework as follows:

$$\hat{u} = \begin{matrix}\arg\min \\ u\end{matrix} \int_{\mathbf{x}} |(u - u^m)|^2 + \lambda(\sum_{ij}(\frac{du_i}{dx_j})^2)d\mathbf{x} \tag{2.4}$$

We can expand on this model by also using an isochoric constraint which tries to penalize volume changes, as was done in Denney [24] and Song [93]. This takes the form $(\nabla.u)^2$ and is motivated by the fact that the myocardium, like most soft tissue, is thought to be approximately incompressible[2]. Alternatives also include the use of thin-plate spline energy terms [55] or B-spline terms [1].

The combination of the smoothness and isochoric terms describes the myocardium in terms of what is essentially an internal energy function. Continuum mechanics models of the myocardium as found in the biomechanics literature [51] are also described as internal energy functions, which also essentially penalize derivatives. So it is a natural step at this point to try to bridge some of this knowledge into the inverse problem of motion estimation. To do this, the regularization term is replaced by an explicit mechanical model, which is in most cases an isotropic linear elastic model[91, 80, 42]. A transversely isotropic elastic model is used by Papademetris [78]. This allows the model to account for the preferential stiffness of the myocardium along the fiber directions. It is interesting to note that, from continuum mechanics theory [62], an internal energy function can describe a real material if and only if it is invariant to rigid translation and rotation, otherwise this material violates the 2nd law of thermodynamics. It can be shown that the classical model of Horn and Schunk is *not* invariant to rotation and would fail this criterion.[3]

If we discretize equation (2.4), differentiate it with respect to u, and concatenate all the individual displacements u into a large vector U we can write the generalized expression:

$$[K]U = F \tag{2.5}$$

where K is the assembled matrix of local derivative operators (as in Kass [54]) and is sparse. This contains the model constraints which can be derived either from a regularization term or an explicit continuum mechanics model. F is the external driving force which tries to deform the model to adhere to the image data. This equation is most easily

[2] There is in fact some change in volume, due to blood flow (reperfusion) into the wall, but this is considered to be small.

[3] We will discuss this in more detail in section 5.2.

solved using the finite element method [9] in cases of complex geometry and especially in three dimensions.

Temporal Smoothness Constraints: There are two types of temporal smoothness constraints in the literature. In the first case, we have an explicit temporal filtering scheme applied to individual displacements. This is primarily, but not exclusively, done in the case where the input data is derived from phase contrast velocity. In the work of Meyer [70], a Kalman-filtering approach is used to smooth the displacement field. Zhu [111] and McEachen [64] parameterize the problem in the frequency domain by expanding the displacement of an individual point over time in terms of Fourier series and try to take advantage of the periodicity of the left-ventricular motion.

The second case involves extending equation (2.5) to include dynamics. This results in the following generalized expression:

$$M\ddot{U} + C\dot{U} + KU = F \tag{2.6}$$

where M is a mass matrix and C is a damping matrix. This approach also results in a form of temporal smoothing, which is motivated by similar approaches in continuum mechanics. In the work of Park [80], this was reduced to $C\dot{\mathbf{U}} = F$ by ignoring the mass matrix and setting the stiffness to 0. In Haber [42] the stiffness term is also preserved. The full dynamical model is employed in Shi [91]. In this case both shape-based displacements and phase-contrast velocity information are used. The full dynamical model is also used in work done in the computer vision and graphics communities by Metaxas and Terzopoulos [101].

We also note that Pentland [48] and Nastar [71] use this approach and by ignoring the damping term, reduce it to a modal finite element equation, which parameterizes the deformation in terms of the eigenmodes of the stiffness matrix K. In both of these approaches, however, there is no explicit notion of correspondence between material points and the displacements are found using a global distance measure.

2.5 Validation of Results

The validation of LV deformation results is an extremely important and often neglected aspect of work in this area. In general, we need to address the following questions:

- Does the imaging modality produce an accurate picture of the underlying geometry and/or displacement and velocity?
- Does the analysis algorithm extract these data accurately and reliably?
- Are the results meaningful for clinical and/or physiological purposes? Do they discriminate between healthy/dysfunctional regions?

In general, the first two questions are difficult to address *in vivo*. Often phantoms are used with known shapes and displacements, so there is ground truth information to compare any measurements with (e.g. Kraitchman [56] and Constable [20]). An example of this is shown in figure 2.8. In Young [108] it was shown that away from the free surfaces of the gel-phantom, a Rivlin-Mooney [62] analytic model accurately reproduced the 2-D displacement of magnetic tags. This showed agreement between the theory (model) and the image-derived displacements. However, the real *in vivo* measurement of the beating heart usually presents additional complexities which introduce problems not typically accounted for in phantoms, such as full and complex 3D motion and fast blood flow through the ventricle. These can generate artifacts in the images and cause significant distortions.

The second question has been attacked in approaches based on MR tagging (e.g. Amini[1] Prince [86] and Haber [42]) using simulations. One example shown in figure 2.9 uses a kinematic model of the left ventricular motion by Arts [5] within an MR tag image simulator [103] to generate synthetic images with known displacements. Comparison with manual extraction has often been used as the gold standard to validate the process of tag-extraction, as in Kraitchman [56].

In the shape-tracking work of Shi [90], implanted markers are used as the gold standard. These markers are physically implanted in the myocardium before the imaging.

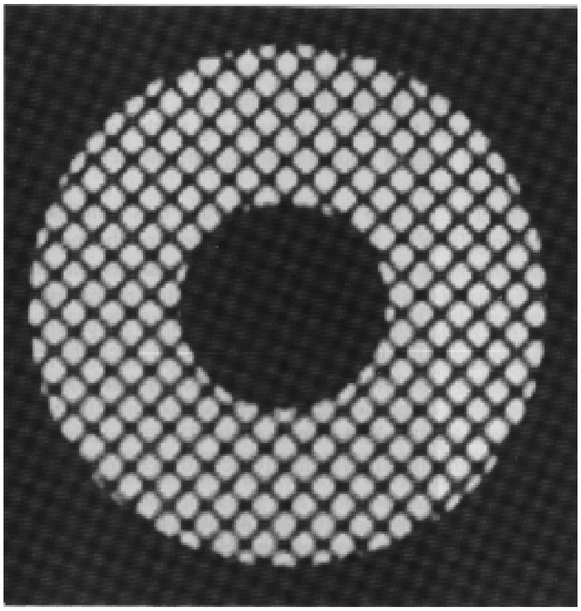


Figure 2.8: MR image of gel phantom with SPAMM (tag) stripes in undeformed state. (From Kraitchman [56]) Courtesy of Dr Leon Axel, University of Pennsylvania.

An MR image of a heart with implanted markers is shown in figure 2.10. This approach to validation tries to attack the first two questions simultaneously. Here, algorithm generated displacements are compared to the marker-displacements (these are easily identifiable from the images). This technique has the disadvantage of comparing trajectories in a smaller number of points, however, it is done on *real* data as opposed to simulations.

The third question is not addressed much in the image analysis literature, quantitatively. Often an example of the results on a normal and a hypertrophic heart is shown and the differences 'correlated' with other evidence from the cardiology literature. It is known from the literature (e.g. Croisille et al, [21]) that on average the changes between normal and abnormal regions in terms of radial and circumferential strains is on the order of $10-15\%$, and much smaller in the case of borderline regions. A quick calculation shows that, in the case of MR tagging based work where the tags are typically 5 voxels apart at end-diastole, the change in the spacing at end-systole is going to be around 0.5 voxels

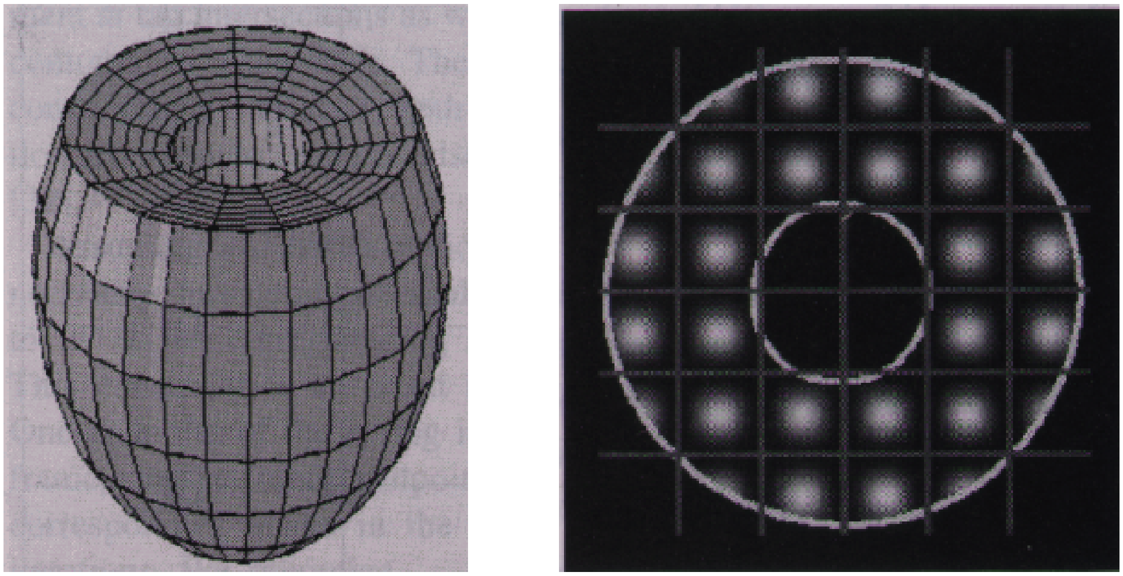


Figure 2.9: Example of the use of the cardiac simulator [5, 103] used to validate methods based on MR tagging. Left: the undeformed prolate spheroidal model of the LV in the reference state. Right: a tagged image corresponding to a selected image plane. (From Amini[1]) Courtesy of Dr Amir A. Amini, University of Washington, St Louis.

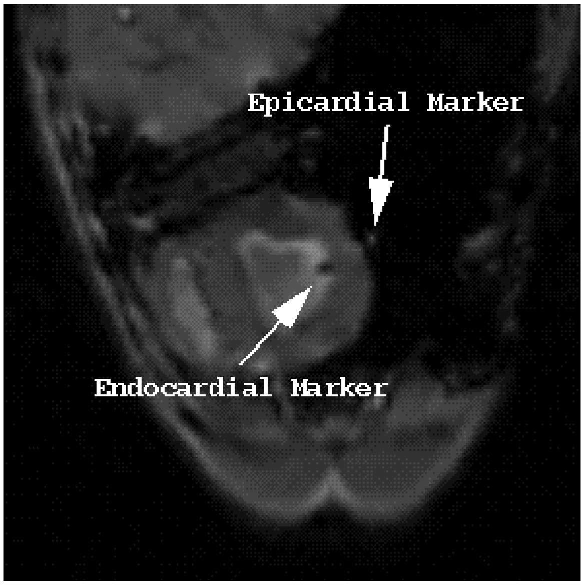


Figure 2.10: 2D MR image slice of left ventricle with implanted markers. These are used to validate shape-based displacement estimates. (From Shi [90])

or less. In the case of shape-based methods where the whole of the ventricle is used, this number is somewhat larger (around 0.8 voxels). If such changes are to be detected reliably, and we were to ignore accumulated tracking errors after the tags and/or boundaries have been extracted, we need to be able to extract tags/boundaries at a precision of 0.25–0.4 of a voxel or less. This is currently beyond the performance level of all automatic algorithms on real data; hence manual and semi-automatic algorithms are used in most cases.

In Croisille [21], the reported results are averaged over a number of studies to reduce the effects of errors in detecting individual tag lines and variations among different subjects. This may be useful for exploring the physiology but not plausible in the case of diagnosis, unless the results are averaged over large sections of the ventricle to reduce noise.

2.6 Conclusions and Further Research Directions

The major problem/bottleneck in most of the work presented in this chapter is the extraction of features such as tag lines and especially left ventricular surfaces from the image data. As mentioned in the previous section, there is a reliance on manual and semi-automatic techniques to obtain this information. Another problem, which is less an issue of image analysis and more an issue of medical imaging technology, is the difficulty of using magnetic resonance in a clinical setting. It is not possible to image patients in an emergency room (as is the case for example with ultrasound) and metallic objects such as pacemakers cause serious problems and dangers when placed in the magnet.

As mentioned earlier, most of the models used to smooth and/or interpolate the displacement field are passive; they do not contain any active contraction information. This can result in an underestimation of the deformation, as the model biases the results towards no change. This was noted in the work of Park [80] and is the reason why no spatial smoothness was employed there. This, however, is not a sufficient solution to the problem as some spatial smoothing is often needed to cope with the noise in the data and the sparseness in the image information. A possibly better solution would be to incorporate some knowledge of the active contraction of the left ventricle during the first half of the cardiac cycle. This has the potential of eliminating the bias problem, although it would introduce more parameters to be set or ideally estimated from the image data. We explore this problem in more detail in chapter 6.

Magnetic resonance imaging represents a promising modality and the development of improved analysis techniques will enhance the possibilities of it being used clinically. In the meantime we note that improvements in 3D echocardiography technology, such as the introduction of harmonic imaging [13] and contrast agents [84], are beginning to make this modality an attractive and somewhat cheaper alternative. We have already reported preliminary work in this area [78]. A more detailed exposition can be found in chapter 7.

Chapter 3

An Interactive Approach to Left Ventricular Segmentation

3.1 Introduction

In this chapter, we present the methodology used to extract the bounding surfaces of the left-ventricular myocardium from an image sequence. These surfaces are used as inputs to the mesh generation and shape-based tracking methods, which will be described in sections 4.3 and 4.4.

For the accurate estimation of cardiac deformation, the accuracy required is above what automated algorithms can currently achieve. We therefore used a semi-automated approach which allows for both user interaction and correction. Recently some interesting work in the area of interactive segmentation has appeared in the literature [60, 50]. To satisfy the need for user interaction at all stages of the segmentation process, we take a slice-by-slice approach to 3D segmentation. In this way the surface is extracted in a 2D fashion one contour at a time (a contour representing the intersection of the surface with the 2D image slice) and reconstructed using shape-based interpolation (see section 4.1.1) and Delaunay triangulation (see section 4.2.1).

Two-dimensional contour extraction in Biomedical image analysis has often been done using deformable models or snakes. These were first introduced by Kass et al[54]. A review article by McInerney and Terzopoulos [66] describes the use of deformable models in more detail. We also note the alternative level-set approach [87, 110] which instead considers the contour to be the zero-level set of a three dimensional function and tries to evolve this function to solve the segmentation problem. The level-set approach is not well suited for easy user interaction[1] so in this work we use a snake-based deformable model approach.

We further represent the two-dimensional contours as B-splines [22]. The choice of B-splines was determined by two major factors (1) the ease of parameterization of a curve with excellent smoothness and continuity properties and (2) the ease of user interaction for editing curves before and after the automated segmentation stage. Also splines are available in the Open-Inventor toolbox[106] used for the visualization part of the segmentation.

Ease of interaction was was probably the principal reason for the use of B-splines as opposed to the Fourier parameterization employed by Staib [95] and Chakraborty [15].

3.2 Parameterizing Closed Curves Using B-splines

In this section we describe how closed curves can be parameterized using B-splines. We start by defining the terms *normalized arclength*, *knots*, *knot points* and *control points*. Next we describe the definition of the B-spline itself. Finally we put the two together to parameterize a closed curve using B-splines.

Definitions We will define a two-dimensional curve as:

$$c(s) = (x(s), y(s)) \ , \qquad s \in [0.0, 1.0) \tag{3.1}$$

[1] In the case of level-sets the definition of the curve is implicit. This makes it is harder to come with an easy way to interactively edit the curve. One way might be to first extract the zero-level set, parameterize it using splines, edit this, and then form the level-set function again. While this is doable, it is also cumbersome.

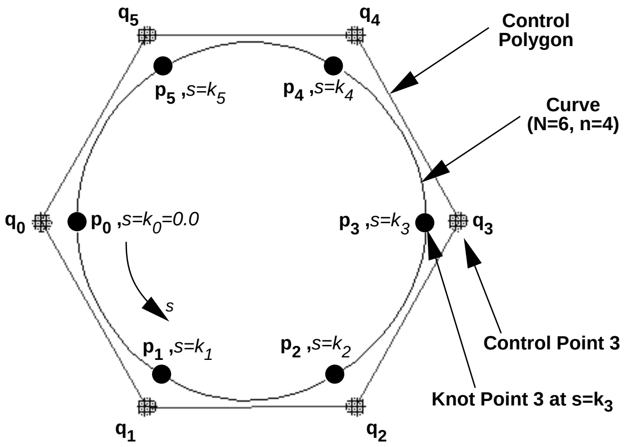


Figure 3.1: The elements of a B-spline. This curve is parameterized using six (N=6) cubic (n=4) B-splines. The parameterization is defined N, n the location of the control points (q_i) and the knot sequence (k_i). The curve is divided into N non-overlapping segments. Adjacent segments are joined together at the knot points p_i. The curve has continuity $\mathcal{C}^2$ at the knot points, and $\mathcal{C}^\infty$ elsewhere.

where s is the *normalized* arclength, that is the arclength divided by the total circumference of the curve. Each curve is divided into N non-overlapping segments. We define the *knot vector* k, to be the concatenation of the normalized arclengths of the points where the adjacent segments of the curve are joined. For example segment 0 and segment 1 are joined at the point $s = k_1$.[2] The point $p_i = c(k_i)$ is called a *knot point*. These definitions are illustrated in figure 3.1.

We further note that the knot vector k has size N. For later notational convenience we define a (recursive) periodic extension to k as:

$$\begin{aligned} k_i = &\quad k_i \quad && 0 \le i < N \\ = &\quad k_{i+N} \quad && i < 0 \end{aligned}$$

[2] Hence the use of the term knot, a place where two different things are joined together.

$$= \quad k_{i-N} \quad i \geq N$$

B-splines Here we follow the notation of Lancaster[58, section 4.4] (see also deBoor[22].)

Definition: Let k_i where $i = -3, -2, \ldots, N+3$ be knots satisfying $k_m < k_{m+1}, m \in (-3, N+3)$. A one-dimensional B-spline of order n, n=1,2,3,4 with these knots is a piecewise (n-1)th degree polynomial not identically zero of continuity class $C^{(n-2)}$ in the region $[k_{-3} \leq x < k_{N+3}]$ and of minimal support. When $n = 1$ we interpret the class C^{-1} as admitting functions with discontinuities at the knots k_i.

B-splines of orders 1 to 4 are shown in figure 3.2(left). Note that a spline is defined as $B_{i,n}$ where i defines the start of the region of support of the B-spline in terms of the knot sequence k_i and n defines the order of the spline. All of the splines in figure 3.2(left) start at $i = 1$ hence are all $B_{1,n}$. A function is approximated as a sum of different splines as also illustrated in figure 3.2(right).

We compute the value of a B-spline recursively as follows:

$$B_{i,n} = \frac{s - k_1}{k_{i+n-1} - k_i} B_{i,n-1}(s) + \frac{k_{i+n} - s}{k_{i+n} - k_{i+1}} B_{i+1,n-1}(s) \tag{3.2}$$

$$B_{i,1} = \begin{cases} 1 & k_i \leq s < k_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

where $i = 0, \ldots, N-1$ and $n = 1, 2, 3, 4$..
B-splines calculated in this way also have the additional property that:

$$\sum_{i=0}^{N-1} B_{i,n}(s) = 1 \tag{3.4}$$

Using the above definitions we represent a function $f(s)$ as a weighted sum of order n

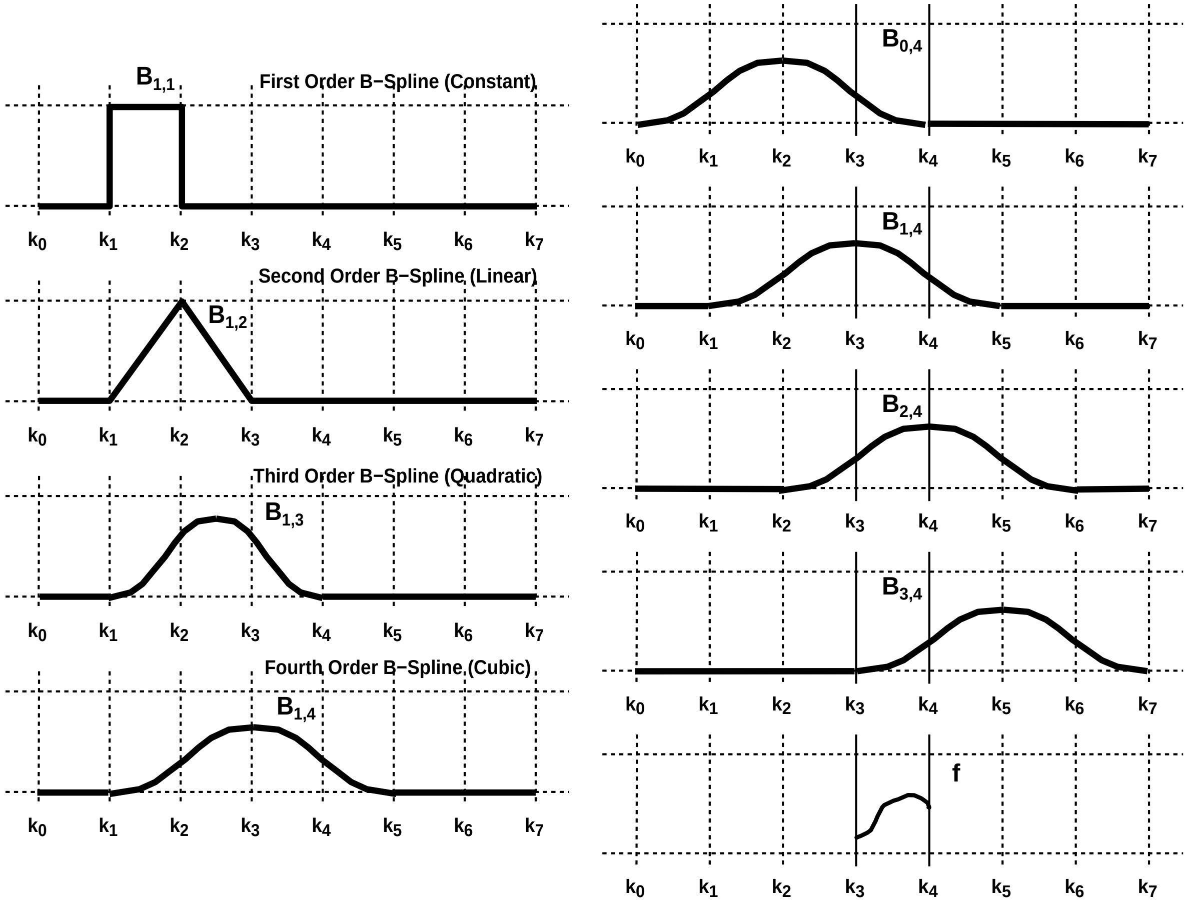

First Order B−Spline (Constant)
B1,1
Second Order B−Spline (Linear)
B1,2
Third Order B−Spline (Quadratic)
B1,3
Fourth Order B−Spline (Cubic)
B1,4
B0,4
B1,4
B2,4
B3,4
f
k0 k1 k2 k3 k4 k5 k6 k7

Figure 3.2: Approximation of a function with B-splines. Left: B-splines of orders 1 to 4. Note (i) the limited support of each spline and (ii) that the order of the polynomial describing each spline is one less than the order of the spline. Right: Representation of a function f as a set of cubic (fourth-order) B-splines. We can represent function f in the region $[k_3, k_4)$ as a linear combination of the B-Splines $B_{0,4}, B_{1,4}, B_{2,4}$ and $B_{3,4}$, e.g. $f(s) = \sum_{i=0}^{3} q_i B_{i,4}$, where q_i are the appropriate weights.

B-splines as:

$$f(s) = \sum_{i=0}^{N-1} q_i B_{i,n}(s) \tag{3.5}$$

where q_i are the appropriate weights. We further note that we can write the derivatives of $f(s)$ as linear combinations of the derivatives of $B_{i,n}(s)$ as:

$$\frac{d^r f(s)}{ds^r} = \sum_{i=0}^{N-1} q_i \frac{d^r B_{i,n}(s)}{ds^r} \tag{3.6}$$

A two-dimensional curve $c(s) = (x(s), y(s))$ is parameterized as:

$$x(s) = \sum_{i=0}^{N-1} q_{x,i} B_{i,n}(s) \quad , \quad y(s) = \sum_{i=0}^{N-1} q_{y,i} B_{i,n}(s) \tag{3.7}$$

So the full representation of the curve $c(s)$ consists of a set of knots k_i, a corresponding set of weights $q_{x,i}, q_{y,i}$ and the choice of the order of the B-spline n (see figure 3.1). We describe the selection of these, next, in reverse order:

Choosing the order of the B-splines: In this work we only use fourth-order ($n = 4$, cubic) B-splines. This ensures at least C^2 connectivity over all the curve which allows us to compute the second partial derivatives needed in the segmentation work (see section 3.3).

Selecting the knots: There are two common choices for setting of the values of the knots k_i. The first is the so-called *uniform parameterization* which sets $k_i = \frac{i}{N}$. A better choice is the *chord length parameterization* which sets $k_{i+1} = \frac{k_i + |p_{i+1} - p_i|}{L}$, where L is the total length of the curve. The chord length parameterization has the advantage of allowing the placement of more knots in regions of high curvature. There still is, however, no firm concept of an optimal knot spacing [33].

Selecting the weights via a control polygon: Often the pair $(q_{x,i}, q_{y,i})$ is given a geometrical interpretation as the coordinates of the equivalent control point for knot point i. These control points are often linked together to form the so called 'control-polygon' for the spline as shown in figure 3.1. Next we define the following vectors which consist of the x and y coordinates of the knot points and the control points respectively:

$$\begin{aligned} p_x &= [x(k_0), x(k_1), \ldots, x(k_{N-1})]' \quad , \qquad q_x = [q_{x,0}, q_{x,1}, \ldots, q_{x,N-1}]' \\ p_y &= [y(k_0), y(k_1), \ldots, y(k_{N-1})]' \quad , \qquad q_y = [q_{y,0}, q_{y,1}, \ldots, q_{y,N-1}]' \end{aligned}$$

It can be shown that $p_x = [W]q_x$ and $p_y = [W]q_y$ where W is an $N \times N$ matrix.[3] We can use this relationship to generate a set of control points from a set of knot points specified by the user in some fashion. This is also exploited in the interactive segmentation part of this work. The user may adjust the knot points (p_x, p_y) which are on the curve and the control points (q_x, q_y) can be computed using a simple matrix multiplication. The matrix W only needs to be inverted once at the start of the process. It is also worth pointing out that for cubic B-splines W is a circulant pentadiagonal matrix and can be inverted using sparse matrix methods [85].

Alternatively the control points q_x, q_y and the knots k_i can be generated by performing a least squares fit to a set of ordered points. Algorithms exist which will automatically select the number of knots as well as the placement of control points and knots given a smoothness criterion. For more information see Dierckx[25, chapters 3-6].[4]

[3] This is easy to see. The position of any point on the curve $v(s) = (x(s), y(s))$. We can write $x(s) = \sum_{i=0}^{N-1} B_i(s) q_{x,i}$ and similarly $y(s) = \sum_{i=0}^{N-1} B_i(s) q_{y,i}$. For the r-th element of p_x and p_y, $s = k_r$ and $B_i(s) = B_i(k_r)$ is a constant. So the r-th element of p_x (and similarly the r-th element of p_y) can be written as a linear combination of the control point coordinates $q_{x,i}$ weighted by the constants $B_i(k_r)$. We collect the values of these constants into the $N \times N$ matrix W.

[4] When using standard packages for the implementation of B-splines such as FITPACK [25] or Open In-

3.3 A B-spline Snake Implementation

A snake is a controlled continuity spline deforming under the influence of image forces. The deformation tries to minimize an energy functional of the form:

$$E = \int_0^1 E_{int}(c(s)) + E_{ext}(c(s))ds \tag{3.8}$$

where E_{int} is the internal energy function which tries to preserve the smoothness of the curve and E_{ext} is the external or potential energy term which tries to attract the curve towards desirable image features. Typically, E_{int} consists of squared derivatives of $c(s)$ with respect to s. E_{ext} is usually defined as the negative of the magnitude of the image gradient.

3.3.1 The Internal Energy Functional

In our implementation we set the internal energy to be equal to:

$$E_{int}(c(s)) = \Big(\frac{\partial^2 x(s)}{\partial s^2}\Big)^2 + \Big(\frac{\partial^2 y(s)}{\partial s^2}\Big)^2 \tag{3.9}$$

We will discuss the external image function in section 3.3.2.

ventor [106] one can only specify knot spacing and control points for open curves. Closed curves can be generated by using the following trick. Consider a curve parameterized using N cubic ($n = 4, N \geq 4$) B-splines with control points $q = [q_0, \ldots, q_{N-1}]$ and knot vector $k = [k_0, \ldots, k_{N-1}]$. Both FITPACK and Open Inventor will require this closed curve to be converted into open-curve notation. This is achieved by padding the knot vector as: $k = [\mathbf{1 - k_{N-1}}, \mathbf{1 - k_{N-2}}, \mathbf{1 - k_{N-3}}, k_0, \ldots, k_{N-1}, \mathbf{1 + k_0}, \mathbf{1 + k_1}, \mathbf{1 + k_2}, \mathbf{1 + k_3}]$. and setting the control point vector to have the form $q = [\mathbf{q_{N-1}}, q_0, \ldots, q_{N-1}, \mathbf{q_0}, \mathbf{q_1}]$. We note here number of extra elements in the vectors k and q (shown in bold-print) is independent of the number of control points N and is solely a function of the order of the B-splines n. These adjustments generate the equivalent open curve for use in algorithms which do not assume closed curves.

The snake $c(s)$ is parameterized using B-splines. This enables us to construct a straightforward numerical algorithm to find the optimal $c(s)$.[5]

We use fourth-order or cubic B-splines. This is the lowest order which ensures C^2 continuity throughout the curve. This enables us to calculate the E_{int} term as defined in equation (3.9). The knots k_i are also kept fixed during the iterations of the snake. Hence the change in the position of the snake is solely a function of the location of the control-points (q_x, q_y).

3.3.2 The External Energy Functional

.

The external energy functional (the E_{ext} term of equation (3.8)) defines the type of feature which we would like the snake to be attracted to. One common form of this is

$$E_{ext}(s) = \int_0^1 -|\nabla I(c(s)|^2 ds \tag{3.10}$$

which tries to attract the snake towards maxima in the local image gradient. This is the most common energy function when one is trying to detect relatively clean boundaries, such as is the case in MR images.

Chakraborty et al [14] demonstrate the improvement that can be obtained by using also an intensity homogeneity constraint for the interior of the contour. This approach leads to a generalized form for $E_{ext}(s)$ as follows:

$$E_{ext}(s) = \sum_{m=1}^{M} \alpha_m \int_0^1 E_m(c(s))ds \tag{3.11}$$

where now the external energy is dependent on M different modules E_m weighted by

[5] The original implementation for uniformly parameterized contours was by Hemant Tagare [no reference available] who generously allowed the author access to his source code. This was subsequently extended for cases where the parameterization was not uniform.

their relative confidence α_m.

In this work we use a combination of the following three modules; the gradient module where as above $E_1(c(s)) = -|\nabla I(c(s))|^2$, $E_2(c(s))$ which is derived from a texture module and $E_3(c(s))$ which is derived from a prior curve $v_p(s)$.

The texture module: This is a classification scheme where each pixel in the image is assigned to a texture class c. This work approach described in detail by Chakraborty [15, 14, Section 3.3] and is derived from the work of Manjunath[63]. We follow here the presentation given in Chakraborty [14] and model the intensity image as a Gaussian Markov random field (GMRF). This models the conditional probability of the image intensity given the classification.

Let S denote the $M \times M$ image lattice, i.e. $S = \{(i,j), 1 \leq i,j \leq M\}$. Let $\{L_s, s \in S\}$ and $\{Y_s, s \in S\}$ denote the labels and the zero mean array obtained from the image data respectively. Note that the labels can belong to only a certain number of texture classes (typically 2 or 3). Let N_s denote the symmetric second order neighborhood of a site s consisting of the eight nearest neighbors. Now, assuming that all the nearest neighbors of s also have the same label as s, we can write the following expression for the conditional density of the intensity at the pixel site s[63]:

$$P(Y_s = y_s | Y_r = y_r, r \in N_s, L_s = l) \quad = \quad \frac{exp(-U(Y_s = y_s | Y_r = y_r, r \in N_s, L_s = l))}{Z(l | y_r, r \in N_s)} \tag{3.12}$$

where $Z(l|y_r, r \in N_s)$ is the partition function of the conditional Gibbs distribution, and

$$U(Y_s = y_s | Y_r = y_r, r \in N_s, L_s = l) = \frac{1}{2\sigma_l^2}\left(y_s^2 - 2\sum_{r \in N_s} \Theta_{s,r}^l y_s y_r\right) \tag{3.13}$$

In (3.13), σ_l and Θ^l are the GMRF model parameters of the l^{th} texture class. Further, the model parameters satisfy: $\Theta^l_{s,r} = \Theta^l_{s-r} = \Theta^l_{r-s} = \Theta^l_r$. These parameters for each region are estimated by a least squares estimate method using a window around a user specified point, representative of that particular region.

Once the intensity image Y^* has been modeled, the next task is to determine the classification. This is achieved by maximizing the posterior distribution of the texture labels given the intensity image:

$$P(L|Y^*) = \frac{P(Y^*|L)P(L)}{P(Y^*)} \tag{3.14}$$

where L corresponds to the classified image with L_s describing the label at the s^{th} pixel. The label field L is modeled as a second order MRF, which says that $P(L_s|L_{S/s}) = P(L_s|L_r, r \in \hat{N}_s)$ where $L_{S/s}$ is the whole label field excluding the site s. It acts as a prior that emphasizes the property that neighboring pixels of the classified image share the same label (see Leahy [59] for details). Maximizing (3.14) gives an optimal Bayes estimate. We maximize (3.14) using the coordinate-wise descent method of Leahy [59], similar to the iterated conditional mode (ICM) algorithm [10, 11].

Once the classification L has been obtained we would like to attract the curve to locations where there is a *texture boundary.* So we create an energy function $E_2(c(s)) = -|\nabla L(c(s))|^2$ to be included in equation 3.11.

The prior module: The third term in equation (3.11) comes from a prior curve. For the purpose of generating $E_3(c(s))$ we represent the prior curve as its distance map (generated using the chamfer method of section 4.1.1. If this prior curve was to be derived from a number of curves, it could simply generate as the zero set of the (possibly weighted) means of the distance functions of these curves.

In this particular work the prior curve is used to enforce a temporal constraint on the segmentation. Consider a time frame t where we are trying to estimate curve $c(t)$.

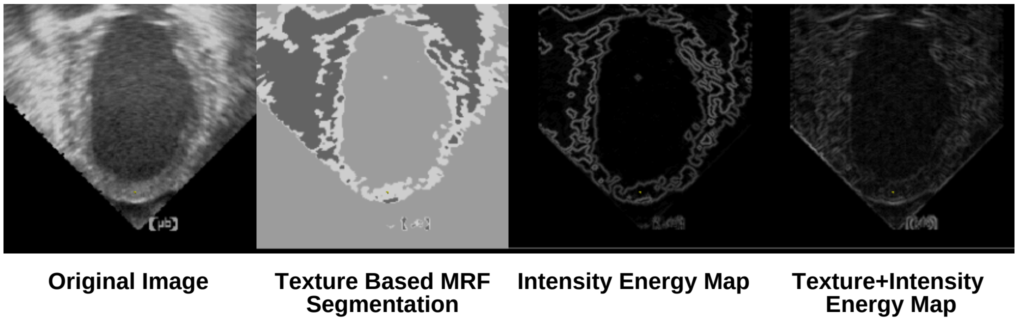


Figure 3.3: External Energy Functions for intensity and intensity+texture snakes. Note that the intensity only energy function is very noisy inside the left-ventricular blood-pool which creates many local minima for the deformable contour. The use of the texture eliminates most of these minima.

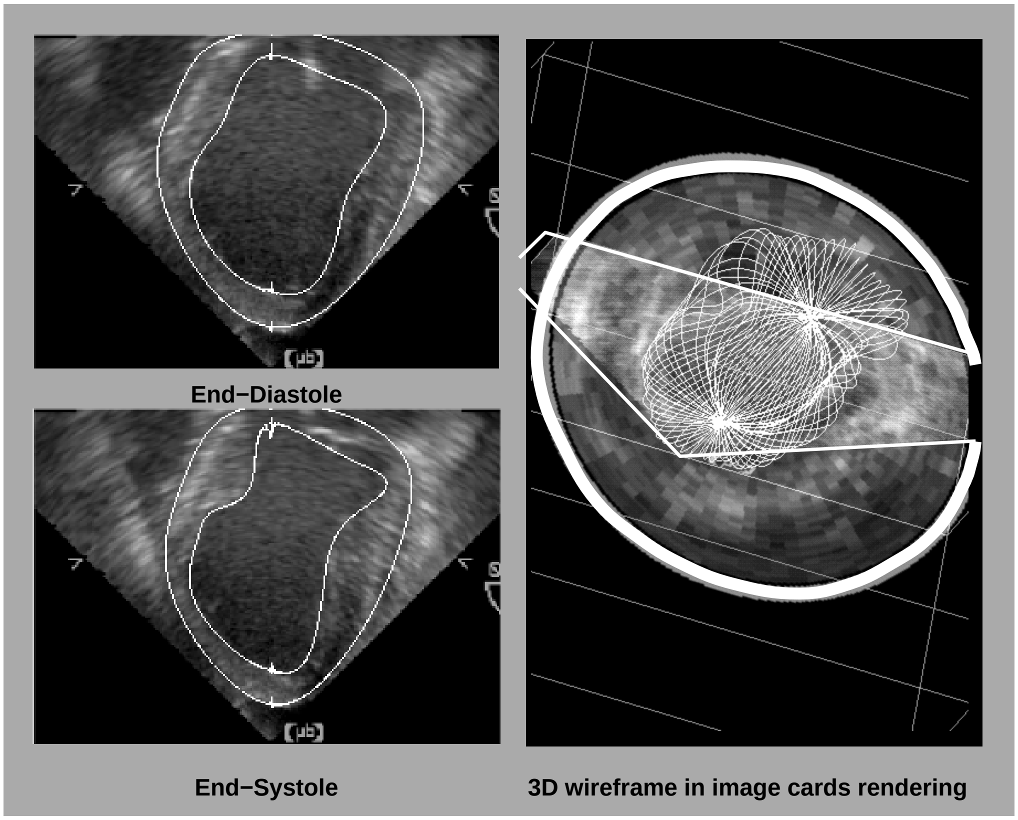


Figure 3.4: Left: Images and superimposed extracted contours. Only two of the eight frames are shown. Right: 3D rendering showing all the wire-frame contours superimposed on a long axis (original) and a short-axis (interpolated) image slices.

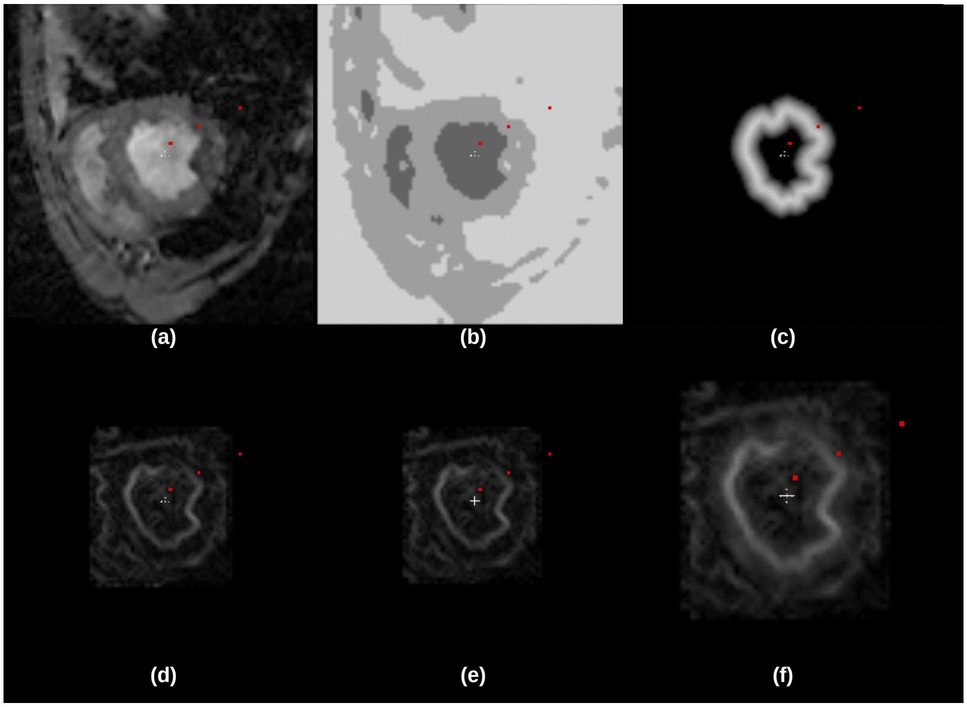


Figure 3.5: This figure illustrates the use of multiple external energy functions. (a) shows the original image, (b) the texture segmentation, (c) the temporal smoothness term, (d) the external energy function using the gradient alone, (e) gradient+texture external energy function and (f) enlarged gradient+ texture+ temporal smoothness external energy function.

We can generate a prior curve for the segmentation as follows:

$$d_m(c_p(t)) = 0.5\beta(d_m(c(t-1)) + d_m(c(t+1))) + (1.0 - \beta)d_m(c_0(t)) \tag{3.15}$$

where $d_m()$ is the distance map of a curve as defined in section 4.1.1. The curves $c(t-1), c(t+1)$ represent the current estimates of the same contour in the previous and next time frames and $c_0(t)$ represents the last estimate of this curve. The factor β is the damping factor. All these are used to generate an estimate for the current curve $c_p(t)$.

Given $d_m(c_p(t))$ (there is no need to explicitly extract $c_p(t)$) we can generate an external energy term $E_3(c(s))$ as follows:

$$E_3(c(s)) = -|d_m(c_p(t))| \tag{3.16}$$

which tries to constrain $c(s)$ to stay close to $c_p(t)$. In a similar way we could impose a known expected thickness constraint such as the one in Zeng [110] to keep a curve within a certain distance from another curve. In that case $E_3(c(s))$ would take the form:

$$E_3(c(s)) = \begin{cases} 0 & |d_m(c_p(t))| < t \\ -1 & \text{otherwise} \end{cases} \tag{3.17}$$

where t is the pre-specified thickness. Note that while both in this definition and also in Zeng[110] there is no explicit correspondence between the two curves/surfaces, an 'asymmetric nearest neighbor' correspondence is implicitly used[6]. This is because at each point p on the curve/surface the value of $d_m(c_p(t))$ is the distance between p and its nearest neighbor on the prior curve c_p.

Minimization of Energy Functional: Having defined the terms of the energy functional of equation (3.8) we describe here the procedure used to obtain the final curve. First given the external energy function $E_{ext}(x, y)$ defined over the image plane we calculate its derivatives with respect to x and y, $\frac{\partial E_{ext}}{\partial x}$ and $\frac{\partial E_{ext}}{\partial y}$. These derivatives are the driving terms for the deformation of the snake. Further we note that the coordinates of control point i, $(q_{i,x}, q_{i,y})$ are the weights for the B-spline B_i (we use B_i to abbreviate $B_{i,4}$ as the order of the B-splines from here on is assumed to be 4). We can write the

[6] This approach runs into problems when the two curves are locally not parallel as whole regions of one curve map to a single point on the other curve. Also, whole regions on the second curve may not contribute to this map resulting in 'cutting corners'. We will discuss this problem in greater detail in section 4.1.2.

energy function E as a sum of N parts each relating to a B-spline part of the snake B_i as:

$$E = \sum_{i=1}^{N} E_i \tag{3.18}$$

E_i has a region of support from $k_i \leq s < k_{i+4}$. The individual elements[7] E_i are defined as:

$$E_i(s) = \int_s B_i(s) \times \Big(\underbrace{E_{int}(s)}_{\text{internal energy}} + \underbrace{E_{ext}(c(s))}_{\text{external energy}} \Big) ds \tag{3.19}$$

where the integration is carried over the region of support of E_i. In this way we also approximate $E_i(s)$ using the same B-spline parameterization. Then essentially we perform a local steepest descent, by moving one control point at a time until convergence. This is best described algorithmically as follows:

```
• $numiter = 0$
• New Iteration:
    – $numiter \mapsto numiter + 1$
    – Set $maxshift = 0.0$.
    – For all control points $i$
        ∗ Calculate current estimate of $E_{int}(s) = E^0_{int}$.
        ∗ Next estimate $E^{\delta x}_{int}$ which is is the internal energy function after shifting
          control point $(q_{i,x}, q_{i,y})$ by $\delta x$.[8]
```

[7] The use of the word *element* here is deliberate. This approximation is essentially a specialized application of the finite element method. We will discuss the details of this method in section 5.3.

[8] Typically $\delta x = \delta y = 0.5$ pixels.

- At this point calculate $\frac{\partial E_{int}}{\partial x}$ as:

$$\frac{\partial E_{int}}{\partial x} = \frac{E_{int}^{\delta x} - E_{int}^{0}}{\delta x}$$

- Then reset curve to original position, shift control point $(q_{i,x}, q_{i,y})$ by δy and similarly calculate $\frac{\partial E_{int}}{\partial y}$.
- Perform steepest descent at control point $(q_{i,x}, q_{i,y}$ by estimating the shift (dx, dy) as:

$$\begin{aligned} dx &= \beta \times \Big(\frac{\partial E_{int}(s)}{\partial x} + \frac{\partial E_{ext}(x,y)}{\partial x}\Big) \\ q_{i,x} &\mapsto q_{i,x} + dx \\ dy &= \beta \times \Big(\frac{\partial E_{int}(s)}{\partial y} + \frac{\partial E_{ext}(x,y)}{\partial y}\Big) \\ q_{i,y} &\mapsto q_{i,y} + dy \end{aligned}$$

  where β is the step size which is set adaptively.[9]

- $maxshift \mapsto max(|dx|, |dy|, maxshift)$.

- If $maxshift > threshold$ and $numiter < maxiterations$ `goto New Iteration:`
- End:

[9] This comes from the work of Hemant Tagare [no reference available]. The user sets a starting value $\beta = \beta_0$. Then until $\beta\sqrt{dx^2 + dy^2} < 3.0$. we scale $\beta \mapsto 0.5\beta$, to ensure that the optimization does not go too fast.

3.4 An Interactive Surface Segmentation Platform

In this section we describe a software platform which implements the ideas presented so far in this chapter. This software package called *SurfaceEdit* has been used to significantly reduce the time needed to accurately segment cardiac images. The package can automatically propagate contours from slice to slice and time-frame to frame if set in 'batch'-mode. Also it has on option to interpolate across frames saving the expert user the need to initialize all the frames before the automated segmentation can start, as illustrated in figure 3.6. Once a set of results is generated the user can use the 'editor module' (see figure 3.8) to correct the curves by moving the knot points. A simple click of the 'update' button updates, in almost real time, the 3D rendering of the surface in the 3D viewer shown in figure 3.7 which can also be displayed in long axis view as shown in figure 3.9.

SurfaceEdit has an intuitive user interface and can simultaneously display orthographic views of the 3D-image, for both Cartesian-space images (such as magnetic resonance) and cylindrical-polar space images (such as 3D ultrasound). It can also display multiple surface sections as well as multiple 3D surface rendering from any angle. All of the above can also be displayed in cine-mode. This is important as sometimes, especially in the case of ultrasound, the expert user needs to see the heart in motion in order to determine where the boundary is. Additionally the colors and transparency of the surfaces can be edited to allow the user to display one surface inside another. The software development was done in C++[97] using the Open Inventor 3D Graphics Toolkit[106] and the Motif[43] toolkit on the Silicon Graphics(SGI) Platform. The FITPACK package [25] was used for the implementation of the spline algorithms.

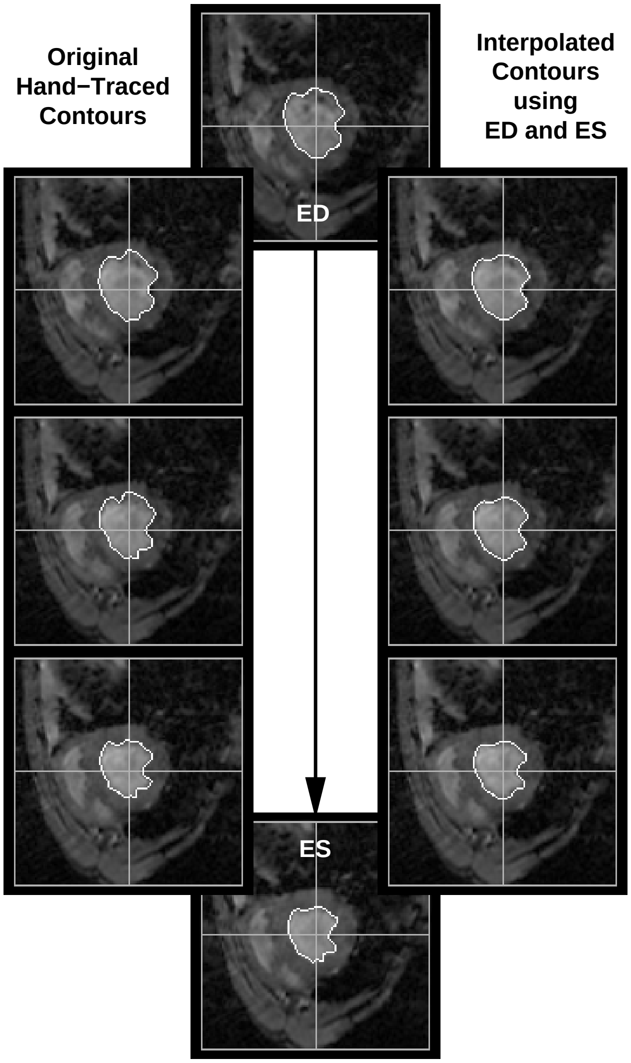
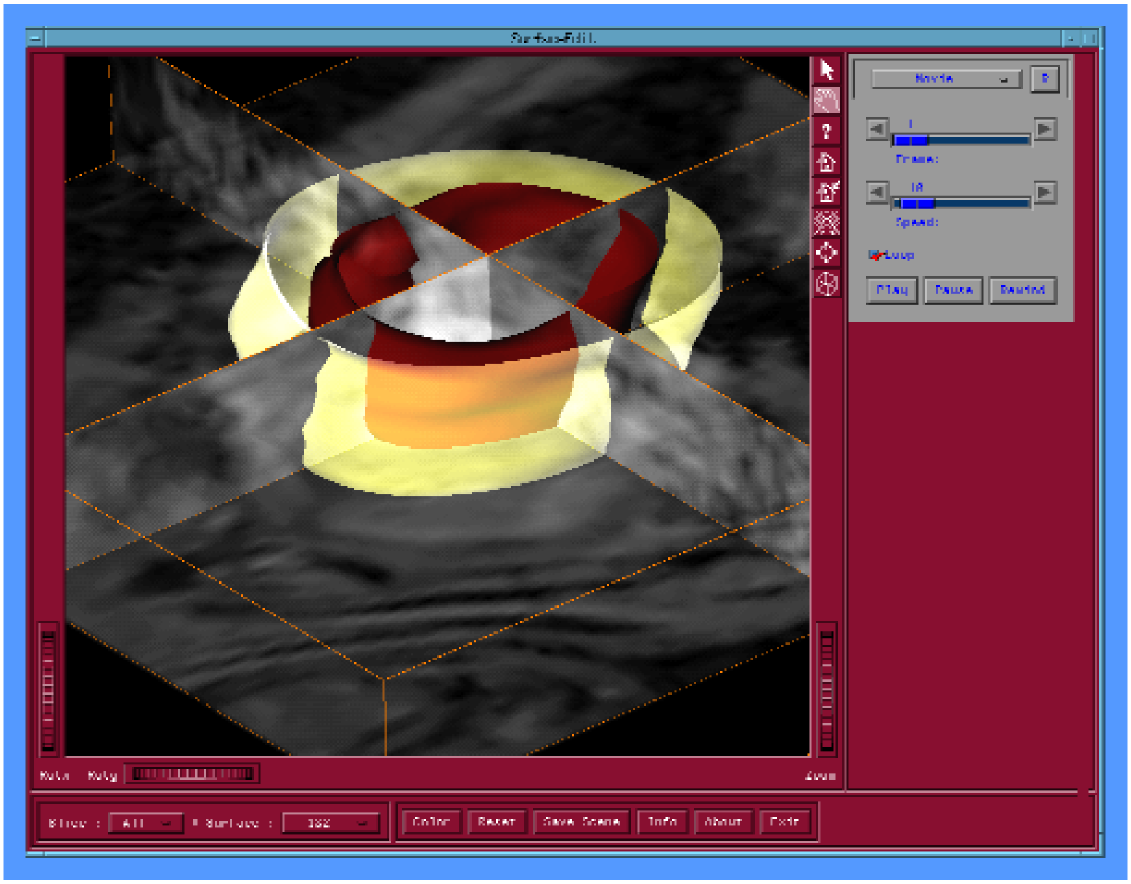


Figure 3.6: This figure illustrates the effectiveness of temporal interpolation. On the left the original hand traced contours for every second frame between end-diastole(ED) and end-systole(ES). One the right the contours produced by linearly interpolating between ED and ES. Though somewhat smoother they still are very close to the 'true' answer and would represent excellent initialization positions for the deformable contour algorithm.



Frame: 1

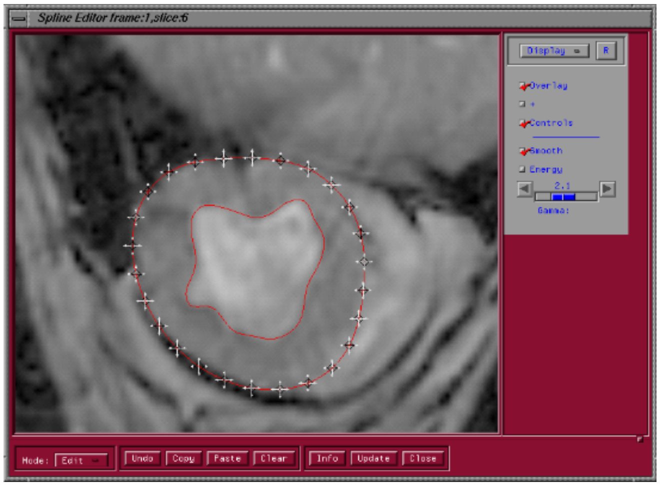
Speed: 10

☑ Loop

Slice : All

Surface : 332

Figure 3.7: 3D Viewer: this shows embedded surfaces and orthogonal image slices which help the expert user evaluate and correct the results of the segmentation. This viewer can also be used to display the images in ‘cine-mode’, i.e. in movie mode.



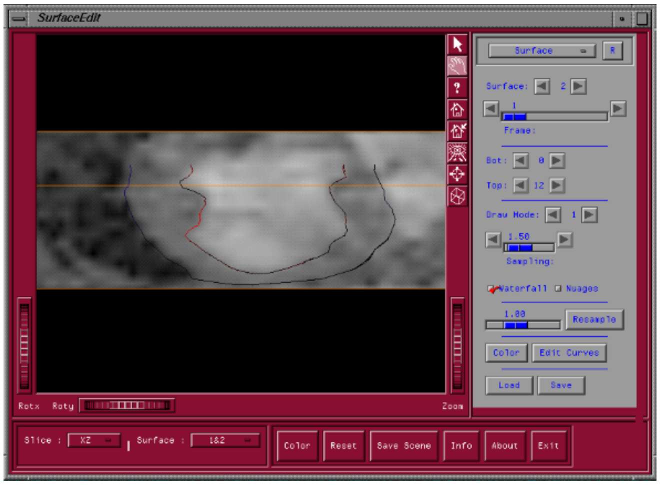
Spline Editor frame:1,slice:6

Display

☑ Overlay
□ +
☑ Controls
☑ Smooth
□ Energy
Gamma: 2.1

Mode: Edit

Figure 3.8: The Editor Module: we use a deformable contour-based segmentation to extract contours on short-axis slices and then form the surfaces using a Delaunay Triangulation. The contours are parameterized using B-splines which allows for easy editing by moving knot points, so that the expert user can easily correct for cases where the image data is not ideal. The editor also allows the user to edit up to four contours at any given time.



SurfaceEdit

Surface

Surface: 2
Frame: 1
Bot: 0
Top: 12
Draw Mode: 1
Sampling: 1.50
☑ Waterfall □ Nuages
1.00

Slice : XZ
Surface : 1&2

Figure 3.9: A long axis view: the user has almost instantaneous feedback in 3D of any changes made in the 2D contour editor. Contours can be propagated both spatially and temporally which reduces the amount of manual input necessary and takes advantage of the smooth variation of the contours across time and space.

Chapter 4

Geometrical Background and Techniques

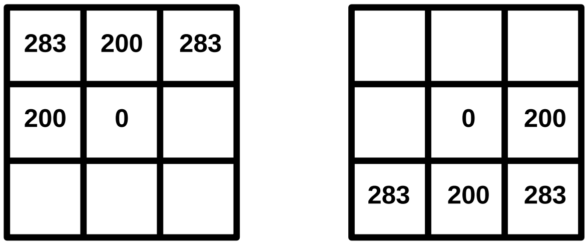
This chapter is divided in four sections. In section 4.1 we present techniques in two dimensions for interpolating between curves and generating 'symmetric-nearest' neighbors for points on two curves. In section 4.2, we review the geometry of surfaces and focus on techniques for surface construction from a set of planar contours, local curvature calculation and an extension to the 'symmetric-nearest' neighbors technique to 3D. Then in sections 4.3 and 4.4 we describe the two two major applications of exclusively geometrical ideas in this work, the generation of a hexahedral mesh for a volume and the shape-based tracking algorithm.

4.1 Geometrical Methods in Two-dimensions

In this section we describe two numerical techniques: the shape-based contour interpolation technique and the symmetric nearest-neighbor correspondence-finding technique.

4.1.1 Shape-Based Interpolation of Contours

The geometrical input to this work is slice-by-slice contours of the left ventricular surfaces, extracted using the methods presented in chapter 3. One of the key post-processing steps



283	200	283
200	0	

	0	200
283	200	283

Figure 4.1: Chamfer transformation templates. The two templates used by the dual chamfering processes to calculating the distance maps: template (a) for the top-to-bottom, left-to-right chamfering, and template (b) for the bottom-to-top, right-to-left chamfering.

in generating either equally sampled surfaces or tesselating between surfaces to generate solids is contour interpolation. Pengcheng Shi in his thesis [89] provides motivation for generating equispaced contours and an introduction to the Chamfer-based shape interpolation technique. In this work we extend this work [89] to the sub-pixel level. This is important because the movement of points on the left-ventricular wall is on average less than one voxel per frame, hence it is crucial that the input surfaces preserve as much as possible a sub-pixel resolution.

The first step in the interpolation process is to convert each contour into a gray-value 2D image, where pixel values represent the shortest distance of points from the contour, with positive values for inside the contour and negative values for outside. After the initialization, where we assign positive distances to points inside the contour and negative distances to points outside the contour, for all points that lie within 2 pixels of the contour, the complete distance map is calculated from two consecutive *chamfering* processes. The first chamfering updates the pixels row by row from top to bottom with a left-to-right ordering within the rows, using the leftmost template in figure 4.1. The second chamfering updates the pixels row by row from bottom to top with a right-to-left ordering within the rows, using the rightmost template in figure 4.1. These templates are scaled versions of the ones used in Shi [89], and this is done to improve sub-pixel resolution. The choices of the original unscaled two 3×3 templates have been justified

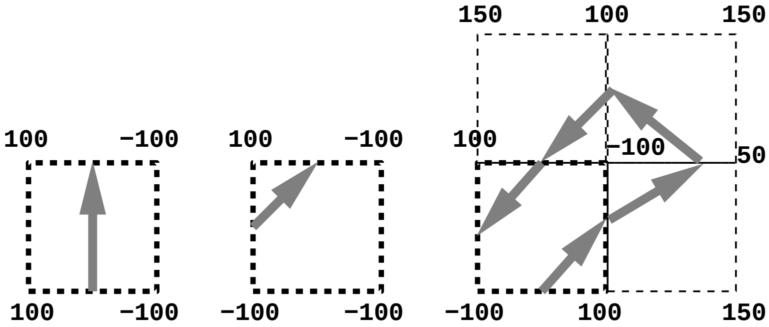


Figure 4.2: Extracting Zero Crossings. The numbers represent the distance values of the output distance map. The new contour (in this case half-way between the two original contours) is shown in a solid line. Note that the contours goes through squares where there is at least one sign change across one of the sides of the square.

to be near-optimal [45]. The resulting image represents the chamfer distance map of the given contour.[1]

The second step in the interpolation process is the generation of the output distance map. This is done by combining the input distance maps in the appropriate way. If we label two contours as c_1 and c_2 and their distance maps to be $d_m(c_1)$ and $d_m(c_2)$ respectively and we need to find the mean contour c_m, we first generate $d_m(c_m) = \frac{(d_m(c_1)+d_m(c_2))}{2}$.

The third step is the extraction of c_m from its distance map d_m. We define c_m to be the zero level set in the distance map d_m and we extract it using a border following scheme adapted from the level-set work of Malladi et al [87] (which in turn is derived from the marching cube work of Lorenson [61].) It is this last step which gives the method its sub-pixel resolution compared to the one used in Shi [89]. There are four possible combinations of distance values for each square connecting the centroids of four pixels; the three non trivial ones are shown in figure 4.2. These are:

1. All distances have the same sign. In this case the contour does not pass through

[1] The chamfer procedure is very efficient as it uses *integer* arithmetic only.

this square. This is the trivial case.

2. Two adjacent points have the same sign and the other two (also adjacent) have the opposite sign. In this case the contour intersects the sides of the square in which there is a sign transition (i.e. sides connecting a point of positive distance to a point of negative distance). See figure 4.2(left).

3. One point having a different sign from the other three. In this case the contour divides the square such that this one point lies on the one side and the other three on the other. See figure 4.2(middle).

4. Two non-adjacent points have the same sign and the other two (also non-adjacent) have a different sign as shown in 4.2(right). In this case the contour has to enter and exit the grid twice. To avoid ambiguity, we define the preferred direction of the contour to be anti-clockwise. Then the contour enters and exits preferentially to accommodate this constraint.

4.1.2 Symmetric Nearest Neighbor Correspondences in Curves

The estimation of a nearest neighbor correspondence between two curves (and two surfaces) plays an important role in many parts of the work presented in this thesis. In most computer vision applications and in previous work [89, 65] the estimation of initial correspondences is done using what we will term an 'asymmetric nearest neighbor' technique. In this case for each point on curve/surface c_1 the nearest point on curve/surface c_2 is found and labeled as the initial point. This has problems when the two curves are locally not parallel as whole regions of one curve map to a single point on the other curve. Also, whole regions on the second curve may not contribute to this map resulting in 'cutting corners' as demonstrated in figure 4.3. In this section we focus on the 2D case; we present extensions to the full three-dimensional case in section 4.2.4.

Motivated by the bimorphism work of Tagare[98, 99] we develop a symmetric technique to estimate initial correspondences without 'cutting corners'. This is important so as to ensure that as much as possible the whole of curve c_1 maps to the whole of curve

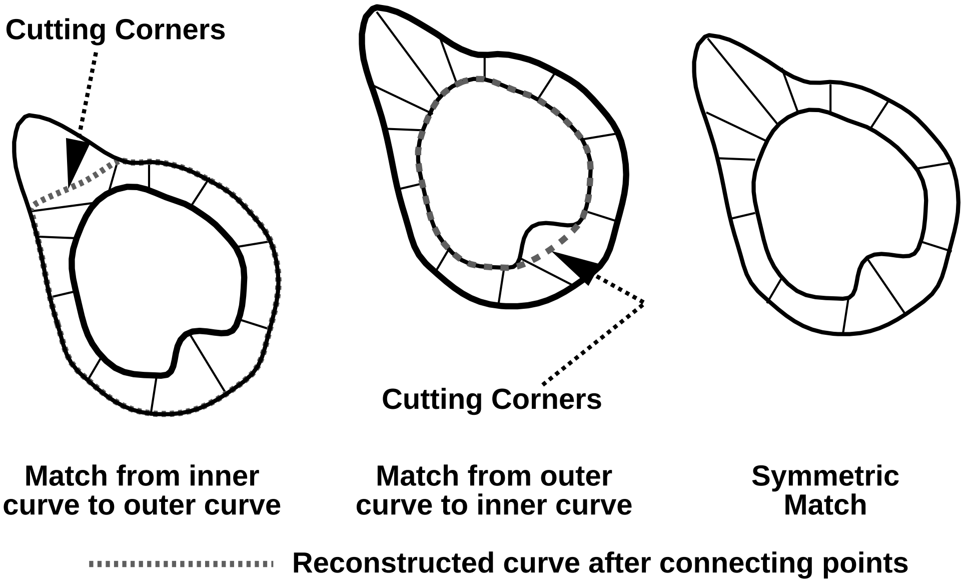

Cutting Corners
Cutting Corners
Match from inner curve to outer curve
Match from outer curve to inner curve
Symmetric Match
Reconstructed curve after connecting points

Figure 4.3: Illustration of problems with asymmetric nearest neighbor matches. The two examples (left and middle) where the correspondence is driven exclusively in one direction show problems such as 'cutting corners' when the two curves are not roughly parallel. In the third case by using a symmetric nearest neighbor map the problem is avoided.

c_2 and that the map is free from singularities (such as two points mapping to the same point) which are not either permissible or plausible in the areas of application of this algorithm.[2] Further, we emphasize that the aim of this technique is *not* to estimate a registration between two curves or two surfaces but rather to generate a set of initial correspondence vectors based purely on distance that can be used as a starting point for a nonrigid registration/correspondence method which incorporates information such as shape.[3] This method is useful in its own right in the case of mesh generation.

[2] In the case of true 3D deformation, material particles cannot appear or disappear. This requires that the map between two solids (and surfaces) be invertible.

[3] We use the 3D extension of this algorithm to initialize the shape-based tracking algorithm in section 4.4.

The symmetric nearest neighbor algorithm has three steps as follows:

1. For all points on curve c_1 find the nearest neighbors on curve c_2 using a Euclidean distance metric. So for example for a point p_1 on curve c_1 we have a corresponding point p_2 on curve c_2. Then for point p_2 estimate its nearest neighbor $\hat{p}_1$ on c_1. If $p_1 = \hat{p}_1$ then the points (p_1, p_2) are symmetric nearest neighbors and the match is retained. Otherwise, the match is discarded.
2. For all points on curve c_1 which do not have symmetric nearest neighbors on c_2, find a matching point on c_2 by interpolating between the matching points of its neighbors. We do this until all points on c_1 have a matching point on c_2.
3. Smooth the displacement field slightly to eliminate potential near-singularities.

Step 1 is self-explanatory, although it can be extremely time consuming for large surfaces (on the order of 10,000 points each) unless the points are somehow sorted to reduce the search time. The more difficult part is the implementation of step 2, which we now describe.

Here we take advantage of the fact that a curve can be parameterized using its arclength. An example will help to illustrate the point: consider the case that curve c_1 has four points $(c_1(0.0), c_1(0.25), c_1(0.5), c_1(0.75))$ which match to different positions on c_2, as illustrated by figure 4.4, and noting that $c_1(s_1)$ represents the point on curve c_1 at arclength of $s = s_1$. In this case step 1 resulted in three symmetric neighbor pairs and left one point without a match. We can represent the points on c_2 by their arclengths as follows:

$$[c_1(0.0), c_1(0.25), c_1(0.5), c_1(0.75)] \mapsto [c_2(0.0), c_2(0.4), ??, c_2(0.9)]$$

In this case point $c_1(0.5)$ has no corresponding point after step 1. To generate a match for $c_1(0.5)$ we interpolate between the corresponding points of $c_1(0.25)$ and $c_1(0.75)$ the nearest points to $c_1(0.5)$ on c_1 that do have *symmetric* nearest neighbors. This results

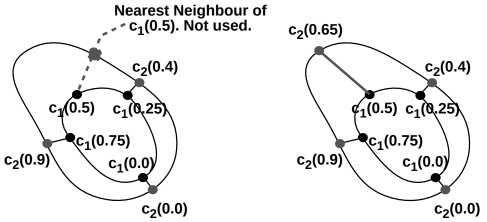


Figure 4.4: An example of the 2D Implementation of the symmetric nearest neighbor algorithm. In this case we try to map the inner curve c_1 to the outer curve c_2. Curve c_1 is defined by four points $(c_1(0.0), c_1(0.25), c_1(0.5), c_1(0.75))$, all of which apart from $c_1(0.5)$ have a symmetric nearest neighbor. The nearest neighbor of $c_1(0.5)$ is shown on the left (bad) and the point $c_1(0.5)$ is mapped to by the algorithm is shown on the right ($c_2(0.65)$ good!).

in $c_1(0.5) \mapsto c_2(0.65)$. Note that we in effect place the corresponding point of $c_1(0.5)$ at the centroid of the (shortest) segment[4] of the curve c_2 connecting the corresponding points of its neighbors ($c_2(0.4)$ and $c_2(0.9)$). This generalization will become useful when we move to 3D.

So the result of step 2 is:

$$[c_1(0.0), c_1(0.25), c_1(0.5), c_1(0.75)] \mapsto [c_2(0.0), c_2(0.4), c_2(0.65), c_2(0.9)]$$

Then in step 3 we smooth the displacements slightly[5] to ensure no near singularities.

[4] Since the curve is closed there are two possible segments of the curve connecting any two points on it. We choose the shortest segment. Then we essentially interpolate along this segment, using the arclength, to find the position of the corresponding point for $c_1(0.5)$, as $0.65 = 0.5(0.4 + 0.9)$.

[5] We smoothed the arclengths on c_2 by convolving them with a small Gaussian kernel.

This could result in a map like:

$$[c_1(0.0), c_1(0.25), c_1(0.5), c_1(0.75)] \mapsto [c_2(0.05), c_2(0.38), c_2(0.62), c_2(0.88)]$$

which tries to equispace the points on c_2 subject to staying close to their original positions. For this approach to work well in practice where the curves are discretized, c_2 has to be sampled much more finely than c_1 (typically 5 to 8 times more).

4.2 Geometrical Methods in Three-dimensions

In section 4.2.1– 4.2.3 we describe the process of constructing a surface from planar contours, non-shrinking surface-smoothing and for the estimation of the local curvatures of a discretized surface.[6] This process is summarized graphically in figure 4.5. In section 4.2.4 we describe an extension of the symmetric nearest neighbor algorithm to 3D.

4.2.1 Delaunay Triangulation Between Planar Contours

In this section, we describe a method to calculate the 2D-constrained Delaunay triangulation [89, section 3.4] for a surface to be constructed from planar contours oriented in the same direction (in this case anticlockwise). This restriction enables the implementation of a simple and fast triangulation algorithm. This algorithm creates the triangulation which has the smallest total length of triangle sides of all possible triangulations between the two planar contours. Consider the case of figure 4.6(A). Here two adjacent triangles are shown. If we flip the middle line (drawn as a dotted line) we can create an alternative triangulation. This triangulation method is optimal in that *no* flipping of connections can decrease the total length of all the sides of all the triangles. For the case of constructing

[6] All of this material is directly derived from the work of Pengcheng Shi [89] and the interested reader is referred to Shi [89, 90] for the details. In this work we simply highlight some of the aspects of this work which are particularly important in the context of this thesis.

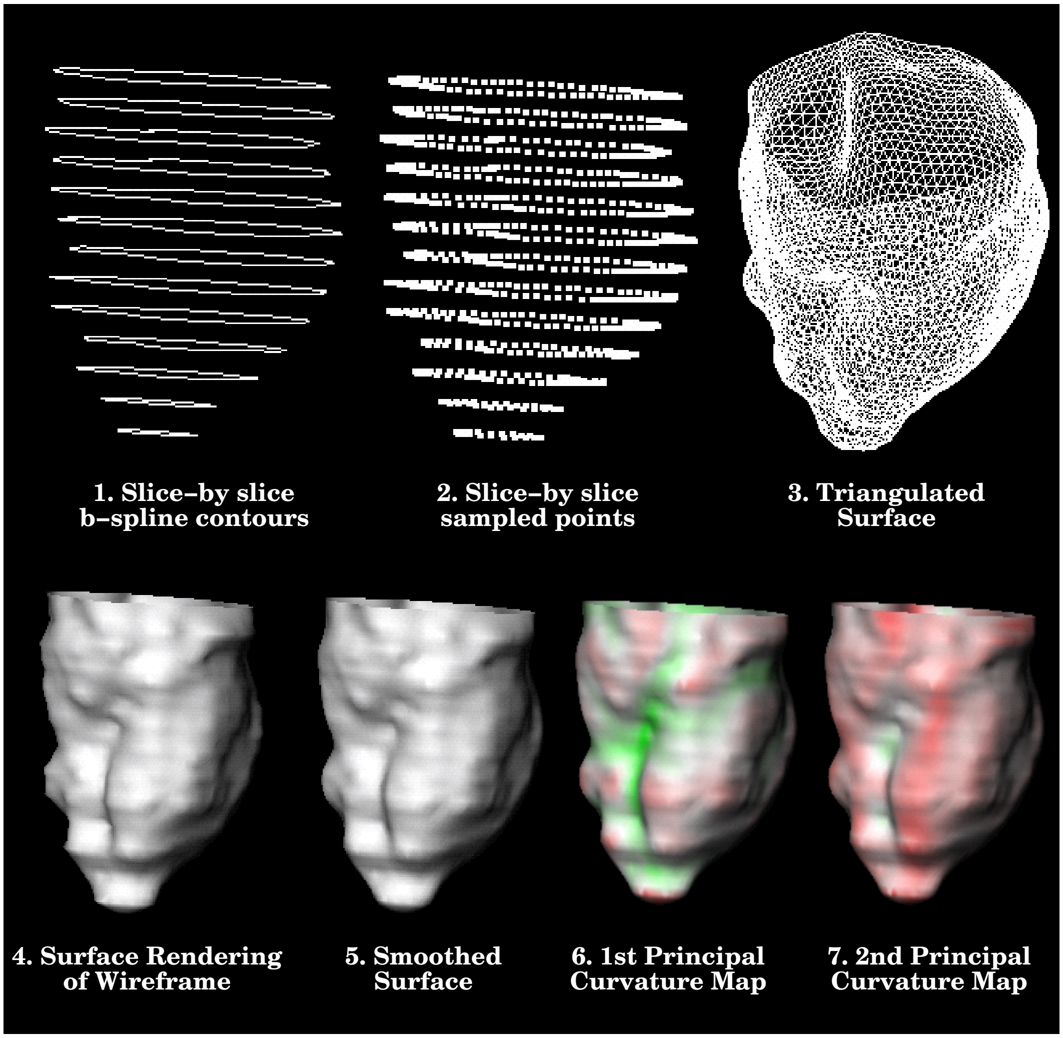

1. Slice–by slice b–spline contours
2. Slice–by slice sampled points
3. Triangulated Surface
4. Surface Rendering of Wireframe
5. Smoothed Surface
6. 1st Principal Curvature Map
7. 2nd Principal Curvature Map

Figure 4.5: Steps involved in moving from slice by slice contours to full surface representation. (1) Slice by slice B-spline parameterized contours as extracted by the segmentation process. (2) Discretized contours as equally-spaced points. (3) Formation of wire-frame by Delaunay triangulation. (4) Surface rendering of surface. (5) Smoothing of surface using non-shrinking smoothing algorithm. (6)+(7) First and second principal curvatures of surface.

a set of triangles between two discretized, anticlockwise oriented, closed planar contours the procedure is as follows:

- `Initialization Step:`

  1. Initialize empty list of triangles.
  2. For a point p_1 on contour c_1 find the nearest point (in the Euclidean sense) to it p_2 on contour c_2. For this p_2 find the nearest point to it $\hat{p}_1$ on contour c_1.
  3. If $p_1 = \hat{p}_1$ label $s_1 = p_1, s_2 = p_2$ as the starting pair of points and goto `Connection step`.
  4. If $p_1 \neq \hat{p}_1$ choose another point on contour c_1 and repeat the initialization step.
  5. The process fails if there is no point p_1 for which this criterion is satisfied. (This is extremely unlikely).

- `Connection Step:`

  1. Given starting points s_1, s_2 find the two test points t_1 and t_2. t_1 is the next point along c_1 from s_1, and t_2 which is the next point along c_2 from s_2. See figure 4.6B.
  2. If $|t_1 - s_2| < |t_2 - s_1|$ label next point $n_p = t_1$ else $n_p = t_2$, and add triangle s_1, s_2, n_p to the list.
  3. If $n_p = t_1$ then set $s_1 = n_p$, else set $s_2 = n_p$.
  4. If $s_1 = p_1$ and $s_2 = p_2$ goto `End`.
  5. Repeat `Connection Step`.

- `End:` procedure stops as we have returned to the starting point.

Proof: In this section we prove that this algorithm generates the triangulation which has the smallest total lengths of the sides of the triangles. First, note that clearly all points

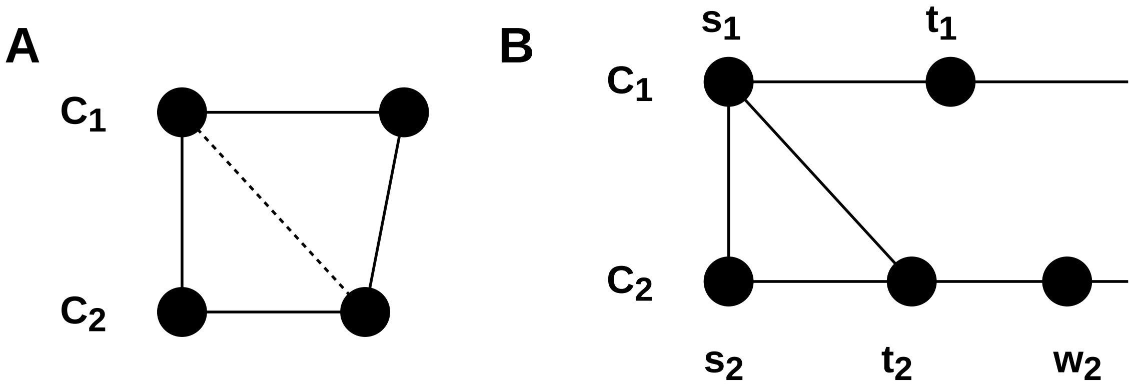


Figure 4.6: Schematic for the proof of the optimality of the triangulation procedure.

on c_1 will be connected to their adjacent neighbors on c_1 and similarly for all points on c_2. This reduces the proof to finding the optimal 'inter-connections' between c_1 and c_2. Given a good starting point p_1 and p_2, we can always choose the shortest possible length (in the connection step) for the next point to be attached, hence this further reduces the proof to showing that the proposed method of initialization using points p_1 and p_2 which are symmetric nearest neighbors is appropriate. This is equivalent to points p_1 and p_2 being part of a triangle in the optimal triangulation.

Instead of using this method for initialization, let us consider the case were we initialize using points g_1 and g_2 which is the pair that generates the globally smallest inter-connection distance between curves c_1 and c_2 as found by exhaustive search. Clearly this pair would satisfy the criteria for optimality. Then we proceed around the contours as per the connection step. Consider the case of figure 4.6B, and assume that t_1 and t_2 are symmetric nearest neighbors. The next triangle will either be s_1, t_2, w_2 if $w_2 - s_1 < t_1 - t_2$ (bad case) or s_1, t_2, t_1 otherwise.

This further reduces the proof to showing that $t_1 - t_2 < w_2 - s_1$. Since t_1 and t_2 are symmetric nearest neighbors, this implies that locally c_1 and c_2 are almost parallel. Hence $t_1 - w_2 < s_1 - w_2$. But $t_1 - t_2 < t_1 - w_2$ as t_1 and t_2 are symmetric nearest neighbors. Therefore $t_1 - t_2 < s_1 - w_2$ which concludes the proof.

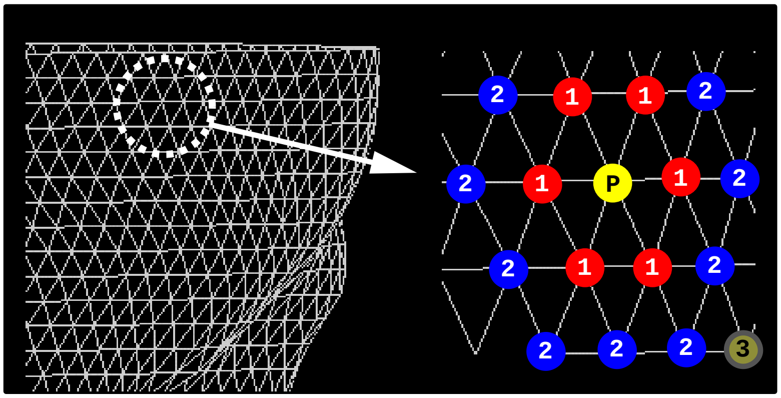


Figure 4.7: Left: portion of a triangulated endocardial surface. Right: closeup illustrating the neighbors of point p. Points labeled (1) are the first order neighbors, points labeled 2 are second ordered neighbors and the point labeled 3 is a third order neighbor. (Not all second and third order neighbor points are shown.)

This implies that in an optimal (from a shortest length viewpoint) triangulation the side t_1, t_2 will exist if t_1 and t_2 are symmetric nearest neighbors. So we can start the triangulation using any pair of symmetric nearest neighbors, as opposed to the more computationally expensive alternative of finding the pair of points g_1, g_2 described above.

Connectivity Distance: The Delaunay triangulation defines the connectivity of the points on each surface and provides the all-important concept of a neighboring point, as illustrated in figure 4.7. We further define the distance between the two points to be the order of their connection. A point has a distance of 0 with itself, a distance of 1 with a first order neighbor, a distance of 2 with a second order neighbor and so on. We will call this the connectivity distance d_c.

4.2.2 Non-Shrinking Surface Smoothing

Once the surface triangulation has been constructed, we smooth the surfaces to correct for noise in the segmentation and to make the computation of curvatures more stable.

In this work we use the non-shrinking two stage Gaussian algorithm proposed by Taubin [100]. It is compared to the more typical one stage Gaussian filtering in Shi [89]. The algorithm works as follows:

- For all points p on surface s define the set of its first order neighbors W.

  1. For all odd-numbered iterations
  $$p \mapsto (1 - \lambda_1)p + \lambda_1 \sum_{q \in W} q$$
  2. For all even-numbered iterations
  $$p \mapsto (1 - \lambda_2)p + \lambda_2 \sum_{q \in W} q$$

with $\lambda_1 = 0.33$ and $\lambda_2 = -0.34$. This alternating smoothing and unsmoothing process was shown to preserve the shape visually better. An example is shown figure 4.5 parts 4 and 5. (For further analysis again see Shi [89, pages 66–75].)

4.2.3 Curvature Computation

Here we briefly review the method used for the computation of curvature. First we briefly review some basic concepts of differential geometry (see DoCarmo [28] and also Shi [89, pages 76–91] for more details.)

Differential geometry of a surface: A general surface $S \subset \mathcal{R}^3$ is defined as follows: For each point $p \in S$ there exists a neighborhood $V \in \mathcal{R}^3$ and a map $\mathbf{x} : U \mapsto V \bigcap S$ on an open set $U \bigcap \mathcal{R}^2$ onto $V \bigcap S \subset \mathcal{R}^3$ such that:

- $\mathbf{x}(u, v) = (x(u, v), y(u, v), z(u, v)) \in S$ is differentiable.
- $\mathbf{x}$ is a homeomorphism. And since $\mathbf{x}$ is continuous by the previous condition, this means that $\mathbf{x}$ has an inverse $\mathbf{x}^{-1} : V \bigcap S \mapsto U$ which is continuous; that is, $\mathbf{x}^{-1}$ is the restriction of a continuous map $F : W \subset \mathcal{R}^3 \mapsto \mathcal{R}^2$ defined on an open set W containing $V \bigcap S$. (This condition prevents self-intersections in S, and also means that objects defined in terms of a parameterization do not depend on this parameterization but rather only on the set S itself.)

- *For each $q \in U$, the differential $d\mathbf{x}_q : \mathcal{R}^2 \mapsto \mathcal{R}^3$ is one-to-one.* (This condition guarantees the existence of a tangent plane at all points of S).

The mapping $\mathbf{x}$ is called a *parameterization* or a *system of local coordinates* in a neighborhood of p. The neighborhood $V \bigcap S$ of p in S is called a *coordinate neighborhood.* This definition allows us to place each point p of a regular surface in a coordinate neighborhood, and to define the local properties of point p in terms of the coordinates u and v.

The plane $d\mathbf{x}_q$, which passes through $\mathbf{x}(q) = p$, does not depend on the parameterization $\mathbf{x}$. This plane is called the *tangent plane* to S at p, and is denoted by $T_p(S)$. The choice of the parameterization $\mathbf{x}$ determines a basis $\{(\partial\mathbf{x}/\partial u)(q), (\partial\mathbf{x}/\partial v)(q)\}$, or $\{\mathbf{x}_u(q), \mathbf{x}_v(q)\}$, of $T_p(S)$, called the basis associated to $\mathbf{x}$. Similarly, a unit *normal vector* at point $\mathbf{x}(q) = p$ of S is determined by

$$N_p = \frac{\mathbf{x}_u \wedge \mathbf{x}_v}{|\mathbf{x}_u \wedge \mathbf{x}_v|}(q)$$

where $\wedge$ denotes cross product. See figure 4.8 for an illustration.

We then proceed to define the following quantities at point $p = (u_0, v_0)$:

$$E(u_0, v_0) = < \mathbf{x}_u, \mathbf{x}_u > \tag{4.1}$$

$$F(u_0, v_0) = < \mathbf{x}_u, \mathbf{x}_v > \tag{4.2}$$

$$G(u_0, v_0) = < \mathbf{x}_v, \mathbf{x}_v > \tag{4.3}$$

$$e(u_0, v_0) = - < N_u, \mathbf{x}_u > = < N, \mathbf{x}_{uu} > \tag{4.4}$$

$$f(u_0, v_0) = - < N_v, \mathbf{x}_u > = < N, \mathbf{x}_{uv} > = - < N_u, \mathbf{x}_v > \tag{4.5}$$

$$g(u_0, v_0) = - < N_v, \mathbf{x}_v > = < N, \mathbf{x}_{vv} > \tag{4.6}$$

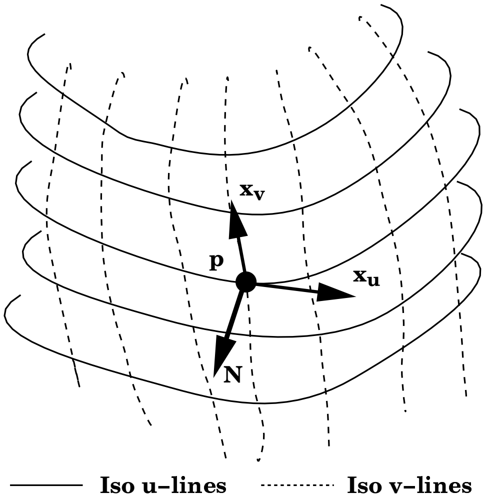


Figure 4.8: At any point p in a differentiable surface we can find a local parameterization x parameterized along vectors u and v. We also define the outward normal of the surface at this point to be N.

These quantities which appear in the definition of the first and second fundamental forms of the surface [28] enable us to define the the *Weingarten Mapping Matrix* as follows:

$$[\beta] = -\begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} E & F \\ F & G \end{pmatrix}^{-1} \tag{4.7}$$

This is also known as the shape operator matrix of the surface. This matrix determines surface shape by relating the intrinsic geometry of the surface to the Euclidean (extrinsic) geometry of the embedding space. The *Gaussian curvature* of a surface can be defined from the Weingarten mapping matrix as its determinant:

$$K = det[\beta] = \frac{eg - f^2}{EG - F^2} \tag{4.8}$$

Meanwhile, the *mean curvature* of a surface is similarly defined as half of the trace of the Weingarten mapping matrix:

$$H = \frac{tr[\beta]}{2} = \frac{eG - 2fF + gE}{2(EG - F^2)} \tag{4.9}$$

We also define the principal curvatures which are the eigenvalues of the Weingarten mapping matrix, with their directions along the two eigenvectors. They are can be calculated in terms of the Gaussian and mean curvatures as:

$$\kappa_1 = H + \sqrt{H^2 - K} \tag{4.10}$$

$$\kappa_2 = H - \sqrt{H^2 - K} \tag{4.11}$$

Calculating the curvature at a point on a discretized surface: We calculate the principal curvatures κ_1 and κ_2 at a point p on a discretized surface s by first fitting a biquadratic surface to the collection of all the points r on s that have a connectivity distance $d_c(p, r) < t$ where t is a constant and defines the scale of the neighborhood. This has to be large enough to avoid local segmentation noise and small enough to capture the local differential properties. In this work where surfaces are sampled to 0.5 voxel spacing we use a window size of $t = 4$.

Before the biquadratic surface is constructed, we first rotate the coordinates of all the points that satisfy $d_c(p, r) < t$ to a local coordinate system with point p as the origin, the local surface normal N as the z axis and two tangent directions as x and y axis. We estimate the normal N by averaging the normals of all the triangles of which point p is a node. Then we estimate the coefficients of the biquadratic surface which takes the form:

$$z = h(x, y) = a_1x^2 + a_2xy + a_3y^2 + a_4x + a_5y \tag{4.12}$$

These are estimated using a least squares fit to the neighborhood points , and can be used then to form the Weingarten mapping matrix and hence compute the curvatures. An example of such curvatures is shown graphically in figure 4.5 parts 6 and 7.

4.2.4 Symmetric Nearest Neighbor Correspondences in Surfaces

In this section, we extend the work of section 4.1.2 to three-dimensions. It is generally true that easy geometrical problems in 2D become almost impossible in 3D as a result of the loss of the arclength parameterization. So the key step here is to find a way of replacing the arclength parameterization. We attempt to do this by using the Euclidean distance and partially using a connectivity distance defined on the surface. We focus here on steps 2 and 3 of the algorithm; step 1 is identical to the 2D case and need not concern us any further.

Some additional definitions: If a point p_1 on surface s_1 is mapped to a point p_2 on surface s_2 then we define the displacement vector $u(p_1) = p_2 - p_1$. Any point p_1 on s_1 that has a corresponding point on s_2 also by definition has a displacement vector.

A description of Step 2: This is the step in which we find corresponding points for all the points on p_1 that do not have a symmetric nearest neighbor. It is best explained algorithmically as follows: (see also figure 4.9.)

```
• Set $i = 0$
• beginning:
• Let point $p_1$ be point $p_i$ on surface $s_1$.
    1. If point $p_1$ has a displacement vector goto endloop.
    2. If none of the first-order neighbors of point $p_1$ have a displacement vector goto
       endloop.
    3. Average the displacement vectors of all the first order neighbors of point $p_1$
       that do have displacement vectors, to generate a displacement vector $u_1$
```

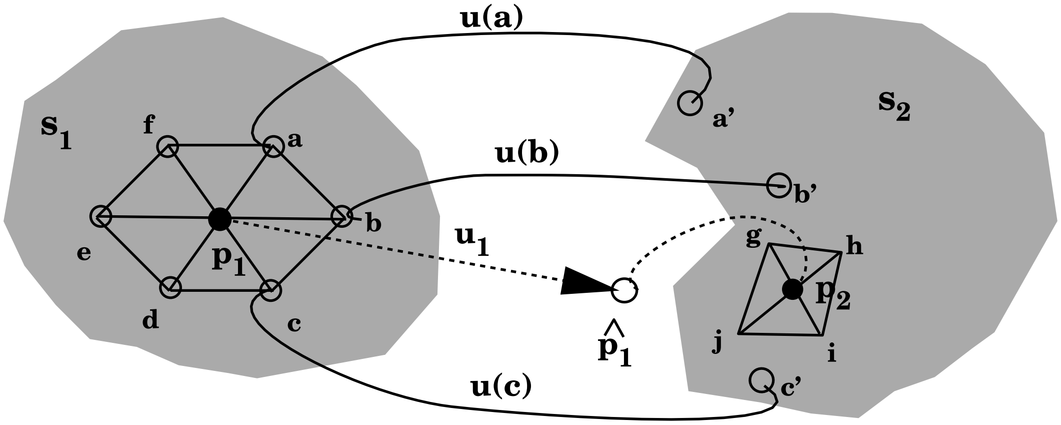


Figure 4.9: Symmetric Nearest Neighbor Algorithm in 3D. A portion of surface s_1 shown on the left centered on a point p_1 which has first order neighbors a, b, c, d, e, f. Of these neighbors a, b, c have symmetric nearest neighbors a', b', c' on s_2 shown on the right. p_1 itself does not have a symmetric nearest neighbor on s_2. We generate the first estimate of the position of the corresponding point of p_1, $\hat{p}_1$, by averaging $u(a), u(b)$ and $u(c)$ the displacement vectors of points a,b,c to estimate a vector u_1 and translating p_1 by u_1. Then $\hat{p}_1$ is mapped to surface s_2 by finding its (asymmetric) nearest point on s_2. This is point p_2 which is the corresponding point of point p_1 on surface s_2. We also define $u(p_1)$ (not shown) as $u(p_1) = p_2 - p_1$. We further show the first order neighbors of p_2 on surface s_2 labeled as g, h, i and j.

4. Translate p_1 by u_1 to a point $\hat{p}_1$.
5. Find the nearest neighbor of point $\hat{p}_1$ on s_2. Label this point as p_2 and then calculate the displacement vector $u(p_1) = p_2 - \hat{p}_1$. p_2 is also the corresponding point of p_1. Now point p_1 has a displacement vector.

- `endloop:` $i \mapsto i + 1$.
- If $i < N_p$ where N_p=number of points on s_1 `goto beginning`.
- If not all points on s_1 have a displacement vector set $i = 0$ and `goto beginning`.
- `end.`

So long as one point on s_1 has a symmetric nearest neighbor after step 1 this algorithm will generate a set of point pairs. This algorithm is illustrated in figure 4.10. We next consider approaches to step 3, the smoothing step.

A Euclidean approach to smoothing: This approach is labeled Euclidean as the term being smoothed is the 'Euclidean distance'. This is an alternating iterative process, and it works as follows:

- For all odd numbered iterations and for all points p_1 on s_1:
  1. Find the average displacement vector u_n of all its first order neighbors. (These would be $u(a), u(b), u(c), u(d), u(e)$ and $u(f)$ of figure 4.9.)
  2. Generate a new displacement vector $u(p_1) \mapsto 0.75u(p_1) + 0.25u_n$.
- For all even number iterations and for all points p_1 on s_1:
  1. Translate p_1 by $\hat{u}$ to a point $\hat{p}_1$.
  2. Find the nearest neighbor of point $\hat{p}_1$ on s_2. Label this point as p_2 and then calculate the displacement vector $u(p_1) = p_2 - \hat{p}_1$. p_2 is also the corresponding point of p_1 on s_2.

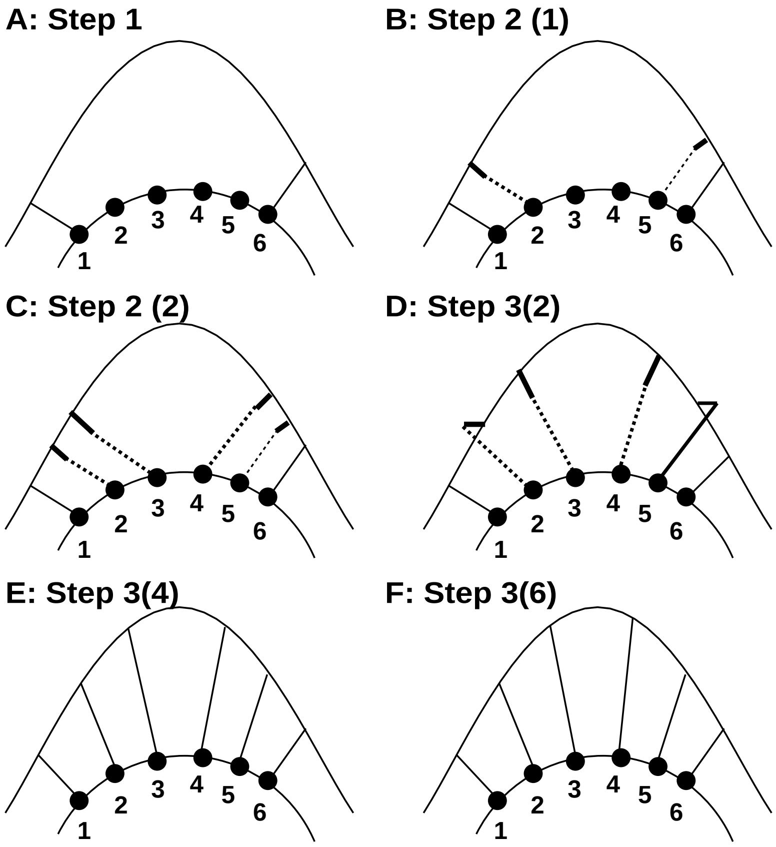


Figure 4.10: Symmetric 3D Nearest Neighbor Algorithm. (This is shown in 2D for simplicity.) Part A shows the result of step 1, where only points 1 and 6 have corresponding points. In part B (Step 2 iteration 1) points 2 and 5 also acquire displacements as at least one of their neighbors has a displacement (points 1 and 6 respectively). Note that the displacement vectors of points 2 and 5 have two parts. The first shown using a dotted line is the average of the displacements of the neighbors, and the second part, shown using a solid line, is as a result of mapping this position to next surface. In part C (Step 2 iteration 2) points 3 and 4 also have displacements. Parts D-F show iterations of the Euclidean distance based approach to smoothing. Note how the map becomes progressively more regular.

A connectivity distance approach to smoothing: In this case we try to maximize the connectivity distances of the corresponding points p_2 on s_2 as follows:

- For all iterations and all points p_1 on s_1:

    1. Generate the set N which contains all the corresponding points of the first order neighbors of p_1 on surface s_2. (Note that the points in N lie on s_2, and would be points a', b', c', d', e' and f'[7] of figure 4.9.)
    2. Generate the set W which contains p_2 and all its first order neighbors. (Again note that the points in W lie on s_2, and would be points g, h, i, j and p_2 of figure 4.9.)
    3. For all points in W look for the point $\hat{p}_2$ that maximizes:

        $$\hat{p}_2 = \begin{matrix}\arg\min \\ p \in W\end{matrix} \left(\begin{matrix}\arg\max \\ q \in N\end{matrix} d_c(p,q) \right)$$

        In words this tries to find the point in W that is nearest to the centroid of N as defined by connectivity distance.
    4. Let $u(p_1) = \hat{p}_2 - p_1$ and let $\hat{p}_2$ be the corresponding point of p_1 on s_2.

This method has the advantage of relying less on the Euclidean distance and more on the geometry of the surfaces. It is computationally more expensive however.

As a final post-processing step for both of this approaches, surface s_1 is translated by translating each point p_1 on s_1 by its corresponding displacement vector $u(p_1)$ to a new point $\hat{p}_1$ and then slightly smoothed using 5 iterations of the non-shrinking algorithm described in section 4.2.2. Then the resulting $u(p_1)$ is adjusted to be $u(p_1) \mapsto u(p_1) + \hat{p}_1 - p_1$. This is needed as it is computationally not feasible to have surface s_2 be sampled a factor of $5 - 8$ times more finely than s_1. In practice s_2 is sampled three to four times more finely than s_1.

[7] d', e' and f' are not shown in the figure but will by now exist as each point has a corresponding point.

It also worth noting that there is no algebraic proof of the quality of these methods.[8] They have been both tested (and especially the Euclidean approach) and have been found to perform well over a large number of datasets.

4.3 Generating Hexahedral Meshes

After we have extracted and tessellated the endocardial and epicardial surfaces we need to construct a solid mesh in the space between them, to represent the heart wall muscle, the myocardium. This is needed for the application of the finite element method[9] in the deformation estimation stage. We choose to divide this solid into hexahedral elements as these have significant numerical advantages over the more common tetrahedral elements.

As mesh generation in three-dimensions is a notoriously difficult problem for complicated geometries[8], we propose here an algorithm which takes advantage of the 'cylindrical-like' geometry of the left ventricle, to make the problem easier. The two basic building blocks of the algorithm are the shape-based contour interpolation method of section 4.1.1 and the symmetric nearest neighbor correspondence algorithm described in section 4.1.2. The algorithm is best described with reference to figures 4.11 and 4.12. It consists of four steps as follows:

- Step 1: Interpolate on a contour by contour basis between the endocardial and epicardial surfaces using shape-based interpolation to generate an appropriate number of in-between interpolated surfaces (typically 3 or 4). Because of the greater geo-

[8] The odd numbered iterations of the Euclidean based smoothing method can be proven to converge. Essentially we are solving a system of the form $[A]x_k = x_{k+1}$ where A is a square $3N \times 3N$ smoothing matrix and x_k is the $3N$ vector of the positions of all the points in iteration k. This is analogous to the Gauss Seidel method[85] which can be shown to converge if the matrix A is diagonally dominant i.e. $A_{ii} > \frac{1}{2}\sum_{r=1}^{N} A_{ir}, \forall i \in [1, N]$. In the Euclidean based smoothing method this is the case as $A_{ii} = 0.75$, and $\sum_{r=1}^{N} A_{ir} = 1$. It is harder to show convergence for the odd numbered iterations as the mapping step is non-linear. However in practice the method converges very rapidly (in 3-5 iterations.)

[9] The finite element method is described in section 5.3.

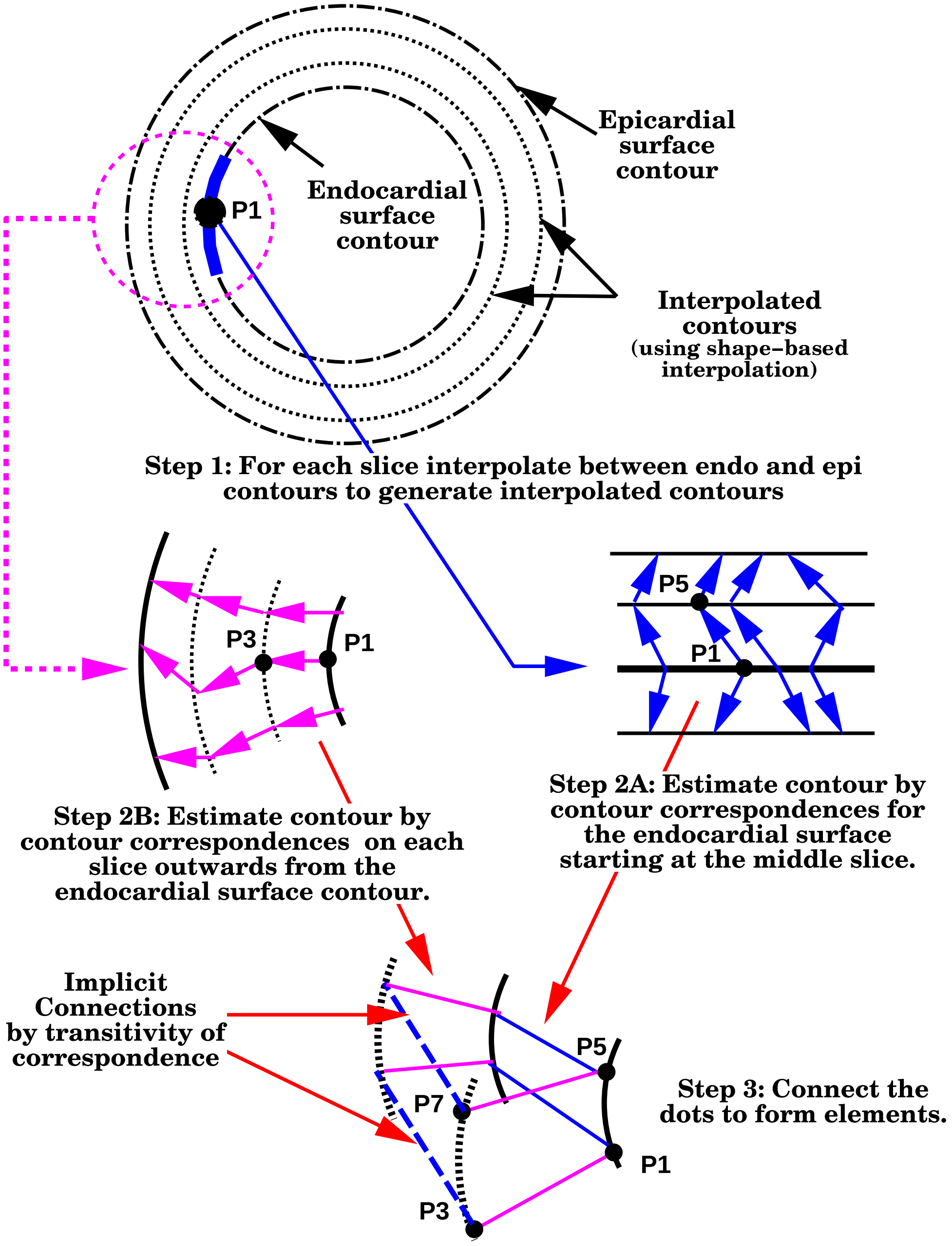

Epicardial surface contour
P1
Endocardial surface contour
Interpolated contours (using shape−based interpolation)
Step 1: For each slice interpolate between endo and epi contours to generate interpolated contours
P3
P1
P5
P1
Step 2B: Estimate contour by contour correspondences on each slice outwards from the endocardial surface contour.
Step 2A: Estimate contour by contour correspondences for the endocardial surface starting at the middle slice.
Implicit Connections by transitivity of correspondence
P5
P7
P1
P3
Step 3: Connect the dots to form elements.

Figure 4.11: A schematic of the mesh generation process. First, we interpolate between the endocardial and epicardial surfaces on a contour by contour basis using shape based interpolation to create the interpolated surfaces. Next, we find correspondences between the contours on the endocardial surface starting at the middle level using the 2D algorithm of described in section 4.1.2. Next, we find correspondences on each slice starting from the endocardium, using the same algorithm. Finally, we connect the dots to generate the elements. A 3D illustration of this can be found in figure 4.12

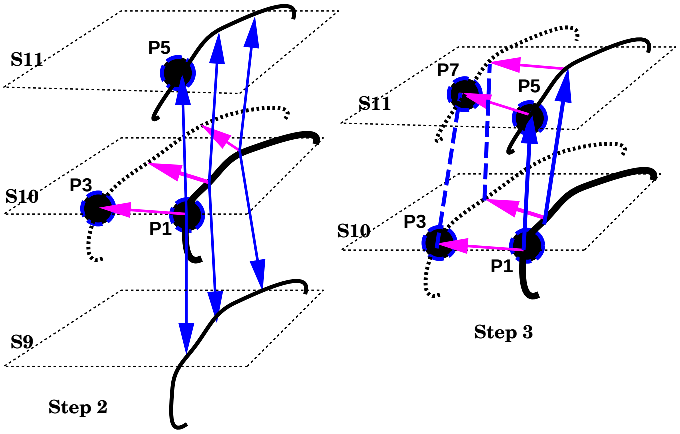


Figure 4.12: A further illustration of the mesh generation process. This figure illustrates steps 2 and 3 of the mesh generation algorithm.

metrical complexity of the endocardium, we space the interpolated surfaces to be preferentially closer to the endocardium.[10] Discretize the contour on the middle slice of the endocardium to the desired number of nodes (typically 35-45).

- Step 2A: Using the symmetric nearest neighbor algorithm, estimate correspondences between slices on the endocardial surface on a contour by contour basis starting in the middle slice. This generates a grid of connected points on the endocardium. These correspondences are shown in blue in figures 4.12 and 4.11.
- Step 2B: For the points present in the correspondence maps of step 2A, find their correspondences within each slice starting at the endocardium and moving on level at a time towards the epicardium. This generates a grid of connected points on each slice. These correspondences are shown in purple in figures 4.12 and 4.11.
- Step 3: Use transitivity of connections to complete the mesh. These connections

[10] Let s_n and s_p be the endocardial and epicardial surfaces respectively. We could generate two in-between interpolated surfaces s_1 and s_2 as $s_1 = \frac{2s_n+s_p}{3}$ and $s_2 = \frac{s_n+2s_p}{3}$. To space the surfaces preferentially closer to the endocardium we actually generate the first interpolated surface s_1 as $s_1 = \frac{3s_n+s_p}{4}$.

are drawn with dotted blue lines in figures 4.12 and 4.11. Because of the grid-like nature of the mesh, once a correspondence is established on the endocardial surface, the correspondence is fixed for the mid-wall and the epicardial surfaces as well. Consider the following example which is illustrated in figure 4.12. A point P1 on slice S10 is mapped to point P5 on slice S11 on the endocardial surface (step 2A), and point P3 on slice S10 on the first midwall surface (step 2B). Further, point P5 on slice S11 on the endocardium corresponds to point P7 on slice S11 of the first midwall surface (step 2B). By transitivity P3 also has to connect to P7. This completes the quadrilateral which forms one face of the element.

4.4 A Shape-Based Tracking Algorithm

The shape-based tracking algorithm tries to follow points on successive surfaces using a shape similarity metric. This distance is based on the difference in principal curvatures. The method was validated using implanted markers [89]. In this work, we modify the initialization step of this algorithm to take advantage of the symmetric nearest neighbor correspondence finding algorithm previously described in section 4.2.4.

The first step in this algorithm is to estimate for all points on surface s_1 their symmetric nearest neighbor, as explained in section 4.2.4. Next, for any given point p_1 on a surface s_1 at time t_1 and which has a corresponding point p_2 on surface s_2 at time t_2 as a result of the symmetric nearest neighbor estimation step we construct a plausible search window W on s_2. This search window W consists of all the points on s_2 which have a connectivity distance less than a threshold t from p_2 on s_2, i.e. $p_w \in W$ iff $d_c(p_2, p_w) < t$.

Next, a search is performed within this plausible region W on the deformed surface s_2 and the point $\hat{p}_2$ which has the local shape properties closest to those p_1 is selected. The shape properties here are captured in terms of the principal curvatures κ_1 and κ_2. This is illustrated in figure 4.13. The distance measure used is the bending energy required to bend a curved plate or surface patch to a newly deformed state. This is labeled as d_{be} and is defined as (see Shi[89]):

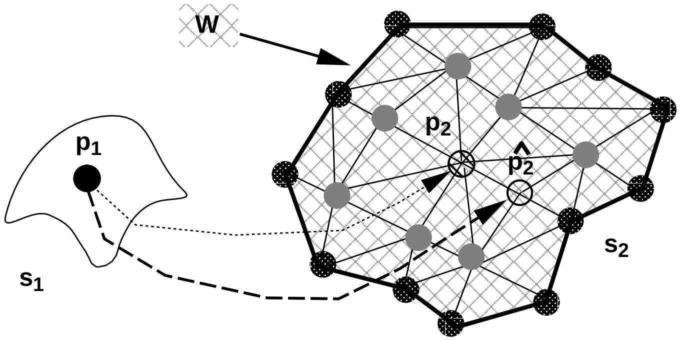


Figure 4.13: The shape-tracking algorithm. For a point p_1 on the original surface, a window W of plausible matching points on the final surface is first generated around point p_2 which is the symmetric nearest neighbor of p_1 on the deformed surface. (In this case $\forall p_w \in W : d_c(p_2, p_w) < 3$). Then the point $\hat{p}_2$ in W which has the most similar shape-properties to p_1 is selected as the candidate match point. The distance function for shape-similarity is based on the principal curvatures.

$$d_{be}(p1, p2) = \Big(\frac{(\kappa_1(p_1) - \kappa_1(p_2))^2 + (\kappa_2(p_1) - \kappa_2(p_2))^2}{2}\Big) \tag{4.13}$$

The displacement estimate vector for each point p_1, u_1^m is given by

$$u_1^m = \hat{p}_2 - p_1 \quad , \quad \hat{p}_2 = \begin{array}{c}\arg\min \\ p_2 \in W\end{array} \Big[d_{be}(p1, p2) \Big]$$

Confidence Measures in the match: The bending energy measures for all the points inside the search region W are recorded as the basis to measure the *goodness* and *uniqueness* of the matching choices. The value of the minimum bending energy in the search region between the matched points indicates the goodness of the match. Denote this value as m_g, we have the following measure for matching goodness[11]

$$m_g(p_1) = -d_{be}(p_1, \hat{p}_2) \tag{4.14}$$

On the other hand, it is desirable that the chosen matching point is a unique choice among the candidate points within the search window. Ideally, the bending energy value of the chosen point should be an outlier (much smaller value) compared to the values of the rest of the points. If we denote the mean values of the bending energy measures of all the points inside window W except the chosen point as $\bar{d}_{be}$ and the standard deviation as σ^d_{be}, we define the uniqueness measure as:

$$m_u(p_1) = \frac{d_{be}(p_1, \hat{p}_2)}{\bar{d}_{be} - \sigma^d_{be}} \tag{4.15}$$

This uniqueness measure has a high value if the bending energy of the chosen point is small compared to some smaller value (mean minus standard deviation) of the remaining bending energy measures. Combining these two measures together, we arrive at one *confidence measure* $c^m(p_1)$ for the matched point $\hat{p}_2$ of point p_1:

$$c^m(p_1) = \frac{1}{k_{1,g} + k_{2,g} m_g(p_1)} \times \frac{1}{k_{1,u} + k_{2,u} m_u(p_1)} \tag{4.16}$$

where $k_{1,g}, k_{2,g}, k_{1,u}$, and $k_{2,u}$ are scaling constants for normalization purposes. We normalize the confidences to lie in the range 0 to 1.

[11] This is the negative of the equivalent definition in Shi [89]. That definition is really a measure of badness!

Chapter 5

Continuum Mechanics Models and the Finite Element Method

This chapter is divided in three sections. In section 5.1 we examine the purely geometrical aspects of continuum mechanics methods. The focus here is the definition of the all-important concept of strain. In section 5.2 we use the concept of strain to define a method for capturing the material properties of an object in terms of a strain energy function. Finally in section 5.3 we present an overview of the finite element method which is the key numerical technique used in this work for the solution of problems involving mechanical models. It must be emphasized however that the finite element method can be used to solve other kinds of partial differential equations (see Huebner [49] for examples), though it is most often used in this context.[1]

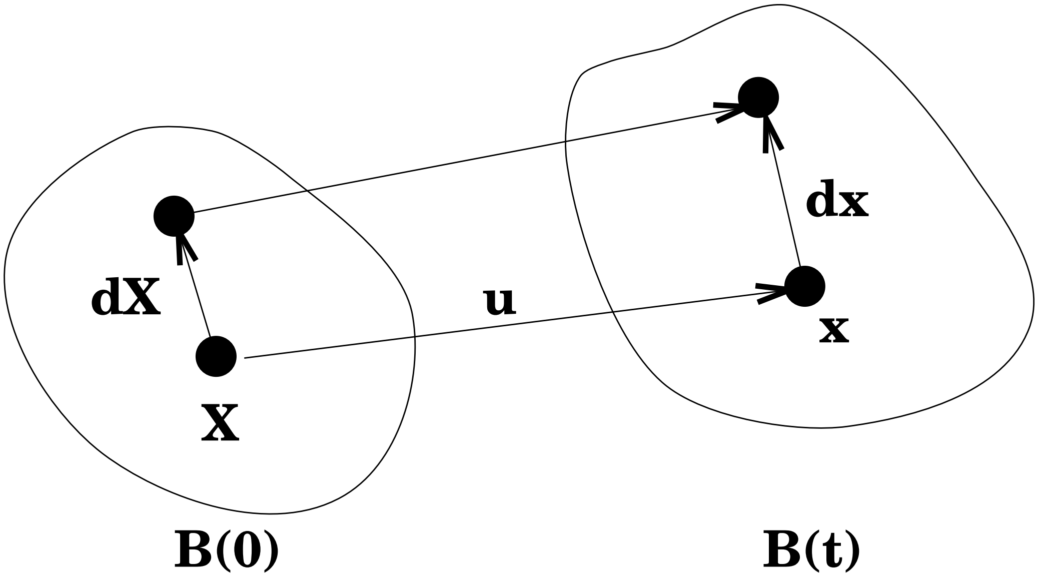


Figure 5.1: Definition of displacement

5.1 Deformations

5.1.1 The Deformation Gradient Matrix

In this section we follow the presentations in Spencer [94, chapter 6] and Hunter[75]. Consider a body $B(0)$ which after time t moves and deforms to body $B(t)$. A material particle initially located at some position X on $B(0)$ moves to a new position x on $B(t)$. If we further assume that material cannot appear or disappear there will be an one-to-one correspondence between x and x, so we can always write the path of the particle as:

$$x = x(X, t) \tag{5.1}$$

[1] A commonly used misnomer is the term 'finite element model'. There exists no such thing. The finite element method is simply a numerical procedure for solving partial differential equations whose source defines the model.

We can also define the displacement vector for this particle as

$$u(t) = x(t) - X \tag{5.2}$$

This relationship is also invertible, given x and t we can find X. If we consider two neighboring particles located at X and $X = dX$ on $B(0)$. In a new configuration $B(t)$ using equation (5.1) we can write:

$$dx = \frac{\partial x}{\partial X} dX \tag{5.3}$$

The Jacobian matrix $F(t) = \partial x(t)/\partial X$ is called *the deformation gradient matrix.* We note that by definition $F(0) = I$. Using this we can rewrite equation (5.1) more fully as:

$$\begin{aligned} dx(t) &= F(t).dX \\ F_{ij} &= \tfrac{\partial x_i}{\partial X_j} \quad , \quad F(0) = I \\ &= \tfrac{\partial u_i}{\partial X_j} + \delta_{ij} \quad , \quad u(0) = 0 \\ \delta_{ij} &= \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{5.4}$$

The mapping defined by equations (5.1)–(5.5) has two components: a rigid motion component and a change in the shape or deformation of the object. For the purposes of capturing the material behavior (to be discussed in section 5.2) we need to extract from F the component which is a function of the rigid motion and the component which is a function of the deformation.

To extract the deformation component we use the polar decomposition[96] to write F as:

$$F = \underbrace{R}_{\text{Rotation Matrix}} \times \underbrace{U}_{\text{Symmetric Matrix}} \tag{5.5}$$

The matrix R is a rotation matrix having the properties $R * R' = I, det(R) = 1$ and U is a symmetric matrix i.e. $U' = U$.

It is also useful to define the *right Cauchy-Green deformation* matrix $G = F'F$. When we apply the polar decomposition we get:

$$G = F'F = U'R'RU = U'U \tag{5.6}$$

This shows that G is independent of the rotation and is purely a function of the deformation. In the case of a pure rotation i.e. $F = R$ we find that $G = I$. This shows that G in the case of a rotation is equal to identity. We also note that G has three invariants under a coordinate transformation defined as follows:

$$\begin{aligned} I_1 &= trace(G) \\ I_2 &= \frac{1}{2}((trace(G)^2 - trace(G^2)) \\ I_3 &= det(G) \end{aligned} \tag{5.7}$$

In particular, in the case of an incompressible material $det(G) = I_3 = 1$. We next consider the important case of small deformations and rotations.

5.1.2 Small Deformations and Rotations

If the deformations and the rotations are small ($< 2-3\%$), we use the approximation[94, section 6.6]:

$$\frac{\partial u}{\partial x} \approx \frac{\partial u}{\partial X} \tag{5.8}$$

From here we can re-write $F = RU$ as:

$$F = RU = (I+\omega)(I+\epsilon) \tag{5.9}$$

Here ω is the small rotation tensor and is antisymmetric. ϵ is the small(infinitesimal) strain tensor and is symmetric. These are defined as:

$$\begin{aligned}
\omega \quad &= \tfrac{1}{2}(F-F') \quad = \begin{bmatrix} 0 & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2}-\frac{\partial u_2}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3}-\frac{\partial u_3}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial u_2}{\partial x_1}-\frac{\partial u_1}{\partial x_2}\right) & 0 & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3}-\frac{\partial u_3}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial u_3}{\partial x_1}-\frac{\partial u_1}{\partial x_3}\right) & \frac{1}{2}\left(\frac{\partial u_3}{\partial x_2}-\frac{\partial u_2}{\partial x_3}\right) & 0 \end{bmatrix} \\
\epsilon \quad &= \tfrac{1}{2}(F+F')-I = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2}+\frac{\partial u_2}{\partial x_1}\right) & \frac{1}{2}\left(\frac{\partial u_1}{\partial x_3}+\frac{\partial u_3}{\partial x_1}\right) \\ \frac{1}{2}\left(\frac{\partial u_2}{\partial x_1}+\frac{\partial u_1}{\partial x_2}\right) & \frac{\partial u_2}{\partial x_2} & \frac{1}{2}\left(\frac{\partial u_2}{\partial x_3}+\frac{\partial u_3}{\partial x_2}\right) \\ \frac{1}{2}\left(\frac{\partial u_3}{\partial x_1}+\frac{\partial u_1}{\partial x_3}\right) & \frac{1}{2}\left(\frac{\partial u_3}{\partial x_2}+\frac{\partial u_2}{\partial x_3}\right) & \frac{\partial u_3}{\partial x_3} \end{bmatrix}
\end{aligned} \tag{5.10}$$

Often, taking advantage of the symmetries these tensors are written in vector form as:

$$e = [\epsilon_{11}\ ,\ \epsilon_{22}\ ,\ \epsilon_{33}\ ,\ \epsilon_{12}\ ,\ \epsilon_{13}\ ,\ \epsilon_{23}]'\ ,\ \theta = [0\ ,\ 0\ ,\ 0\ ,\ \omega_{12}\ ,\ \omega_{13}\ ,\ \omega_{23}]'$$

This e is the classical definition for strain in infinitesimal linear elasticity[94].

5.1.3 Finite Deformations

The infinitesimal deformation measures are applicable only for very small deformations and rotations. In the case of soft-tissue deformation and specifically the left ventricle these are not applicable. Using equation (5.6) for the definition of G we can define the *Lagrangian (or Green) strain* tensor E as:

$$E = \frac{1}{2}\Big(C - I\Big) \tag{5.11}$$

The components of E become equal to zero when there is no deformation ($G = I$), and in the case of small deformations and rotations reduce to the strain tensor[2] of classical infinitesimal elasticity theory. We can also write this in component form as:

$$E_{ij} = \frac{1}{2}\Big(\sum_k \frac{\partial x_k}{\partial X_i}\frac{\partial x_k}{\partial X_j} - \delta_{ij}\Big) = \frac{1}{2}\Big(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \sum_k \frac{\partial u_k}{\partial X_i}\frac{\partial u_k}{\partial X_j}\Big) \tag{5.12}$$

5.1.4 Some Further Properties of the Strain Tensor

Given a strain tensor E_x (a 3×3 matrix) which was computed in a coordinate frame x parameterized by three unit vectors x_1, x_2, x_3 we can transform it to a coordinate frame y similarly parameterized by unit vectors y_1, y_2, y_3 as follows. First construct the 3×3 rotation matrix R. Each component of R, R_{ij} is given by the dot product of x_i and y_j, i.e. $R_{ij} =< x_i, y_j >$. This results in $R : x \mapsto y$. Using this matrix R we can write the

[2] The finite strain tensor has the form $\frac{1}{2}(F'F - I)$ as opposed to the infinitesimal strain tensor which is defined as $\frac{1}{2}(F + F') - I = \frac{1}{2}(F + F' - 2I)$. Hence the approximation involved in the infinitesimal strain tensor is $F + F' - 2I \approx F'F - I$. If we define $F = I + dF$ we can write $F'F - I = (I + dF)'(I + dF) - I = dF' + dF + dF'dF$ and $F' + F - 2I = dF + dF'$. So in making the infinitesimal approximation the assumption is that the second order term $dF'dF \approx 0$, and so can be ignored. This is easily seen from equation (5.12).

image of E_x in the y coordinate frame E_y as:

$$E_y = RE_xR' \tag{5.13}$$

We also note that the eigenvalues of E are known as the principal strains and the eigenvectors as the principal directions. These are invariant to a change of coordinate frame. The principal strains are particularly useful in the case of comparing strains produced from two sets of measurements whose relative coordinate transformation is unknown.

5.2 Material Models

So far we have restricted our description to the geometry of the deformation. In this section we extend this to account for what happens when a material deforms and relate the deformation to the change in the internal structure of the material. Before proceeding to give examples of possible material models we first note that there some theoretical guidelines which must be observed[32]. The most important ones for this work are:

1. *The axiom of objectivity*–this requires the material model to be invariant with respect to rigid motion or the spatial frame of reference.

2. *The axiom of material invariance*–this implies certain symmetry conditions dependent on the type of anisotropy of the material, and implicitly reduces the number of free parameters.

The first axiom can be satisfied by postulating an internal or strain energy function which depends on the gradient deformation matrix F only through the Green deformation tensor G, the Green strain tensor E, on in small deformation cases the infinitesimal strain tensor ϵ. The strain energy function serves as the material model. If we postulate an internal energy which is not invariant to a global rotation we arrive at the following problem. Suppose that work is needed to rotate the object clockwise. From conservation of energy principles, this energy will be returned when the object is turned counter-clockwise. We can keep turning the object counter-clockwise to get more and more energy and in this way we have created a *perpetual motion machine* and not a material model.

5.2.1 Linear Elastic Energy Functions

In this section e will be used to denote the vector form of either the Green strain tensor E or the infinitesimal strain tensor ϵ as appropriate. The simplest useful continuum model in solid mechanics is the linear elastic one. This is defined in terms of an internal energy function W which has the form:

$$W = e'Ce \tag{5.14}$$

where C is a 6×6 matrix and defines the material properties of the deforming body[3], as it relates the change in geometry (strain) to the internal energy function W. The simplest model is the isotropic linear elastic model used widely in the image analysis literature [42, 30]. In this case the matrix C takes the form:

$$C^{-1} = \frac{1}{E}\begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{bmatrix} \tag{5.15}$$

[3] This class of model is linear as it results in a linear stress-strain relationship i.e. $\sigma = C\epsilon$. We do not use stresses in this work so we will not express material models explicitly in terms of their stress-strain relationships. In this chapter, we deliberately avoid the terms ‘force’, ‘stress’ and ‘equilibrium’. These would be inappropriate as the problem we are trying to solve has no real forces as such. The use of the word ‘forces’ in related work such as Terzopoulos[101] in the context of physics-based vision may have been appropriate as the authors were not trying in any way to use real physics in their methods. In this work, since we are using *real* mechanical models to model *real* tissue properties we would only use words such as force to describe *real* forces.

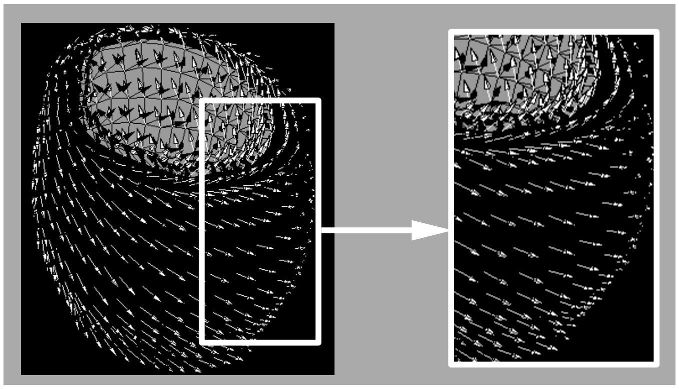


Figure 5.2: Fiber direction in the left ventricle as defined in Guccione et al [39]. More details can be found in section 7.2.2.

where E is the Young's modulus which is a measure of the stiffness of the material and ν is the Poisson's ratio which is a measure of incompressibility.

In this work, the left ventricle of the heart is specifically modeled as a transversely elastic material to account for the preferential stiffness in the fiber direction. This is an extension of the isotropic linear elastic model which allows for one of the three material axis to have a different stiffness from the other two. In this case the matrix C takes the form:

$$C^{-1} = \begin{bmatrix} \frac{1}{E_p} & \frac{-\nu_p}{E_p} & \frac{-\nu_{fp}}{E_f} & 0 & 0 & 0 \\ \frac{-\nu_p}{E_p} & \frac{1}{E_p} & \frac{-\nu_{fp}}{E_f} & 0 & 0 & 0 \\ \frac{-\nu_{fp}E_f}{E_p} & \frac{-\nu_{fp}E_f}{E_p} & \frac{1}{E_f} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{2(1+\nu_p)}{E_p} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_f} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_f} \end{bmatrix} \tag{5.16}$$

where E_f is the fiber stiffness, E_p is cross-fiber stiffness and ν_{fp}, ν_p are the corresponding Poisson's ratios and G_f is the shear modulus across fibers. ($G_f \approx E_f/(2(1+\nu_{fp}))$. If $E_f = E_p$ and $\nu_p = \nu_{fp}$ this model reduces to the more common isotropic linear elastic model. The fiber stiffness was set to be 3.5 times greater than the cross-fiber stiffness [39]. The Poisson's ratios were both set to 0.4 to model approximate incompressibility.

5.2.2 Non-Linear Energy Functions

Linear models do not capture the progressive hardening of many materials (especially soft tissue) when it is stretched. In the case of linear elastic models the effective stiffness is a constant with respect to the strain whereas in practice the stiffness increases as the strain increases.[4] Even though, in this work we use a linear model, the following summary of non-linear models is included here for the sake of completeness.

One common non-linear model in the case of isotropic incompressible materials is the Mooney-Rivlin material model[62]. In this case the internal energy function is a function only of the invariants of the right Cauchy-Green deformation matrix G (this is as a result of the axiom of material invariance) and can be written as:

$$W(I_1, I_2) = a(I_1 - 3) + b(I_2 - 3) \tag{5.17}$$

with the further constraint that the solution must satisfy $I_3 = 1$. This is often imposed as a Lagrange multiplier in an optimization framework.

In the work of Guccione and McCulloch[39] a transversely isotropic model is used for the myocardium, defined as follows:

$$\begin{aligned} W &= \frac{C}{2}(e^Q - 1) - \frac{p}{2}(I_3 - 1) \qquad (5.18) \\ Q &= b_1 E_{11}^2 + b_2(E_{22}^2 + E_{33}^2 + E_{23}^2 + E_{32}^2) + b_3(E_{12}^2 + E_{21}^2 + E_{13}^2 + E_{31}^2) \end{aligned}$$

[4] This is an effect of a transition in the process of stretching. In elastomers, at low strains, the stretching results mostly in 'uncoiling' the long polymer chain molecules which effectively results in low stiffness. At higher strains, once the chains are fully uncoiled, the stretching process is trying to extend the polymer chains themselves which gives rise to a much higher stiffness.

In this case, the model can have different stiffness along the local x direction from the one in the y and z directions. Also the incompressibility constraint is imposed by penalizing the variation of the third strain invariant I_3 from 1. Further refinements of this work, including the incorporation of active contraction and electrophysiology, can be found in Hunter[75].

5.3 The Finite Element Method

The finite element method is a numerical analysis technique for obtaining approximate solutions to a wide variety of engineering problems[49]. The key to this method is that the domain of problem is divided into small areas or volumes called *elements*. The problem is then discretized on an element by element basis and the resulting equations *assembled* to form the global solution. In this work we discretize the problems using the custom mesh generation technique described in section 4.3.

5.3.1 An Example Problem

In this section we will describe an example problem and outline how it could be solved using the finite element method. We will pose the problem in terms of an energy minimization framework where the goal is to estimate the displacement field $u(x, y, z)$ which is an optimal tradeoff between an internal energy function[5] $W(C, u)$ and approximating a noisy displacement field $u^m(x, y, z)$ in a weighted least squares sense.

We define the optimal solution displacement field u is the one that minimizes functional $P(u)$. This is defined as:

$$P(u) = \int_{vol} (\quad W(C, u) + V(u, u^m))d(vol)$$

$$W(C, u) = e(u)'Ce(u)$$

[5] Note that although W is defined as function of the strain e, as e is a function of the displacement u, W can also be written as a function of the displacement field u.

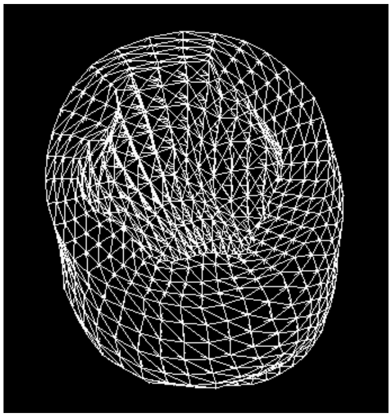


Figure 5.3: A 3D hexahedral mesh generated by interpolating and filling between the endocardial and epicardial boundaries.

$$V(u, u^m) = \alpha(u^m - u)^2$$

where $W(C, u)$ is the internal energy function defined by a strain energy function. C is the constitutive law and e is the local strain which is a function of the displacements u. $V(u, u^m)$ is the external energy term. u^m is the original (shape-tracking) displacement estimate and α is the confidence in the match.

5.3.2 Outline of the Solution Procedure

Step 1: Divide Volume into elements (tetrahedra or hexahedra) to provide the basis functions for the discretization. In figure 5.3 a myocardium is shown tessellated into hexahedral elements. (See section 4.3.)

Step 2: Discretize the problem by approximating the displacement field in each element as a linear combinations of displacements at the nodes of each element. For a hexahedral

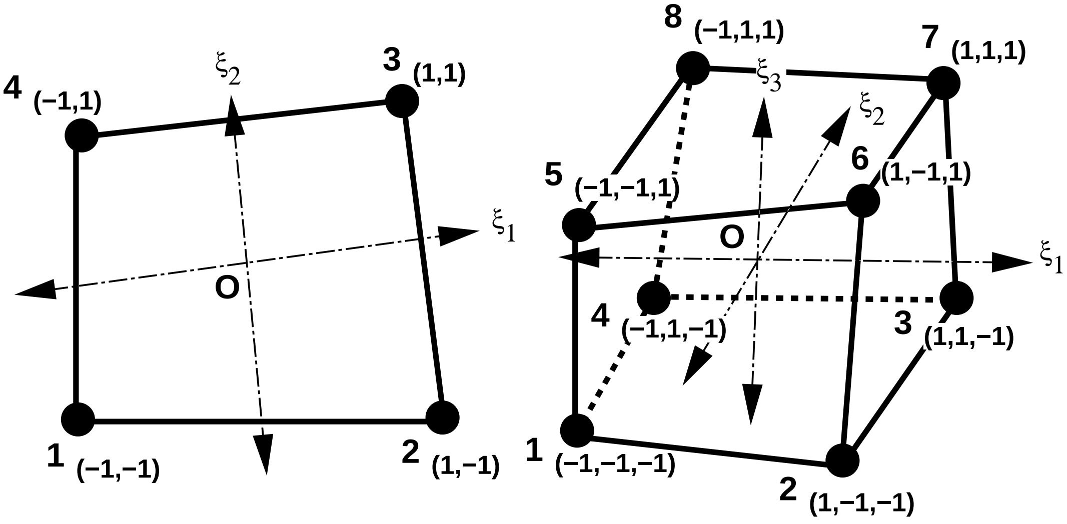


Figure 5.4: Definition of local element coordinate system ξ_i and node coordinates for the nodes of a 2D 4-node isoparametric element (left) and a 3D 8-node isoparametric element(right). For example, in the 2D case, node 1 has coordinates (-1,-1). The centroid of the element O is the origin of the element specific coordinate system. Note also that the axes are not necessarily orthogonal.

element this discretization can be expressed as:

$$u \approx \sum_{i=1}^{8} N_i u_i$$

where N_i is the interpolation shape function for node i and u_i is the displacement at node i of the element. For the isoparametric hexahedral element shown in figure 5.4, we define a local coordinate system ξ_i, and in this the shape functions N_i take the form[49, section 5.5]:

$$N_i(\xi_1, \xi_2, \xi_3) = \frac{1}{8}(1 + \xi_1\xi_{1,i})(1 + \xi_2\xi_{2,i})(1 + \xi_3\xi_{3,i}) \tag{5.19}$$

where $(\xi_{1,i}, \xi_{2,i}, \xi_{3,i})$ are the local coordinates of node i. It is easy to verify that the shape function N_i takes a value of 1 at node i, a value of $\frac{1}{8}$ at the origin and a value of 0 at all other nodes. These functions are the generalization in 3D of the linear splines of figure 3.2.

Step 3: Write down internal energy equation as the sum of the internal energy for each element:

$$W(u) = \sum_{all\ elements} [\int_{v_{el}} e'Ced(v_{el})] \tag{5.20}$$

We further note that in an element we can approximate the derivatives of u with respect to components of the global coordinate system x as follows (note that the u_i are constant in this expression):

$$\frac{\partial u}{\partial x_k} = \sum_{i=1}^{8} \frac{\partial (N_i u_i)}{\partial x_k} = \sum_{i=1}^{8} \frac{\partial N_i}{\partial x_k} u_i$$

However the shape functions N_i are expressed in terms of the local coordinate system ξ. Using the chain rule we can write:

$$\left\{ \begin{array}{c} \frac{\partial N}{\partial \xi_1} \\ \frac{\partial N}{\partial \xi_2} \\ \frac{\partial N}{\partial \xi_3} \end{array} \right\} = \left[\begin{array}{ccc} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_3}{\partial \xi_1} \\ \frac{\partial x_1}{\partial \xi_2} & \frac{\partial x_2}{\partial \xi_2} & \frac{\partial x_3}{\partial \xi_2} \\ \frac{\partial x_1}{\partial \xi_3} & \frac{\partial x_2}{\partial \xi_3} & \frac{\partial x_3}{\partial \xi_3} \end{array} \right] \times \left\{ \begin{array}{c} \frac{\partial N}{\partial x_1} \\ \frac{\partial N}{\partial x_2} \\ \frac{\partial N}{\partial x_3} \end{array} \right\} \tag{5.21}$$

or equivalently in matrix notation as $N_\xi = [J] \times N_x$.

Hence we can calculate the desired derivatives N_x from the known derivatives N_ξ by inverting the Jacobian as follows: $N_x = [J]^{-1} N_\xi$. As long as the elements do not have intersecting sides the Jacobian will remain invertible.

Note also that the derivatives of the displacement field u (i.e. $\frac{\partial u}{\partial x_k}$) are a linear function of the nodal displacements u_i. Since the infinitesimal strain tensor consists of only sums and differences of partial derivatives (see equation (5.10)) the infinitesimal strain tensor can also be expressed as a linear function of the nodal displacements.[6] This can be written in matrix form as $e = Bu$. Substituting this in equation (5.20) we get:

$$\begin{aligned} W(u) &= \textstyle\sum_{all\ elements} \quad U^{e\prime}\Big[\int_{v_{el}} B'CBd(v_{el})\Big]U^e \\ &= \textstyle\sum_{all\ elements} \quad U^{e\prime}[K^e]U^e \end{aligned}$$

where K^{e7} is the element stiffness matrix[8], and U^e is a vector obtained by concatenating all the displacements of the nodes of the element i.e. :

$$U^e = [u_{1,x}, u_{1,y}, u_{1,z}, \ldots, u_{8,x}, u_{8,y}, u_{8,z}]$$

where $u_i = (u_{i,x}, u_{i,y}, u_{i,z})$ is the displacement of node i.

[6] The finite strain deformation case is non-linear and does not allow for this simplification. The subsequent expressions are so complicated that it makes the material beyond the scope of this brief overview. The reader is referred to Bathe [9].

[7] The integration is carried out using Gaussian quadrature [49].

[8] Each component of K^e indicates the 'stiffness' between any two nodes. One could in some sense think of K^e_{14} as the stiffness of a spring connecting the x-directions of local nodes 1 and 2. (This '2' is *not* a typo. The first three rows of K^e correspond to the components of the displacement of node 1, the second three to the displacement of node 2 etc. See the definition of U^e.)

Step 4: Rewrite the internal energy function in matrix form. First, we define the global displacement vector U as:

$$U = [u_{1,x}, u_{1,y}, u_{1,z}, u_{2,x}, u_{2,y}, u_{2,z}, \ldots, u_{n,x}, u_{n,y}, u_{n,z}]' \tag{5.22}$$

where n is the total number of nodes for the solid. We also define the global stiffness matrix K as the assembly of all the local element stiffness matrices K^e as:

$$K = \sum_{all\ elements} \mathcal{I}(K^e) \tag{5.23}$$

where $\mathcal{I}$ is the re-indexing function. This takes an element K^e_{ij} and adds it to the element K_{kl}, where k and l are the global node numbers of local nodes i and j.[9]

The internal energy can now be written as $W(U) = U'KU$.

Step 5: Write down the external energy function as a weighted least squares term:

$$V(u) = \sum_{i=1}^{n} \alpha_i (u^e_i - u_i)^2$$

If there is no initial displacement estimate for a given node j set $\alpha_j = 0$.

[9] Within an element the nodes are always numbered from 1 to 8. However this is a local index (shorthand) to the global node numbers. When the global matrix is assembled the local indices (1 to 8) need to be converted back to the global indices (e.g. 1 to n). K^e has dimensions 24×24 and K has dimensions $3n \times 3n$. K^e_{14}, which is the stiffness between the x-directions of local nodes 1 and 2 would be part of K_{kl} where $k = 3(a-1)+1$ and a is the global index of local node 1 and $l = 3(b-1)+1$, where b is the global index of local node 2. Since nodes appear in more than one element the final value of K_{kl} is likely to be the sum of a number of local K^e_{ij}'s.

Step 6: Rewrite external energy in a matrix form: We define the global initial displacement vector U^m in the same way as U above (see equation (5.22)) and the global confidence matrix A to be a diagonal matrix with the confidence values for each displacement (α_i) forming the elements of the leading diagonal as follows:

$$A = \begin{bmatrix} a_1 & & & & & & \\ & a_1 & & & & & \\ & & a_1 & & & & \\ & & & \dots & & & \\ & & & & a_n & & \\ & & & & & a_n & \\ & & & & & & a_n \end{bmatrix} \tag{5.24}$$

The external energy can be rewritten as $V(U) = (U^m - U)' A (U^m - U)$.

Step 7: Form total potential energy equation $P(U) = W(U) - V(U)$.

Step 8: Solve for U. Differentiate $P(U)$ w.r.t U and set to 0. This results in the final equation

$$KU = A(U^m - U)$$

This is then solved for U using sparse matrix methods.[10] U represents the values of u at the nodes, and by means of the finite element approximation ($u \approx \sum_{i=1}^{8} N_i u_i$) we can compute the resulting values of the displacement field u anywhere in the volume.

[10] In the case of finite deformations we end up with an expression of the form $K(U) = A(U^m - U)$ which is solved iteratively.

Chapter 6

Modeling the Displacement Field

In this chapter we expand on material presented in section 2.4 regarding the use of modeling for interpolation and smoothing. In section 6.1 we present the general regularization framework and discuss a probabilistic formulation for this as well as some generic implications. Next in the section 6.2 we focus on the common first-order regularization function, which we examine in some detail. We also briefly examine the thin-plate functional. In section 6.3 we consider the use of the linear elastic functional and discuss the problems associated with this as well as various possible solutions. Finally in section 6.4 we describe a possible extension to the elastic model paradigm, the *Active Elastic Model*.

6.1 The General Regularization Framework

6.1.1 The Energy Minimization Framework

In this section we describe a framework in which the goal is to estimate a displacement field u which approximates another displacement field u^m. We will assume that u^m is derived from some image-based algorithm, such as the shape-based tracking algorithm, where the relationships between different displacements are not modeled. We simplify the approximation problem to be a least-squares fit of u to u^m subject to some constraints.

This takes the form:

$$\hat{u} = \begin{matrix} \arg\min \\ u \end{matrix} \left(\int_V W(\alpha, u, x) + c(x)|u^m(x) - u(x)|^2 dv \right) \tag{6.1}$$

where:

- $u(x) = (u_1, u_2, u_3)$ is the vector valued displacement field defined in the region of interest V and x is the position in space.
- $u^m(x) = (u_1^m, u_2^m, u_3^m)$.
- $c(x)$ is the spatially varying confidence in the measurements u^m.
- $W(\alpha, u, x)$ is a positive definite functional which defines the approximation strategy and is solely a function of u, a parameter vector α and the spatial position x.

This is commonly known as the regularization approach which was already described in section 2.4. $W(\alpha, u, x)$ is known as the *stabilization functional.* In certain cases the input displacement field u^m is *sparse* and is defined only on a finite number (P) of points p within V. In this case the overall functional takes the form:

$$\hat{u} = \begin{matrix} \arg\min \\ u \end{matrix} \left(\int_V W(\alpha, u, x) dv + \sum_{i=1}^{P} c(p_i)|u^m(p_i) - u(p_i)|^2 \right) \tag{6.2}$$

6.1.2 A Probabilistic Interpretation

We now derive a probabilistic interpretation of the energy minimization framework. In this setup again we aim to estimate the output displacements u from a set of measurements u^m. We further assume that we are given the measurement probability density function $p(u^m|u)$, which also corresponds to the noise model for the measurements, and the prior probability density function for u, $p(u)$.[1] We pose this as a Bayesian a-posteriori

[1] We will not define the basic terms of probability here, they can be found in standard textbooks such as Papoulis [79].

estimation problem. Within this framework, the solution $\hat{u}$ is the u that maximizes the posterior probability density $p(u|u^m)$. Using Bayes' rule we can write the posterior probability as:

$$\hat{u} = \begin{array}{c}\arg\max\\ u\end{array} \left\{ p(u|u^m) = \frac{p(u,u^m)}{p(u^m)} = \frac{p(u^m|u)p(u)}{p(u^m)} \right\} \tag{6.3}$$

First we note that $p(u^m)$ is a constant once the measurements have been made and can therefore be ignored in the maximization process. We can re-write the above expression by taking logarithms to arrive at:

$$\hat{u} = \begin{array}{c}\arg\max\\ u\end{array} \Big(\log p(u) + \log p(u^m|u) \Big) \tag{6.4}$$

This expression is now in the same general form as equation (6.1). As previously demonstrated by Geman and Geman[38] and applied to medical image analysis problems (e.g. Christensen [16], Gee [37]), there is a correspondence between an internal energy function and a Gibbs probability density function. Given an energy function $W(\alpha, u, x)$ we can write an equivalent prior probability density function $p(u)$ (see equation (6.3)) of the Gibbs form[38]):

$$\begin{aligned} p(u) &= k_1 \exp(-W(\alpha, u, x)) \\ \log(p(u)) &= \log(k_1) - W(\alpha, u, x) \end{aligned} \tag{6.5}$$

where k_1 is a normalization constant.

Next we define the noise $n = u - u^m$. Then we can model the noise probabilistically, using a multivariate Gaussian distribution, as:

$$p(n) = k_2 \exp(\frac{-n'\Sigma^{-1}n}{2})$$

$$\log p(n) \quad = \quad \log k_2 - \frac{1}{2} n' \Sigma^{-1} n \tag{6.6}$$

where k_2 is also a normalization constant and Σ is the covariance matrix which in this case can be assumed to be diagonal. The mean of the noise is assumed to be equal to zero. Substituting for n in this expression we get:

$$\log p(u^m|u) = k_2 - \frac{1}{2}(u^m - u)' \Sigma^{-1} (u^m - u) \tag{6.7}$$

By an appropriate choice of Σ the second term can be mapped to the data adherence term of equation (6.2). In this case Σ^{-1} will be a diagonal matrix with values $c(p_i)$ on the leading diagonal very similar to the matrix A of equation (5.24).[2]

6.1.3 Advantages of the Probabilistic Interpretation

In the soft tissue deformation problem there are usually two types of information: (i) the image derived data which is corrupted by noise and (ii) the material properties of the soft tissue.

The data term is best modeled probabilistically in order to allow for the construction of a proper noise model. Here we can use ideas from the field of Digital Signal Processing (see for example Openheim and Schafer[74]). The material term however is best defined in terms of a continuum mechanical model. The ability to generate an equivalent probability density function for an internal energy function, as was done in equation (6.5), allows us to take a continuum mechanics model defined in terms of an internal or strain energy function and generate a probability density function which can then be used together with the probabilistic noise model within a Bayesian Estimation framework.

[2] This is very similar to the way the classical least squares problem is converted into a Bayesian estimation problem by assuming a Gaussian noise model. The advantage in both cases is that this generalization allows for more complicated models for the noise to be introduced more cleanly.

6.1.4 The Problem of Different Units

There is one fundamental problem with the probabilistic framework, which is also present but less obvious in the energy minimization framework. This is the problem of 'different units'. This problem arises because the model stiffness is measured in different units from the noise variance. It is best explained by means of an example.

Let as assume for the moment that $W = e(u)'Ce(u)$ which is the linear elastic model defined in equation (5.14) and the noise model used is model of equation (6.6). When these are substituted into equation (6.4) we get (ignoring the constant terms k_1 and k_2):

$$\hat{u} = \underset{u}{\arg\max} - \Big(e(u)'Ce(u) + \frac{1}{2}(u^m - u)'\Sigma^{-1}(u^m - u)\Big) \tag{6.8}$$

Given the fact that the u^m's are constant and that u, and hence the $e(u)$'s, are unknowns, the user controlled terms are C and Σ. C defines the mechanical model and Σ^{-1} the inverse covariance. We can write both of these matrices in this general form (using the $n \times n$ matrix M to be either C or Σ^{-1}) as:

$$M = \begin{bmatrix} M_{11} & \dots & M_{1n} \\ \dots & & \dots \\ M_{n1} & \dots & M_{nn} \end{bmatrix} = M_{max}[\bar{M}] \quad , \quad [\bar{M}] = \begin{bmatrix} \frac{M_{11}}{M_{max}} & \dots & \frac{M_{1n}}{M_{max}} \\ \dots & & \dots \\ \frac{M_{n1}}{M_{max}} & \dots & \frac{M_{nn}}{M_{max}} \end{bmatrix} \tag{6.9}$$

where M_{max} is the maximum value of M. In the case of the material matrix C, C_{max} would the highest value of the stiffness or the Young's Modulus, whereas in the case of the Covariance matrix Σ^{-1}, Σ^{-1}_{max} would be the smallest variance, or the highest confidence in any of the measurements. We can now rewrite equation (6.8) as:

$$\hat{u} = \underset{u}{\arg\max} - \Big(C_{max}e(u)'[\bar{C}]e(u) + \frac{\Sigma^{-1}_{max}}{2}(u^m - u)'[\bar{\Sigma}^{-1}](u^m - u)\Big)$$

$$\hat{u} = \underset{u}{\arg\max} - \Big(\underbrace{e(u)'[\bar{C}]e(u)}_{\text{dimensionless}} + \frac{\Sigma_{max}^{-1}}{2C_{max}} \underbrace{(u^m - u)'[\bar{\Sigma}^{-1}](u^m - u)}_{\text{dimensionless}} \Big) \tag{6.10}$$

At this point, it is clear that the absolute values of C_{max} and Σ_{max}^{-1} enter into the functional only through their ratio $\frac{\Sigma_{max}^{-1}}{C_{max}}$. Given that the rest of the expressions in equation (6.10) are dimensionless[3] for equation (6.10) to add up from a dimensionality viewpoint we need to convert this ratio $\frac{\Sigma_{max}^{-1}}{C_{max}}$ in order to also make it dimensionless.[4] This is done by multiplying by a scaling constant k_{sc} of the appropriate units i.e.

$$\frac{\Sigma_{max}^{-1}}{C_{max}} \mapsto \frac{k_{sc}\Sigma_{max}^{-1}}{C_{max}} \tag{6.11}$$

From a dimensionality viewpoint the value of the scaling constant k_{sc} is completely arbitrary.[5] This value can be interpreted as defining in some sense the ratio of the relative confidences in the data *as a whole* and the model *as a whole*. One method for setting the value of this constant can be found in section 7.2.3.

6.1.5 Soft Tissue Objects as Markov Random Fields

In using the Gibbs form (equation (6.5)) we have modeled the displacement field of the solid probabilistically as a Markov Random Field, an example of this is shown in fig-

[3] The term 'dimensionless' is used to describe a quantity that is a pure number and has no associated units. A dimensionless quantity will have the same value regardless of the system of units used in its calculation. For example the ratio of two lengths will the same regardless of whether the lengths are measured in meters or in feet.

[4] C_{max} is measured in Pascals and Σ_{max} in voxels. Hence their ratio will not be dimensionless.

[5] Consider the following example. We are trying to optimize the design criteria for a new computer and two criteria are speed S in MHz and cost C in dollars. We proceed to optimize the criterion $\alpha S + \beta C$. The value of the ratio $\frac{\alpha}{\beta}$ which will determine the optimal S and C is completely arbitrary as S and C have different units. It is up to the designer/salesperson to select the value that matches some other external criterion.

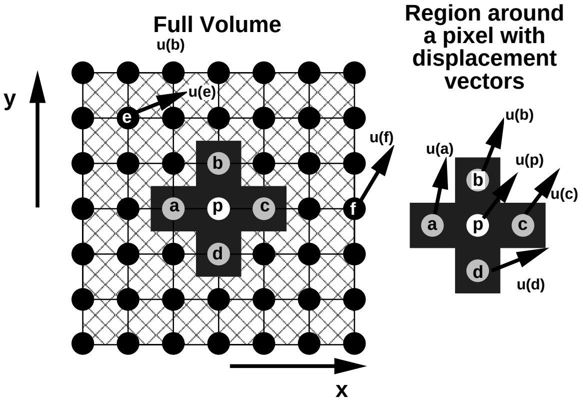


Figure 6.1: Example of an object discretized by particles shown as black circles. If the displacement field is modeled as a first-order Markov Random Field (MRF) the displacement of a specific particle p depends only on external data and the displacements of its immediate neighbors a, b, c, d.

ure 6.1. The Markov random field (MRF) then can be thought of as the probabilistic analog of the continuum mechanical model. There are two interesting similarities: (i) Both can be defined using energy functions and (ii) the energy functions at any given point are functions only of the values of that point and its immediate neighbors. In the case of the MRF point (ii) comes from the fact that the the Gibbs probability density function is often defined on first and/or second order cliques which are very local neighborhoods of the point. So if the displacement field is modeled as a MRF, the probability of the displacement of a given point p effectively only depends on the displacement of its neighbors. In the case of the mechanical model described using a strain energy function, the value of the internal energy function, which via exponentiation in equation (6.5) becomes the probability density function, at a given point depends only on the local strains. These local strains are only dependent on the displacements of the neighbors of the point and not on the displacements of the whole volume.

6.2 A First-Order Regularization Functional

In this section we begin by examining the most common regularization functional, first proposed by Horn and Schunk[47, 46] and subsequently used by many others with various modifications [86, 24, 109, 93]. In this case $W(\alpha, u, x)$ takes the form:

$$W(\alpha, u, x) = \sum_{i,j} \alpha_{ij} \Big(\frac{\partial u_i}{\partial x_j} \Big)^2 \tag{6.12}$$

which tries to enforce smoothness by penalizing all first order derivatives, hence the name. The main motivation for its use is the assumption that it makes very weak and generic assumptions about the underlying material properties. We will show this statement to be false later in this section. A perennial problem with this model is the setting of the values of the constants α_{ij}, for which there is no good criterion.

6.2.1 The Two Dimensional Dense Case

In this case, for simplicity, we will consider the two dimensional dense case. Here we assume that u^m is defined over the whole volume of the object V. We further set all the weighting constants α_{ij} equal to a single constant λ. We substitute for this W in equation (6.1) to obtain:

$$\begin{aligned} \hat{u} = \quad \underset{u}{\arg\min} \Big(\quad & \int_V \lambda \Big(\Big(\frac{\partial u_1}{\partial x_1} \Big)^2 + \Big(\frac{\partial u_1}{\partial x_2} \Big)^2 + \Big(\frac{\partial u_2}{\partial x_1} \Big)^2 + \Big(\frac{\partial u_2}{\partial x_2} \Big)^2 \Big) \\ & + (u_1 - u_1^m)^2 + (u_2 - u_2^m)^2 \quad dx_1 dx_2 \Big) \end{aligned} \tag{6.13}$$

This can be divided into two functionals one for each component of $\hat{u}$. Since the two functionals will have same form, we consider only the first component $\hat{u}_1$. In this case we have:

$$\hat{u}_1 = \underset{u}{\arg\min} \Big(\int_V \lambda \Big(\Big(\frac{\partial u_1}{\partial x_1}\Big)^2 + \Big(\frac{\partial u_1}{\partial x_2}\Big)^2 \Big) + (u_1 - u_1^m)^2 \;\; dx_1 dx_2 \Big) \tag{6.14}$$

A Frequency Domain Interpretation Taking the Fourier transform ($\mathcal{F} : (x_1, x_2) \mapsto \eta_1, \eta_2$) and using the the capital letters signify the function in the transform domain i.e. $U = \mathcal{F}(u), U^m = \mathcal{F}(u^m)$ etc.) results in:

$$\hat{U}_1 = \underset{U}{\arg\min} \int_\eta \lambda \Big((\eta_1 U)^2 + (\eta_2 U)^2 \Big)^2 + (U - U^m)^2 \;\; d\eta_1 d\eta_2$$

Using calculus of Variations we 'differentiate' this functional with respect to U to get

$$\hat{U}_1 = \frac{U^m}{1 + \lambda(\eta_1^2 + \eta_2^2)}$$

which has the same basic form as a spatial low-pass filter with λ controlling the cut-off frequency. Thus this first order regularization model can be seen to be a generalization of the low pass filter.

Limiting Case–The Translational Model: In the limiting case as $\lambda \to \infty$ this reduces to taking the D.C. term of U^m which makes u^m a constant over the whole object. This is a complicated way of deriving the translational model for the displacements which has all the derivatives equal to zero. In this case we can rewrite equation (6.14) to take the form:

$$\begin{aligned} \hat{u}_1 \quad &= \quad \underset{u}{\arg\min} \Big(\int_V (u_1 - u_1^m)^2 \;\; dx_1 dx_2 \Big) \\ &\quad \text{subject to: } \Big(\frac{\partial u_1}{\partial x_1}\Big)^2 + \Big(\frac{\partial u_1}{\partial x_2}\Big)^2 = 0 \end{aligned} \tag{6.15}$$

This effectively defines u_1 to be a constant k_1. The problem is reduced to finding the k_1 that minimizes the functional. In this case k_1 will be the spatial average of u_1^m.

6.2.2 Relationship with Infinitesimal Linear Elasticity

The linear elastic model was defined to have the form $W = e'Ce$ in equation (5.14). We note that for the infinitesimal strain case, we had defined the strain tensor ϵ and the small rotation tensor ω as (see equation (5.10)).

$$\epsilon_{ij} = \frac{1}{2}\Big(\frac{\partial x_i}{\partial u_j} + \frac{\partial u_j}{\partial x_i}\Big) \tag{6.16}$$

$$\omega_{ij} = \frac{1}{2}\Big(\frac{\partial x_i}{\partial u_j} - \frac{\partial u_j}{\partial x_i}\Big) \tag{6.17}$$

We further note that $\epsilon_{ij} + \omega_{ij} = \frac{\partial x_i}{\partial u_j}$. This allows us the rewrite the first order regularization functional in terms of the strain and rotation tensor as;

$$W(\alpha, u, x) = \sum_{i,j} \alpha_{ij}(\epsilon_{ij} + \omega_{ij})^2 \tag{6.18}$$

The first non-trivial observation that can be made by looking at equation (6.18), is that the first order regularization model implicitly assumes small deformations and rotations, as it is solely a function of the infinitesimal deformation and rotation tensors. More importantly however, as it is a function of ω, this functional is *not* invariant to a global rotation (even allowing for the small rotation case). In this case it violates the axiom of objectivity, (see section 5.2.) This means that *no* real material could possibly behave in this way. Further we contradict the desired underlying assumption in the use of this model, that it makes weak and generic assumptions for the material properties. In fact this model makes assumptions so strong that no possible material could behave

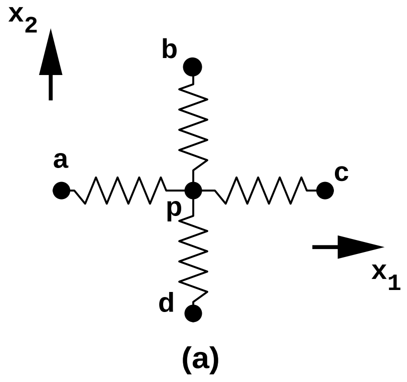


Figure 6.2: Example of an object described by a set of springs connecting neighboring points. (a) Four elements of a simple solid. (b) An element of the simple solid and (c) An element of a more complicated solid model.

this way.[6]

6.2.3 The Discrete Spring Model

One way to make the first-order regularization model invariant to rotations is to set $\alpha_{ij} = 0$ when $i \neq j$. This results in the so-called 'spring-model' which is illustrated in figure 6.2 (a). This model tries to describe the solid as a discrete set of point masses connected by springs. Alternatively, and more frequently, the same stabilizing functional is derived from the local internal energy function of the springs. To further simplify this we will only consider half the setup as shown in figure 6.2(b). Assuming constant stiffness k for all springs, and small deformations we can write this internal energy function as:

$$W(\alpha, k, u) = \alpha k\Big((u_1(c) - u_1(p))^2 + (u_2(b) - u_2(p))^2\Big) \tag{6.19}$$

[6] The fact that reasonable results have often been obtained using this first order regularization model probably has to do with the quality and density of the input data u^m. Given perfect data *no* model is needed, and given very good data, even a poor model will do a reasonable job.

Next we note that $\frac{\partial u_1}{\partial x_1} = \frac{u_1(c)-u_2(p)}{l}$ and $\frac{\partial u_2}{\partial x_2} = \frac{(u_2(b)-u_2(p)}{l}$, where l is the length of the springs. Substituting for these we get:

$$W(\alpha, k, u) = \alpha k l^2 \Big(\Big(\frac{\partial u_1}{\partial x_1}\Big)^2 + \Big(\frac{\partial u_2}{\partial x_2}\Big)^2 \Big) = \lambda \Big(\Big(\frac{\partial u_1}{\partial x_1}\Big)^2 + \Big(\frac{\partial u_2}{\partial x_2}\Big)^2 \Big) \tag{6.20}$$

This can be recognized is a form of the first order regularization functional of equation (6.12), with all the constants $\alpha_{ij}, i \neq j$ equal to zero. Further we note that using the second half of figure 6.2(a) will result in another expression of the same form and the two can be added to yield the final expression.

This model now is a simplification of an infinitesimal isotropic linear elastic model (see equation (5.15)), with the Poisson's ratio $\nu = 0$. This implies that shearing is not penalized. One way to fix this is to add diagonal springs as shown in figure 6.2(c). However at this point it is probably easier to abandon this discrete model and go to the full continuum model.

6.2.4 A Second-Order Regularization Functional and the Affine Model

Another common model is the second order regularization functional, which in two dimensions has the form:

$$\begin{aligned} \hat{u} = \quad \underset{u}{\arg\min} \Big(\quad & \int_V \alpha \Big(\Big(\frac{\partial^2 u}{\partial x^2}\Big)^2 + \Big(\frac{\partial^2 u}{\partial y^2}\Big)^2 + \Big(\frac{\partial^2 u}{\partial x \partial x_2}\Big)^2 + \\ & \Big(\frac{\partial^2 u_2}{\partial x^2}\Big)^2 + \Big(\frac{\partial^2 u_2}{\partial y^2}\Big)^2 + \Big(\frac{\partial^2 u_2}{\partial x \partial x_2}\Big)^2 \Big) \\ & + (u_1 - u_1^m)^2 + (u_2 - u_2^m)^2 \quad dx_1 dx_2 \Big) \end{aligned} \tag{6.21}$$

The solution to this takes the form known as the 'thin-plate' spline as used by Bookstein and others[12]. It is again interesting to note the limiting case where $\alpha \to \infty$.

In this case u and v take the form:

$$u_1 = a_1 x_1 + b_1 x_1 + c_1 \ , \ u_2 = a_2 x_1 + b_2 x_2 + c_2 \tag{6.22}$$

which is the affine mapping. So if $\alpha = \infty$ the process of solving equation (6.21) is reduced to estimating a parametric form of the displacement as defined by equation (6.22), using a straightforward least-squares approach.

This model, unlike the first order regularization model, is invariant to rigid rotation and hence satisfies the axiom of objectivity. It is, however, also invariant to an affine transformation which means that there is some deformation for which there is no penalty, as the affine transformation can also change the shape of the object. This is a problem in real tissue we do not have energy free deformations.[7]

6.3 The Use and Abuse of Linear Elasticity

The isotropic infinitesimal linear elastic model was most likely introduced into the medical image analysis literature as a means of avoiding the arbitrariness of setting parameters for the generic first-order regularization model. The isotropic linear elastic model has two[8] parameters, the Young's modulus and the Poisson's ratio. Moreover by virtue of the observations of section 6.1.4, the absolute value of the Young's modulus is not important, only its ratio to the highest data confidence is important.

There are two fundamental problems with the use of this model: (i) the obvious restriction to small deformations, and (ii) a bias towards no deformation. While problem (i) is important and easy to observe from the name of the model[9], it is (ii) that constitutes

[7] The exception is the case of actively deforming tissue, see section 6.4.

[8] Compare this with the possible *nine* parameters in the generic first-order regularizer of equation (6.12). Even though these nine parameters can all be set to be equal, hence reducing the number to one, there is no *principled reason* for doing so.

[9] This is also easily solved by using a finite strain formulation and perhaps also a non-linear elastic

the bigger problem. Often, given poor performance, there have been solutions proposed with problem (i) in mind (such as the fluid model) whereas the real problem was problem (ii). Also, even when problem (ii) was observed[31] the solutions were ad-hoc.

The problem of bias The easiest way to see the bias problem is the following: Since the elastic model penalizes all deformations, any estimation framework which uses it as a prior model or internal energy model as defined in equations (6.4) and (6.1) will underestimate the actual deformation. The linear elastic model can be thought of as a prior probability density function on the strain with zero mean and variance proportional to the reciprocal of the Young's modulus.

When the linear elastic model is used to regularize estimates of myocardial deformation (with strains of the order of $20 - 30\%$) this causes serious problems.

In some respects the thin-plate spline model of section 6.2.4 has an advantage here in that it penalizes the deviation from an affine transformation and not the total transformation. If most of the deformation can be captured by an affine model this would effectively only generate a bias in that component of the deformation left over after the affine transformation. This is probably why it is successfully employed in many brain registration problems.

A number of methods have been proposed to implicitly deal with this problem, we discuss these next, but note that none of these has dealt with the cause of the problems, they are in sense trying to limit, with varying degrees of success, its effects.

6.3.1 Zero Stiffness

One approach by Park[80] eliminates the elastic model altogether and provides some noise reduction by temporal filtering. While this eliminates the problems associated with bias it also forfeits all the usefulness of exploiting the spatial relationships between different points in the model. The method is successful in part because the input data are very clean.

model.

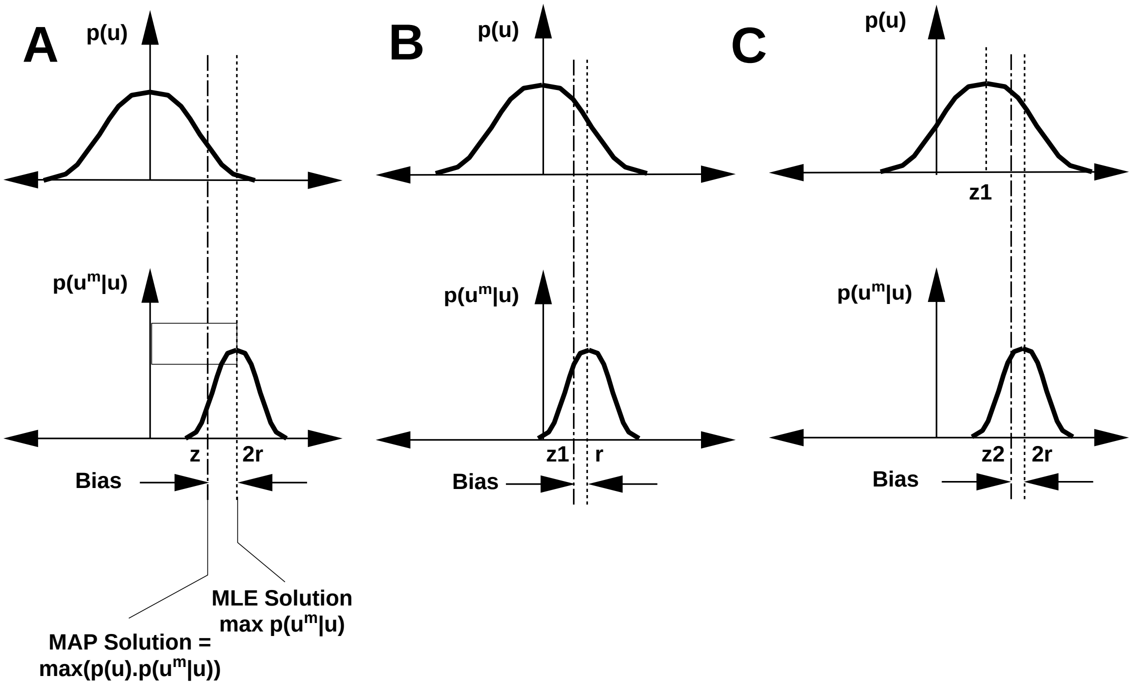


Figure 6.3: Example of bias in the maximum a-posteriori approach. The bias is the difference between the true solution and the actual solution. Note that the bias is more pronounced as the true solution (the maximum of $p(u^m|u)$) deviates more from the prior mean (which is zero).

6.3.2 Bias Correction

This is essentially the approach we use in chapter 7. If at the end of a step there is some known information about the position of a point, (that is should lie on a surface or line), the point gets mapped to this surface via a 'nearest' neighbor method. This eliminates bias in some directions but not others (i.e. bias is corrected perpendicular to the surface but not along the surface).

6.3.3 The History-Free Approach

In this case the problem is divided into a number of small steps and at the start of each step the strain is assumed to be zero.[10] By splitting the problem into many small problems the effect of the bias is reduced, as in each step the deformation is small. Consider the example shown in figure 6.3. In case (A) the whole measurement $2r$ is applied at once resulting in a large bias $2r - z$. In the second case (figure 6.3) the measurement is applied incrementally in two steps B and C. In step B we apply a displacement r and we get an output $z1$. If the process does not remember the past, for the second step C, though we apply a displacement $2r$, in practice this is the same as $2r - z1$ as the new position of the solid is taken to be the rest state. So in this case $p(u)$ has a mean of $z1$. This reduces the bias in the second step resulting in a better overall estimate and a bias reduction.

The incremental approach substantially reduces the bias, but as the history of the deformation is lost at each step it cannot capture issues such as relative hardening of parts of the model. Hence in this way we cannot capture aspects of real materials such as progressive hardening with increased strain (using non-linear elastic models) as at each step the strain is assumed to be zero. We also note that this is the approach effectively used in deformable model segmentation and optical flow estimation where at each step the model is assumed to be deformation free.

6.3.4 Fluid Model

This is essentially the limiting case of the history free approach. In the work of Christensen[17] it takes the differential form:

$$\mu \nabla^2 v + (\lambda + \mu) \nabla (\nabla . v) = F \tag{6.23}$$

[10] This is part of the solution used in chapter 7 of this thesis in the estimation of left ventricular deformation.

where F is the image derived forcing function and v is the local velocity vector related to the displacement u as[17]:

$$v = \frac{du}{dt} = \frac{\partial u}{\partial t} + \sum_{i=1}^{3} v_i \frac{\partial u}{\partial x_i} \tag{6.24}$$

where in this definition u and v are defined is a Eulerian Framework, as opposed to the standard Lagrangian framework used in solid mechanics.[11]

The isotropic linear elasticity model can also be written in differential form by differentiating the energy functional posed in equation (6.1) and generating a force F by grouping together all external displacements u^m. This takes the form (as derived in Christensen [18]):

$$\mu \nabla^2 u(X) + (\mu + \lambda)\nabla(\nabla . u(X)) = F \tag{6.25}$$

where λ and μ are the Lamè constants which are defined in terms of the Young's modulus E and the Poisson's ratio ν as[49]:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)}$$

If we compare equations (6.23) with (6.25) we see that they are essentially the same, with the one being in terms of the velocity v and the other in terms of the displacement u. The fluid model can be seen to be the limiting case of the history free approach of the previous section (section 6.3.3) as the step size goes to zero. First note that

[11] In the Lagrangian formulation the vector u is attached to the particle originally at location X whereas in the Eulerian formulation u is the displacement of the particle currently at this position. As Strang points out, in the context of Fluid Mechanics[96]: "The fluid is flowing past Euler, who sits at a point and watches Lagrange go by."

$v = \lim_{\delta t \to 0} \frac{u(t+\delta t)-u(t)}{\delta t}$. Then if the problem were solved using the history-free approach and a large number of steps the effective displacement u would approach the velocity as the step size gets smaller.

The 'fluid-model' approach has the advantage of explicitly stating its assumptions properly and possibly some numerical advantages. However it does not essentially change the solution that would have been obtained given the history-free approach and a linear elastic solid.

6.4 Active Elastic Models

6.4.1 Problems With Passive Models

The rational for the use of biomechanical models in the recovery of soft tissue deformation from medical images, is that they capture something of the real material properties of the object. If the object though, as in the case of the left ventricle, is deforming actively, a passive model such as those discussed earlier in this chapter has severe bias problems.

We can try to deal with the effects of the bias problems in a number of ways as discussed in the previous section, but none of these methods can provide the following properties:

1. Incorporate a prior model for the deformation which preferentially penalizes some deformations but not others.

2. Include the ability to model the deformation from start to finish and at any time in the process penalize the deformation from the original state.

Regarding the first point, most elastic models will penalize deviations from rigid motion, that is all deformations. Models based on the thin-plate spline (see section 6.2.4) penalize any deviations from an affine deformation. This would be a good choice if we knew that the true deformation was on average affine, but this is not very likely in arbitrary soft tissue deformation.

The second point would allow the imposition of constraints such as fiber hardening or locking. In the case of the left ventricle (and generally where elastomers are concerned) a material will become rigid in certain directions after a certain amount of deformation. Any attempt to deform it further in this direction will result in a twisting motion as the deformation has to be captured in a direction other than the one that has locked.

6.4.2 A Proposed Extension

One possible correction for the elastic models is the adjustment of the model for non-zero bias. Consider the following generalization of the standard linear elastic model, which we will label the *active elastic model*:

$$W = (e - e^a)'C(e - e^a) \tag{6.26}$$

This is the equivalent of having a non-zero mean prior probability density for the deformation. The strain e is divided into two parts. The part e^a which is energy free and the part $e - e^a$ which is penalized. If this model where used, we would be assuming that the expected value of the deformation would be close to e^a and not to zero as is currently done. This has the advantage over the thin-plate spline model (which also penalizes only part of the deformation) of being able to map the the active deformation directly in terms of local strains.

6.4.3 A Hierarchal Estimation Scheme for Finding the Active Component

In this scheme we are proposing an approach for solving for the active component e^a in a multi-frame estimation setup, such as for left ventricular deformation. The problem is to be solved in an iterative fashion where we iterate over the frame-set a number of times until convergence.

The first step in the approach is the generation of a database of strains from a previously analyzed set of experiments of the same type. We will label this prior database to generate a prior probability distribution for e^a, $p(e^a)$, with mean e^d.

At any given frame we label the value of the strain at the previous time frame as e^- and the value at the next frame as e^+ if these are available (unless it is the first iteration we will have estimates of these.) We use this to generate the average strain $e^t = 0.5(e^- + e^t)$, and we model the difference $e^t - e^d$ with a zero mean Gaussian distribution as:

$$p(e^t|e^a) = k \exp\left(-(e^t - e^d)'\Sigma^-1(e^t - e^d)\right) \tag{6.27}$$

Then $\hat{e}^a$ can be defined as the maximum a-posteriori estimate of e^a given measurements e^t and the prior probability density function of e^a derived from the strain database. Thus e^a would take the form:

$$\hat{e}^a = \underset{e^a}{\arg\max}\, p(e^t|e^a)p(e^a) \tag{6.28}$$

This best estimate of e^a, $\hat{e}^a$ can then be used as the mean from the prior probability density function of e itself, by inserting it into the 'active' energy function of equation (6.26).

We further note that there is an interesting side issue here. In equation (6.26) the matrix C plays a role similar to the covariance matrix. However we can determine an alternative covariance matrix from the estimation of $\hat{e}^a$. It is not clear what the best choice ought to be, but it is possible that the matrix C could also be adjusted to take account of the probabilistic variation of the strains given the values of the adjacent frames and the strain database. If for example the strain in a particular direction does not vary over a number of experiments we would be tempted to increase the stiffness in that direction to keep this variation low regardless of whether this would contradict the underlying material properties. In practice, one would hope, that the strain along stiffer material directions, as measured from biomechanical experiments, would be less variable and hence C as derived from the model would be close to the estimated covariance of e as derived from the strain database.

Chapter 7

Estimating Left Ventricular Deformation

In this chapter we turn our attention to the major practical application in this thesis: the estimation of left ventricular deformation from three-dimensional medical images from a variety of modalities.

In section 7.1 we first describe how the images were acquired. Following this in section 7.2 we focus on how the general methodology developed in chapters 3–6 was applied to the analysis of the left ventricular image sequences. In section 7.3 we compare the output of the algorithm to implanted sonomicrometers and markers used as a gold standard. Finally in section 7.4 we describe the output of this method from various datasets and see how these correlate with invasive measurements such as histochemical markers of infarction and measures of myocardial blood flow.

7.1 Image Acquisition

7.1.1 Canine MR-images

ECG-gated magnetic resonance imaging was performed on a GE Signa 1.5 Tesla scanner. Axial images through the LV were obtained with the gradient echo cine technique. The imaging parameters were: section thickness=5 mm, no intersection gap, 40 cm field of

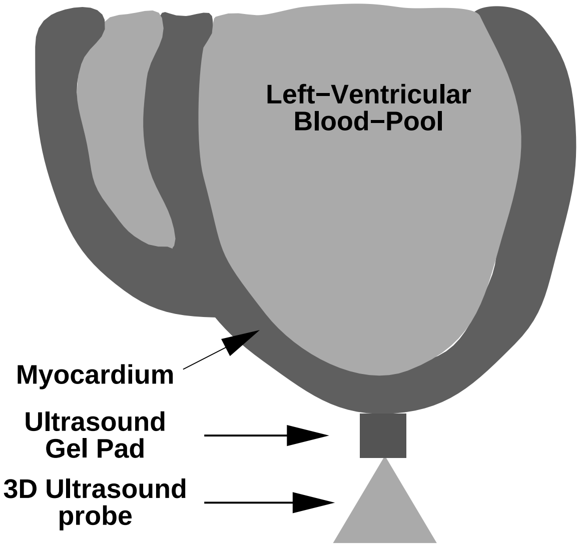


Figure 7.1: Image acquisition geometry for the 3DE images, in the case of open chest dogs.

view, TE 13 msec, TR 28 msec, flip angle 30 degrees, flow compensation in the slice and read gradient directions, 256 x 128 matrix and 2 excitations. The resulting 3D image set consists of sixteen 2D image slices per temporal frame, and sixteen temporal 3D frames per cardiac cycle, with an in-plane resolution of $1.6mm$ and a slice thickness of $5mm$. The dogs were positioned in the magnetic resonance scanner for initial imaging under baseline conditions. The left anterior descending coronary artery was then occluded, creating an infarcted region producing mechanical dysfunction, and a second set of images was acquired. An example of such an acquisition was shown in figure 2.2. In some of the studies, markers were implanted for validation purposes. This will be discussed in more detail in section 7.3.

7.1.2 3D Echocardiography (3DE)

The 3DE images were acquired using an HP Sonos 5500 Ultrasound System with a 3D transducer (Transthoracic OmniPlane 21349A (R5012)). The 3D-probe was placed at the apex of the left-ventricle of an open-chest dog using a small ultrasound gelpad (Aquaflex) as a standoff. This is illustrated in figure 7.1. Each acquisition consisted of 13–17 frames per cardiac cycle depending on the heart rate. The angular slice spacing was 5 degrees resulting in 36 image slices for each 3D frame. In some of the studies we also implanted sonomicrometer crystals for validation, see section 7.3 for the details.

7.1.3 Human MR-images

The Human MR images were acquired using breath-hold techniques at 16 slice levels. These acquisitions provide exquisite full, cine-3D image sequence magnitude data within several minutes at a spatial resolution of 1.5mm x 1.5mm x 5mm and a temporal resolution equal to the duration of the cardiac cycle divided by 20 phases (usually around 40msec).

7.1.4 Dynamic Spatial Reconstructor Data

The Dynamic Spatial Reconstructor is a three-dimensional X-Ray computed tomography scanner at Mayo Clinic. It can provide accurate, stop-action images of moving organs of the body. The canine data we are using was acquired at 33 msec frame intervals in real time, with the spatial resolution of 0.91mm in all three dimensions. For more information the reader is referred to Robb[88].

7.2 Image Analysis

7.2.1 Segmentation and Shape-Based Tracking

The endocardial and epicardial surfaces were extracted interactively using a software platform [76] which was described in section 3.4. In the case of the 3DE images the contours were extracted from the original images then resampled to generate planar contours in Cartesian space, to match the output from the MR and the DSR data. Interpolated contours were generated between the extracted ones using chamfer interpolation (see section 4.1.1) to give isotropic sampling of the resulting surfaces. The distance between adjacent points on the surface was approximately 0.5 voxels. The surfaces were then reconstructed using Delaunay Triangulation (section 4.2.1) and smoothed using the non-shrinking algorithm described in section 4.2.2. Curvatures were calculated (section 4.2.3) and the shape based tracking algorithm applied to generate a set of initial matches and confidence measures for all the points on the surface. (see section 4.4)

Probabilistic modeling the initial displacement estimates: Given a set of displacement vector measurements u^m and confidence measures c^m, we model these estimates probabilistically by assuming that the noise in the individual measurements is normally distributed with zero mean and a variance σ^2 equal to $\frac{1}{c^m}$. In addition we assume that the measurements are uncorrelated. Given these assumptions we can write the measurement probability for each point as:

$$p(u^m|u) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(u-u^m)^2}{2\sigma^2}} \tag{7.1}$$

This constitutes the data term of the deformation model.

7.2.2 Modeling the myocardium

The myocardium is modeled as a transversely isotropic linear elastic solid. This model is described in section 5.2.1 and enabled us to capture the preferential anisotropy of the tissue along fiber directions. The fiber orientations were modeled using the model of Guccione et al [39] which resulted in the fiber pattern shown in figure 5.2.

This model assumes that fibers lie in the plane defined by the local circumferential (C) and longitudinal (L) axes. First we define the cardiac-specific coordinate system shown in figure 7.2. In the undeformed state, the radial (R) axis points outwards, the circumferential axis (C) is along the circumference of a planar section and the longitudinal axis (L) is vertical. The fiber (F) and cross-fiber axis (X) lie in the plane defined by C and L. The fiber orientation can then be defined by the angle θ as shown in the diagram. The epicardial fiber angle varied between $-43°$ at the base and $-53°$ at the apex, and the endocardial fiber angle varied between $82°$ at the base and $97°$ at the apex. All the

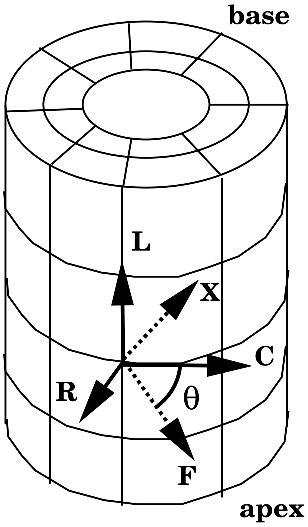


Figure 7.2: Coordinate system used to define fiber orientation. The fiber direction (F) lies in the plane defined by the circumferential (C) and longitudinal axis(L) at an angle θ anti-clockwise from the circumferential axis.

other fiber angles can be found by linearly interpolating both along the vertical and the radial directions[39].

The model resulted in an internal energy function $W(C, u)$, where C represents the material properties and u the displacement field. This was used to generate an equivalent prior probability density function $p(u)$ of the Gibbs form:

$$p(u) = k_1 \exp(-W(C, u)) \tag{7.2}$$

Geometrically the myocardium was discretized using the algorithm described in section 4.3 to produce a hexahedral mesh. This mesh consisted of $1000 - 2000$ elements (depending on the geometry).

7.2.3 Integrating Model and Data

Having defined both the data term (equation (7.1)) and the model term (equation (7.2)) as probability density functions we naturally proceed to write the overall problem in a Bayesian estimation framework. Given a set of noisy input displacement vectors u^m, the associated noise model $p(u^m|u)$ (data term) and a prior probability density function $p(u)$ (model term), find the best output displacements $\hat{u}$ which maximize the posterior probability $p(u|u^m)$. Using Bayes' rule we can write.

$$\hat{u} = \underset{u}{\arg\max}\, p(u|u^m) = \underset{u}{\arg\max} \Big(\frac{p(u^m|u)p(u)}{p(u^m)} \Big) \tag{7.3}$$

The prior probability of the measurements $p(u^m)$ is a constant once these measurements have been made and therefore drops out of the minimization process. In this expression we also note that there is an undefined constant. This is the scaling factor k_{sc} that translates the stiffness of the mechanical model to the effective maximum value of the covariance matrix of its equivalent probability density function $p(u)$. This was discussed in more detail in section 6.1.4. The value of this constant (k_{sc}) sets the relative weight of the data term to the model term. We set this adaptively to be as large as possible (which pushes the optimum towards the data side) subject to solution convergence. In this way we make the following assumption: the best solution is the one which adheres as much as possible to initial estimate of the displacement field but still results in a connected solid. Convergence fails when the Jacobian of the deformation field[1] becomes singular. In this case we lower the value of this weight to produce a smoother displacement field.

Model bias and correction: We also note that the mechanical model prior is generated by a passive biomechanical model. As this does not capture the active deformation of the heart, it has a major weakness in that it penalizes all deformations. This model could be thought in some sense as having a mean of zero strain and a variance propor-

[1] The Jacobian of the deformation is the matrix F defined in figure 5.1.

tional to the reciprocal of the stiffness. It will tend to bias the strain estimates towards zero. As a certain amount of deformation *does occur* the use of this passive model results in an underestimation of the deformation. At this point the problem is dealt with by forcing the nodes which lie on the endocardial and epicardial surfaces at time t to lie on the segmented surfaces at the time $t+1$. (See also section 6.3.)

7.2.4 Numerical Solution

Taking logarithms in equation (7.3) and differentiating with respect to the displacement field u results in a system of partial differential equations, which we solve using the finite element method [9]. This is almost identical to the example problem described in section 5.3.

For each frame between end-systole (ES) and end-diastole (ED), a two step problem is posed: (i) solving equation (7.3) normally and (ii) adjusting the position of all points on the endocardial and epicardial surfaces so they lie on the endocardial and epicardial surfaces at the next frame using a modified nearest-neighbor technique and solving equation (7.3) once more. This ensures that there is a reduction in the bias in the estimation of the deformation.

7.2.5 Strain Analysis

For the purpose of analyzing the results, the left-ventricle of the heart was divided into a number of cross-sectional slices, slice 1 being at the apex of the ventricle, with the slice number increasing towards the valve plane. Each slice was further subdivided into 8 sectors, as shown in figure 7.3. We report, depending on the application, the average of radial(RR), circumferential(CC) and longitudinal(LL), fiber (FF) and cross-fiber (XX) strains for these sectors. In some cases we will report average strains over endocardial and epicardial half-sectors, again as shown in figure 7.3 in the case of sector 7.

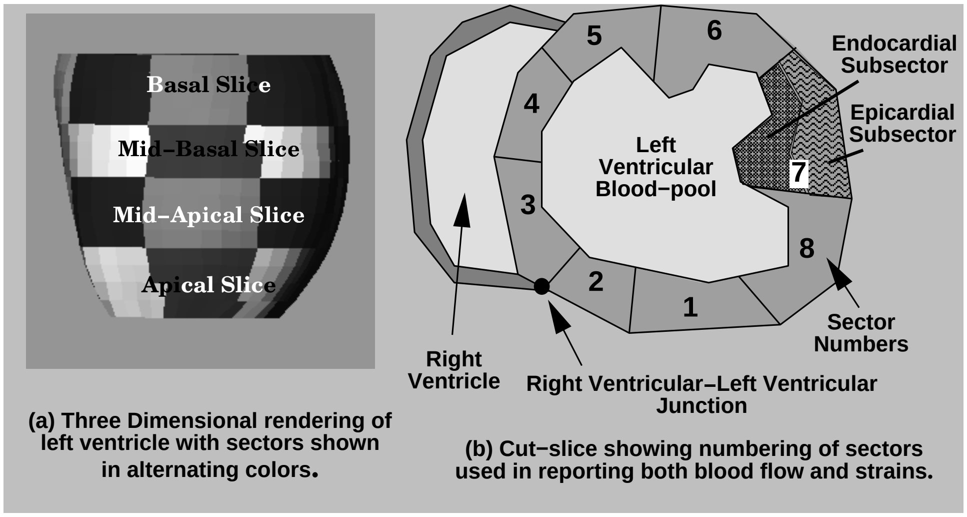

Basal Slice
Mid–Basal Slice
Mid–Apical Slice
Apical Slice
(a) Three Dimensional rendering of left ventricle with sectors shown in alternating colors.
5
6
4
3
2
1
8
7
Endocardial Subsector
Epicardial Subsector
Left Ventricular Blood–pool
Sector Numbers
Right Ventricle
Right Ventricular–Left Ventricular Junction
(b) Cut–slice showing numbering of sectors used in reporting both blood flow and strains.

Figure 7.3: Division of the left ventricle(LV) into slices and sectors for the the purpose of reporting results, and comparing the postmortem and regional blood flow data. In this example the LV is divided into four slices, although this number differed depending on the size of the LV and the purpose of the data analysis. Each sector consists of approximately 75 elements in the finite element mesh. In some cases we divide each sector into two half-sectors, an endocardial half-sector and an epicardial half sector. (For an example see sector 7.)

7.2.6 Measures of Myocardial Viability

In this section we present two techniques used to invasively assess myocardial viability. The results of these techniques are used to assess whether the image derived strains are an effective measure of the underlying state of the tissue, that is if they can be used to distinguish between different pathophysiological states of the myocardium.

Postmortem: Triphenyl-Tetrazolium Chloride (TTC) staining was used to define the extent of cell necrosis (death) following five hours of coronary occlusion, thus defining the area of actual injury in the tissue. The regional volume of the postmortem injury zones are found by digitizing color photographs of the TTC–stained *post mortem* myocardial

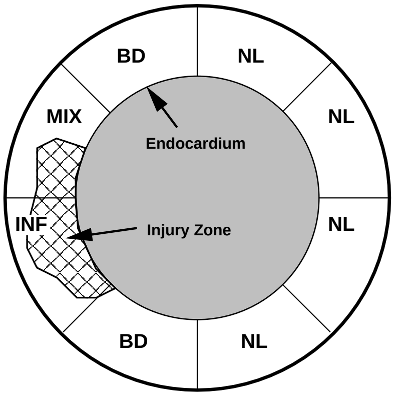


Figure 7.4: Labeling of sectors on one postmortem slice based on TTC staining. A sector was labeled as Infarct (INF) if the injury extended around the full extent of the endocardium within the sector. When the infarct area was less substantial ($<$ 25%) the sector was labeled as Mixed (MX). In cases where a sector had little or no injury zone but was next to a MX or an INF sector it was labeled as a Border (BD) (This part of the labeling also accounted for the labels of the sectors in slices above and below the sector). All other sectors were labeled as normal (NL).

slices (5mm thick) from the excised hearts. The endocardial, epicardial and infarction zone boundaries of each *post mortem* left ventricular slice are hand-traced, aligned, and stacked to reconstruct the three-dimensional profile of the injury zone. Each slice is divided into 8 sectors, as was the case with the regional strains (see figure 7.3). Each sector is then labeled as Infarct (INF), Mixed (MX), Border (BD) or Normal (NL) depending on the percentage of injury within the sector and the labels of the neighboring sectors as described in figure 7.4. We also calculate the percentage of the injury in each sector.

Regional Blood Flow: In the 3DE studies, where the postmortem information was not available, the regional blood flow in the myocardium was used to identify the underlying functional state.[2] The regional blood flow was determined using a radio-labeled microsphere technique. Here, radio-labeled microspheres were injected into the left atrium and reference blood samples were drawn from the femoral arteries. Regional myocardial blood flow was calculated using a method previously described by Sinusas et al[92]. We again divide the left ventricle into four slices (as shown in figure 7.3) and each slice into 8 sectors. A sector is considered to be in the risk area if endocardial microsphere flow was less than $0.25 ml/min/g$ at the time of the occlusion. In the case of LAD occlusion the normal region was defined by 5 transmural sectors located in the posterior lateral wall at the base of the heart (sectors 5,6,7 of the basal slice and sectors 6,7 of the mid-basal slice).

7.3 *In-Vivo* Validation

In this section we present validation of the image derived strains using implanted markers and sonomicrometers as *gold standards.* We note that, to the best of our knowledge, this is the only such validation currently in the literature.

7.3.1 Implanted Image-Opaque Markers:

Methodology: To validate the image-derived strains markers were implanted on canine hearts as follows: First the canine heart was exposed through a thoracotomy. Arrays of endocardial, midwall and epicardial pairs of markers were then implanted as shown in figure 7.5. They were loosely tethered, combinations of small copper beads (which show up *dark* in the MR images) at the endocardial wall and the midwall region and small plastic capsules filled with a 200:1 mixture of water to Gd-DTPA at the epicardial wall (which show up *bright* in the MR images). Marker arrays were placed in two locations on the canine heart wall. The location of each implanted marker is determined in each temporal frame by first manually identifying all pixels which belong to the marker area

[2] These blood flow measurements were also available for the canine MR studies, but since the post-mortem information was also available, the blood flow measurements were not used in that case.

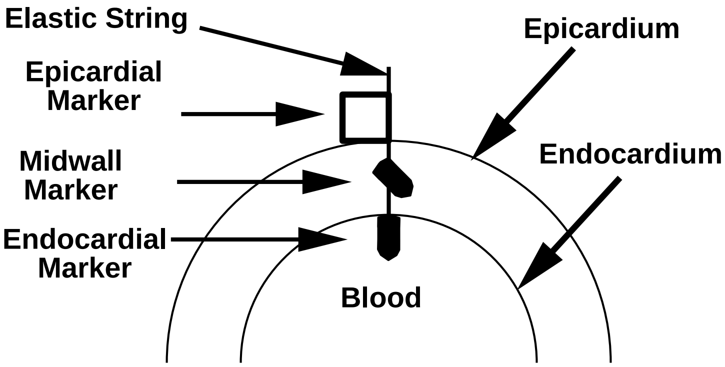


Figure 7.5: Implantation of Image-Opaque Markers. This figure shows the arrangement of markers on the myocardium. First a small bullet-shaped copper bead attached to an elastic string was inserted into the blood pool through a needle track. Then the epicardial marker was sutured (stitched) to the myocardium and tied to the elastic string. Finally, the midwall marker was inserted obliquely through a second needle track to a position approximately half-way between the other two markers.

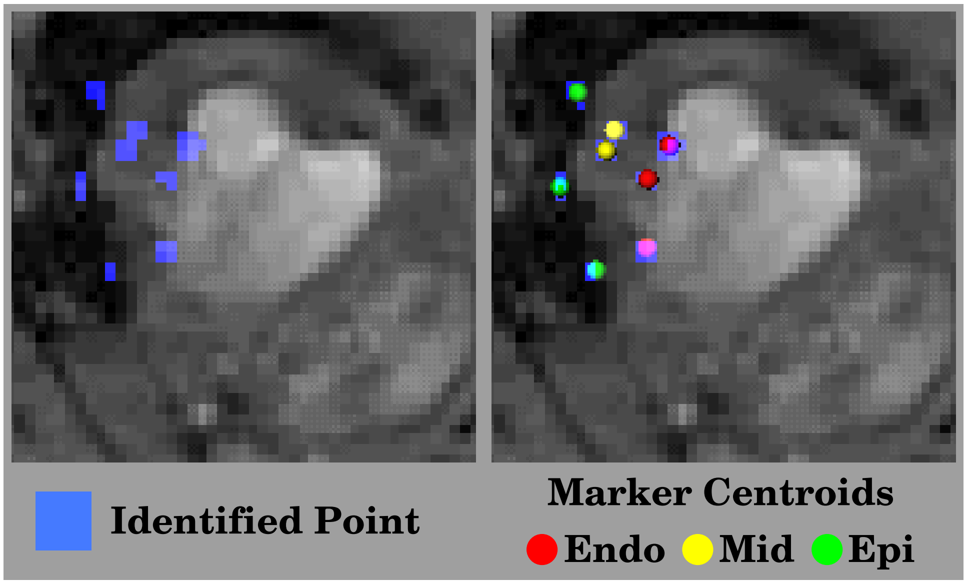


Figure 7.6: Localization of implanted markers. Arrays consisting of 12 markers each were placed at two positions on the left ventricle. In this figure, we show the portion of one marker array as it intersected a short-axis MR image slice. A human observer identified the pixels corresponding to each marker (left) and the marker positions (right) were found by calculating centroids of these points.

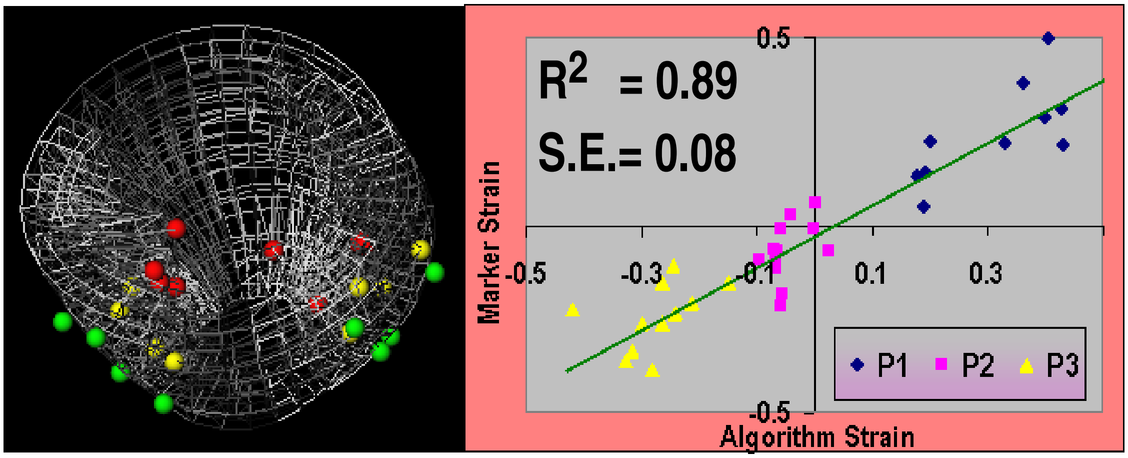


Figure 7.7: Algorithm-derived Strains vs. Implanted Marker-derived Strains. Left: Reconstructed LV volume from cine-MRI at ED with marker positions noted as spheres (red=endo,yellow=mid,green=epi). Right: Scatter plot of principal strains derived from baseline and post-infarction cine-MRI studies using algorithm vs. same strains derived from implanted marker clusters at two positions in the LV wall for $N = 4$ dogs (There was a total of 12 useable extracted marker arrays).

(because of imaging artifacts the marker 'image' extends to more than one voxel) and then computing the 3D centroid of that cluster of points, weighted by the grey level[3]. Figure 7.6 shows a short-axis MR slice of the heart with the identified marker pixels shown in blue (left). The marker centroids are shown on the right.

Results: The image-derived strains were compared to strains derived from implanted markers. In the case of the markers the strains were computed as follows using only the epicardial and endocardial markers. In each region of the LV that contained markers, groups of either 6 or 8 markers (depending on the geometry) were connected to form either prism or hexahedral elements. Given the known displacements, we then calculated the strains in these markers. These strains were compared to the average strains in the

[3] In the case of dark markers the image is first inverted.

elements contained within each marker array. We used principal strains[4], as the marker arrays where large and included elements where the cardiac-specific directions varied widely.

Comparison results are shown in figure 7.7 for $N = 4$ dogs (2 acquisitions per dog, one pre-occlusion and one post-occlusion). We observe a strong correlation of the principal strain values ($r^2 = .89$).

7.3.2 Sonomicrometers

Methodology: In the case of the 3DE images we validate the strain estimates using implanted sonomicrometers. The canine heart is again first exposed through a thoracotomy. With the aid of an implantation device constructed in our laboratory, two crystal-arrays each consisting of 12 crystals (3 sub-epicardial, ∼2.0 mm, 6 mid-wall and 3 sub-endocardial, ∼0.75 mm diameter) were placed in the heart wall. To define the LV long axis a crystal was implanted in the LV apex and two at the base of the LV, one near the bifurcation of the left main coronary artery and the second in the posterior wall. Finally, to define a fixed coordinate space, 3 crystals attached to a plexi-glass frame were secured in the pericardial space under the right ventricle.

Digital sonomicrometry employs the time of flight principal of ultrasound to measure the distance between a transmitter and a receiver. A total of 32 crystals are used in each study. The distances between all possible pairs of crystals are recorded along with LV and aortic pressure at a sampling frequency of greater than 125 Hz. There are a number of preprocessing steps involved in obtaining the positions of the crystals over time from the crystal to crystal pair lengths. These are described by Dione et al[27]. The efficacy of this technique was illustrated by additional work [69] that showed that the distances obtained with sonomicrometers compared favorably ($r = 0.992$) with those obtained using the more established technique of tracking implanted bead displacements using biplane radiography.

[4] These are defined in section 5.1.4.

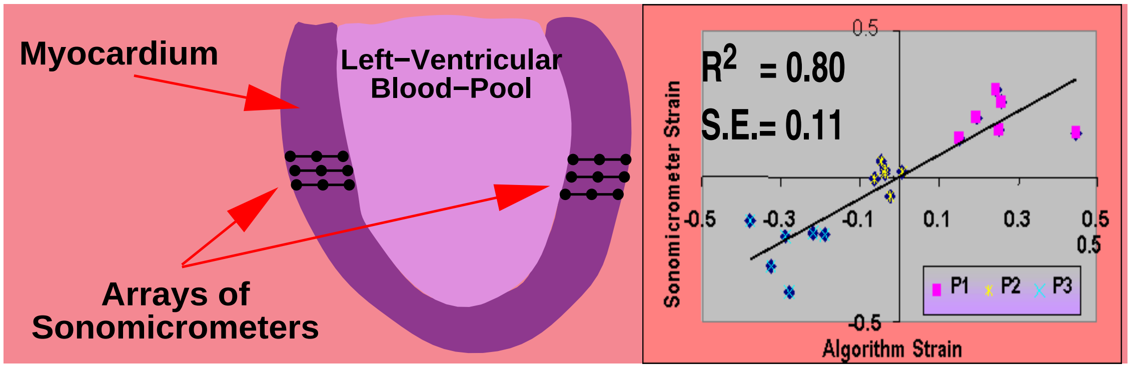


Figure 7.8: 3DE Algorithm-Derived Strains vs. Sonomicrometer-derived Strains. Left: Placement of arrays of sonomicrometers in the Left Ventricular Wall. Right: Scatter plot of principal strains derived from N=3 3DE studies using the algorithm vs. same strains derived from sonomicrometer arrays (12 crystals in each cluster) at two positions in the Left Ventricular wall. Note the high correlation between the two sets of strain values ($r^2 = .80$).

Results: We compared our image-derived strains to concurrently-estimated sonomicrometer-derived strains at several positions in the LV myocardium in the same dogs. The sonomicrometers were visually located from the images and the two nearest sectors of algorithm-derived strains were selected for comparison purposes. The comparison of the principal strain components in two separate regions for a set of 3 studies (the sonomicrometer data was not available for study ‘D4’) showed a strong correlation ($r^2 = 0.80$). Here we compare the principal strains as it is difficult to estimate the cardiac specific directions in the case of the sonomicrometer data. A scatter plot of algorithm-derived principal strains versus sonomicrometer derived principal strains is shown in figure 7.8. This validation is still in a preliminary stage and we hope in the future to also validate strain patterns which are not fully averaged across the wall.

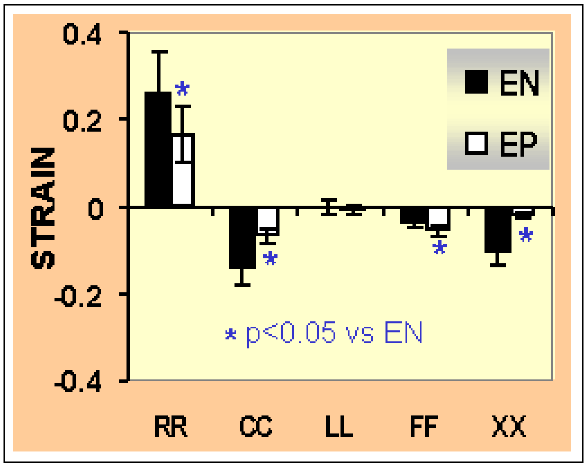


Figure 7.9: Average strain information at baseline for N=8 dogs. Endocardial (EN) vs. epicardial strains (EP) are significantly different for all strains except longitudinal ($p < .05$); Note that RR=radial, CC=circumferential, LL=longitudinal, FF=along fiber, XX=cross-fiber strains.

7.4 Results

In this section we presents results obtained using this algorithm on Magnetic Resonance (both canine and human), 3D Echocardiography and DSR Images. Further, in the case of canine MR we compare the results with postmortem information and in the case of 3DE with myocardial flow measurements. No complementary measure was available for the human MR and the DSR images.

7.4.1 Canine MRI

Normal: For reporting purposes, the left ventricle was divided into three slices each consisting of eight sectors. We observed uniformity of Radial (R) and Circumferential (C) strains (ranges: R:$15 \pm 6\%$ to $23 \pm 7\%$; C:$-9 \pm 5\%$ to $-12 \pm 2\%$). Regional LV strains and shears were consistent between dogs and comparable to values derived using both implanted markers and MR tagging [21]. Figure 7.9 shows average strains in the

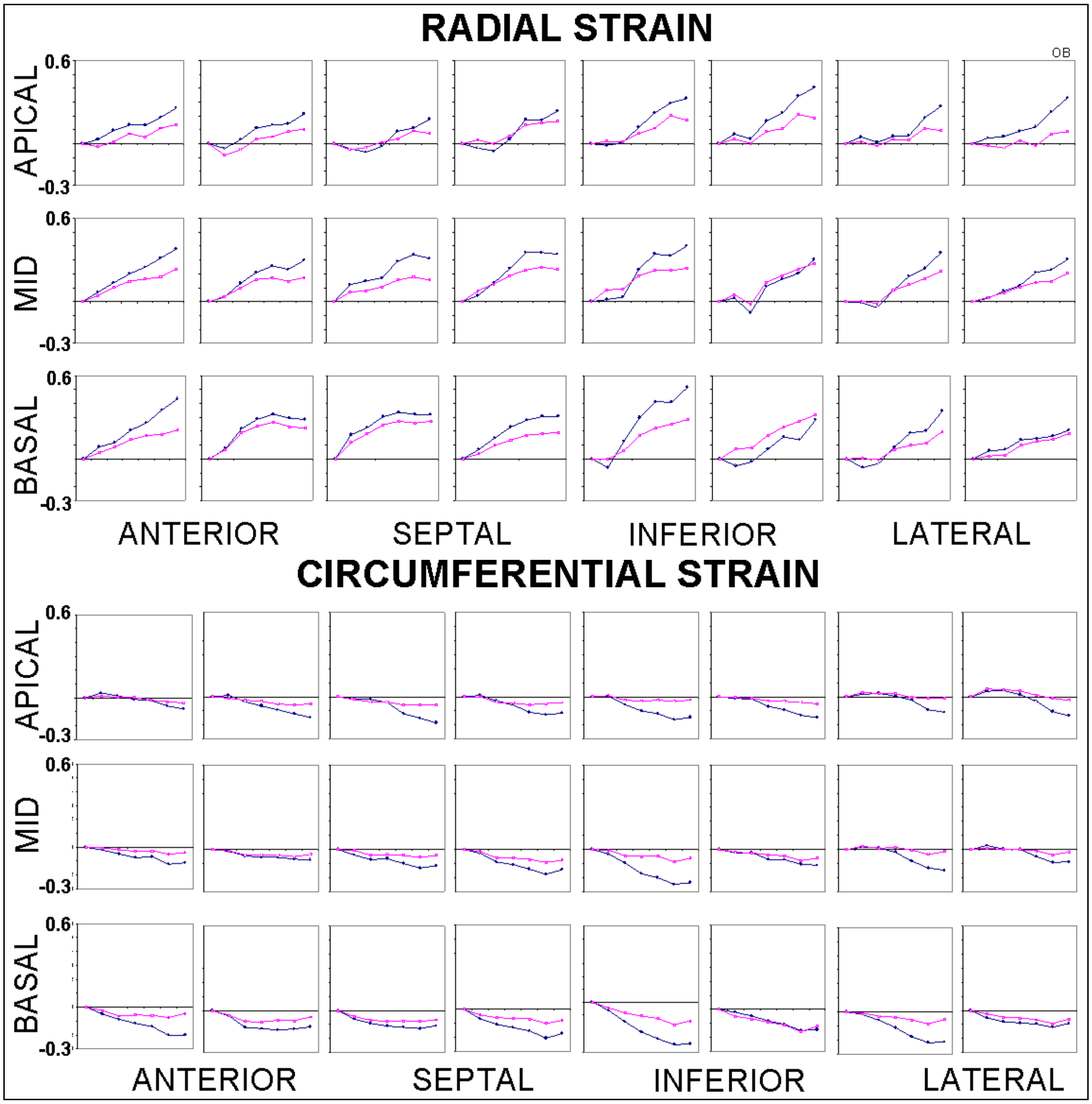


Figure 7.10: Baseline (normal) canine LV strains derived from cine-MRI. Development of radial and circumferential strain at 3 slice levels in 8 radial sectors in a single study. Each plot shows the strain evolution from ED to ES in 2 transmural halves (endocardial half=blue, epicardial half=magenta).

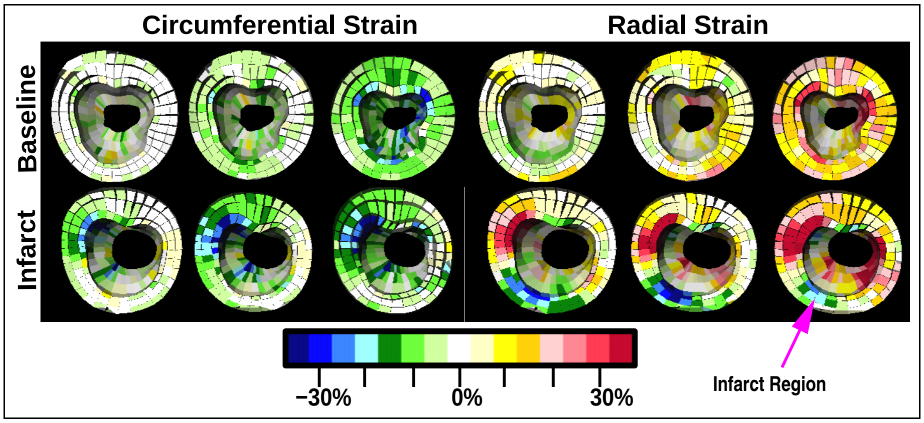


Figure 7.11: Strain Development in Post-Infarction (and vs. Baseline) Canine LV derived from cine-MRI. Left side: mid-ventricle cutaway views through the 3D reconstructed volume show the strain patterns that develop at 1/3, 2/3 and 3/3 of the time between ED and ES. The leftmost displays illustrate the circumferential (CC) strains (normal ED-ES shortening is in the blue-green region). The displays just to the right show the radial (RR) strains (normal ED-ES thickening in yellow-red region).

endocardial half-sectors and the epicardial half-sectors. Note that statistically significant differences were observed between the endocardial and the epicardial half-sectors.[5]

Figure 7.10 shows the temporal development of Radial and Circumferential strains from End-Diastole (ED) to End-Systole (ES) for one canine study. Here we plot strain for half-sectors (each sector is divided into an endocardial half and an epicardial half). This is also illustrated in the top half of figure 7.11 which compares the raw non-averaged strain patterns with those obtained after LAD occlusion.

[5] In the simplified case of a thick cylinder contracting without changing its volume, it can be shown that the in-plane (perpendicular to the long-axis of the cylinder) deformation varies as a function of $\frac{1}{r^2}$ where r is the distance from the long axis. Hence, were this model to be applied in the case of the left ventricle, it would predict that the radial and circumferential endocardial strains would be larger than the corresponding epicardial strains. While this model offers a course approximation to the actual deformation, it is nice to see that the real results are in qualitative agreement with it.

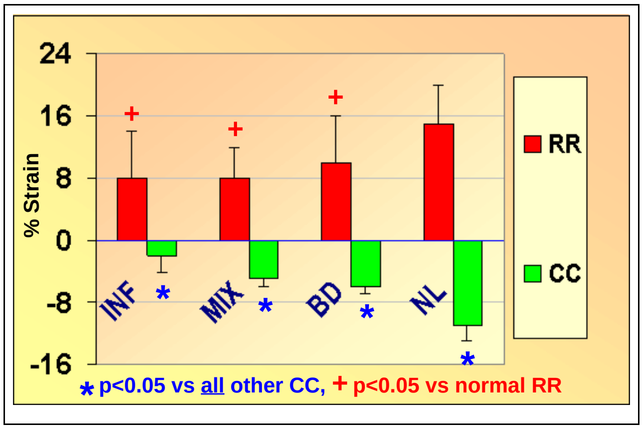


Figure 7.12: Average radial and circumferential strains (vertical axis is % strain) for different *postmortem*-classified regions, from N=8 post-occlusion cine-MRI studies. Note that CC is able to separate all classifications (INF=infarct,MIX=mixed,BD=border,NL=normal), while RR can only separate NL from the other classifications.

Post-Occlusion The occlusion of the LAD causes significant changes in the observed strain patters as expected. A pre-occlusion/post-occlusion comparison is shown in figure 7.11. This shows the same pattern as the raw images shown in figure 2.2.

For quantitative analysis, the ventricle was divided to have the same number of slices as the histochemical staining maps of the actual injury zone, to make registration between the two easier. In the first part of the analysis each slice was further subdivided into eight sectors. The histochemical staining maps were used to label these sectors as one of four categories: infarcted (INF), mixed (MIX), adjacent (BD), and normal (NL).

Given the relative uniformity of the radial and circumferential strains from the normal data-set, we tested whether any of the strain components as estimated in the post-occlusion studies could be used to discriminate between these different classes (INF, MIX, BD, NL). We found that the circumferential strain discriminated all myocardial regions to a level of significance $p < 0.05$. This demonstrated that this methodology can be applied to discriminate different regions non-invasively as shown in figure 7.12.

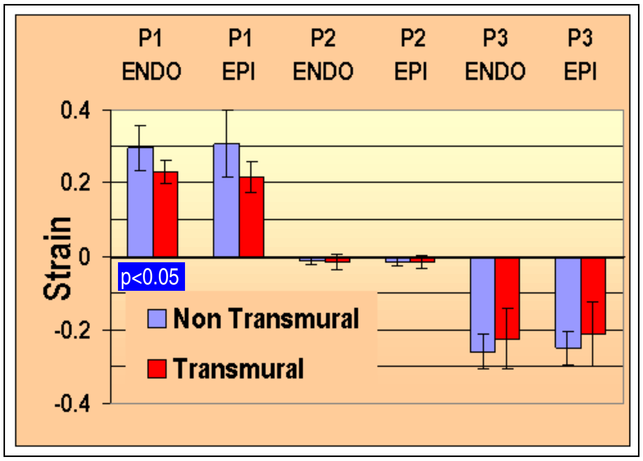


Figure 7.13: Results from Nontransmural ($N = 6$) vs. Transmural ($N = 6$) Acute Canine Studies: Shown are the 3 *principal* strains (p1,p2,p3) derived from cine-MRI. Note significant difference in first principal endocardial strain.

In the second part of the analysis we attempted to see whether this methodology could distinguish between animals where there was *post mortem*-confirmed globally transmural injury, as opposed to nontransmural injury. Here, we used 12 studies performed experimentally and imaged as described above, which separated into two $N = 6$ groups based on the following *post mortem* criteria. The first group, labeled transmural, contained the dogs that had two or more *post mortem* infarct (I) sectors with greater than 75% injury. The other dogs were placed into the nontransmural group. For testing purposes, we then compared the principal strains within the endocardial and epicardial halves of 1.) the sectors having greater than 75% infarct in the transmural dogs and 2.) the sectors having greater than 25% infarct in the nontransmural dogs. We found that there was a significant difference between the transmural and nontransmural dogs in the values of the endocardial, first principal strains, indicating the plausibility of using 3D strain for separating these physiological states. A graph of all of the endocardial and epicardial principal strains for both the transmural and nontransmural dogs is shown in figure 7.13.

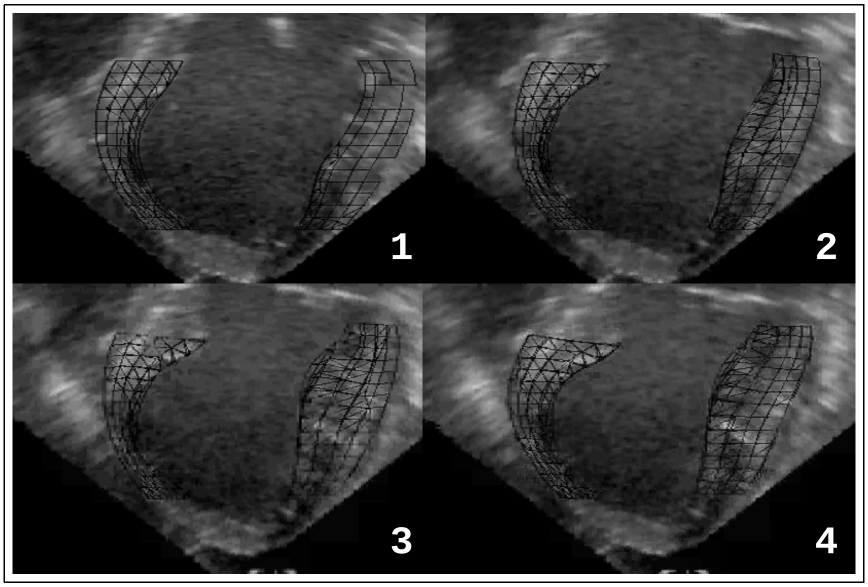


Figure 7.14: "3DE-tissue-tagging"- a slice through a 3D visualization with the algorithm-driven deforming mesh overlaid on one slice through a 3DE dataset at four time points between ED and ES. This demonstrates the output of the algorithm which tries to follow (or tag) material points in time, similar to the Magnetic Resonance Tagging approach.

7.4.2 3D Echocardiography

We report here on results from 3DE studies ($N = 4$). The images were obtained either before (D1 and D2) or after occlusion of the left anterior descending coronary artery (D3 and D4), using the procedure described in section 7.1.2

The potential of our methodology is illustrated in figure 7.14, which shows a cut through our tracked 3D mesh overlaid on a slice through the original 3DE image data over time. This could be seen as a form of software-derived, 3DE-based "tissue tagging" somewhat in the sense of MR tagging. Note the spreading grid lines near the endocardium on the right as the LV thickens from ED to ES. There is also an infarct region in the lower left half of the image which exhibits bulging instead of contraction. The progressive development of regional radial and circumferential strains for 'D3' is shown in figure 7.15.

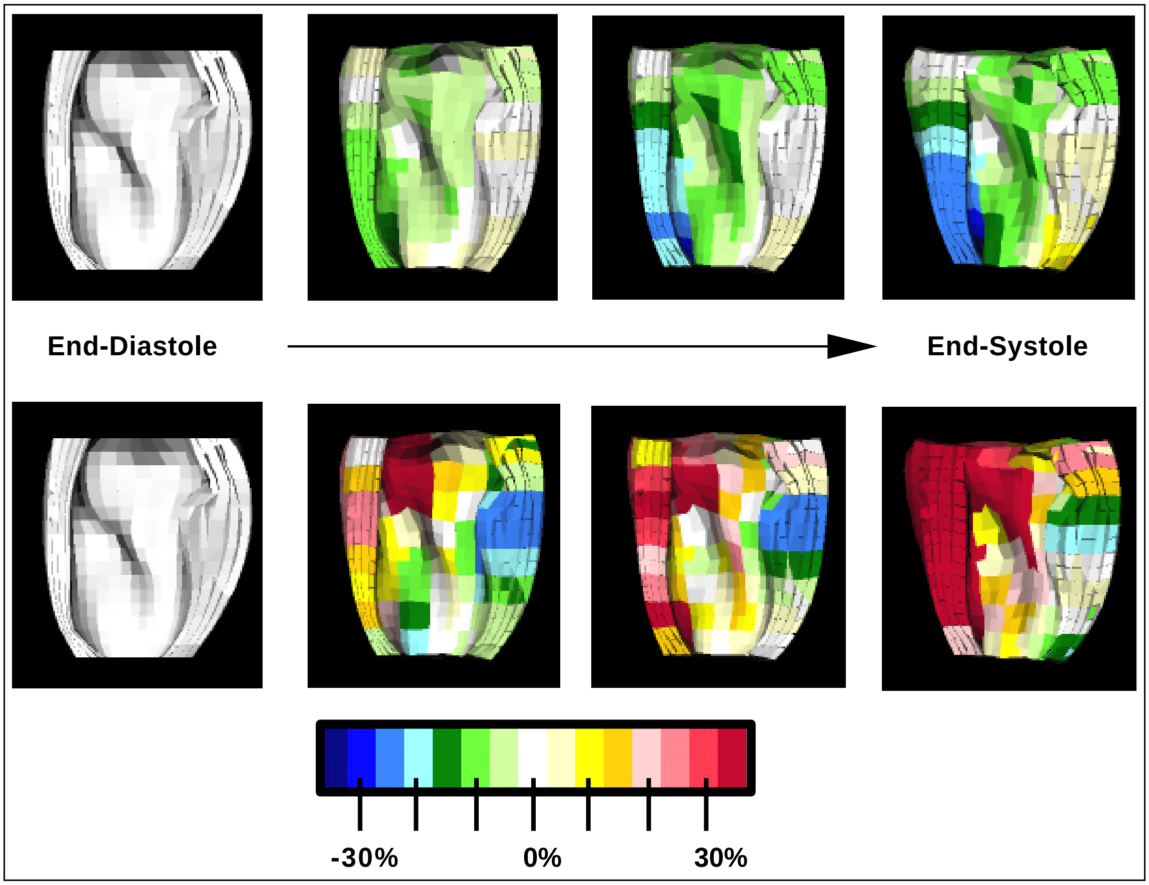


Figure 7.15: A long-axis cut-away view of the LV showing 3DE-derived circumferential (top) and radial (bottom) % strain development at 4 time points between ED and ES in a dog following LAD occlusion (on the lower right half of the heart). The strains shown here are averaged in eight transmural sectors in each slice as described in figure 7.3. Note the normal behavior in the left half of the heart, showing positive radial strain (thickening) and negative circumferential strain (shortening) as we move from ED to ES. The lower right half of the heart where the affected region was located showed almost the opposite behavior, as expected.

Study	D1	D2	D3	D4
Normal Radial Strain	17.7	13.8	22.4	17.2
Normal Circumferential Strain	-13.4	-13.1	-8.4	-12.4
Normal Longitudinal Strain	-4.3	-3.2	-3.4	-3.1
Risk Area Radial Strain	n/a	n/a	-4.3	-13.7
Risk Area Circumferential Strain	n/a	n/a	1.9	-7.3
Risk Area Longitudinal Strain	n/a	n/a	-0.7	-2.0

Table 7.1: Summary of results for four animal studies. There was no risk area (n/a=not applicable) in studies D1 and D2 as the 3DE images, in these cases, were obtained before coronary occlusion.

The quantitative results are summarized in Table 7.1. Function in the risk area, which was independently defined by microsphere flow, was markedly reduced compared to non-affected regions and the control normal animal. The radial strain is notably smaller in the risk area after coronary occlusion. The circumferential strain becomes less negative also indicating a loss of function. There was a small decrease in the longitudinal strain as well.

It is interesting to note that in a recent publication, Croisille [21] reported similar values (Radial=$23.2 \pm 1.9\%$, Circum.=$-10.5 \pm 2.0\%$ and Long. =$-7.5 \pm 1.0\%$) for strains in the normal regions of dog hearts using three-dimensional tagged MRI. However, they observed smaller reductions in strains post-occlusion, which can be attributed to coronary reperfusion in their model. This probably allowed for partial recovery of function in the risk region.

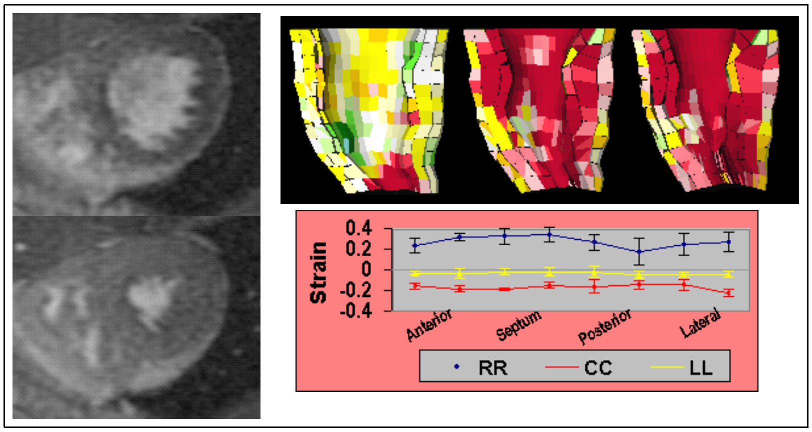


Figure 7.16: Human cine-MRI-derived results. Left: Magnitude breath-hold ED and ES images at a single slice level. Upper right: (see color scale in fig 7.11) radial strains at 3 long axis time points between ED and ES. Lower right: mean *cardiac-specific* strain values for $N = 3$ studies at mid-LV.

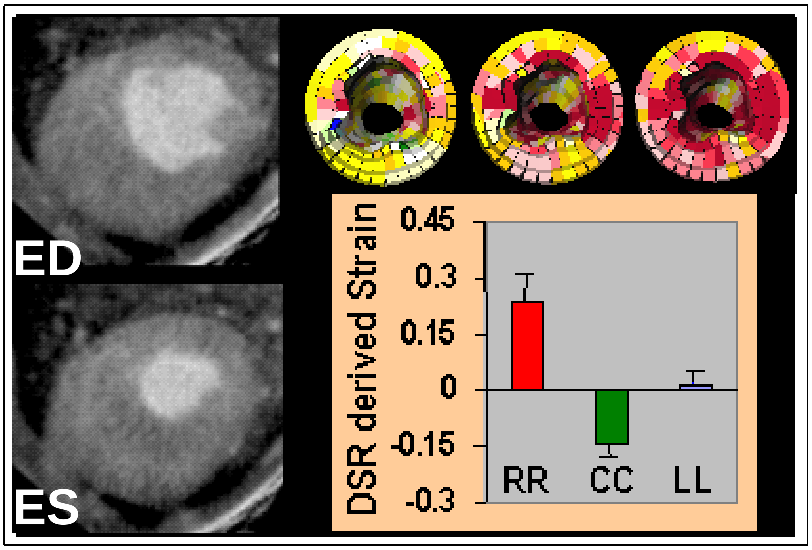


Figure 7.17: Algorithm-derived Strains from Cine-CT (DSR) Images. Left: Example axial slice from baseline dog study at end-diastole (ED) and end-systole (ES). Upper right: Radial strains at 3 time points ED to ES. Lower right: Average radial (RR), circumferential (CC) and longitudinal (LL) strains for $N = 3$ dogs.

7.4.3 Human MRI

We also tested the algorithm on $N = 3$ sequences of breathhold images of normal human subjects. The one difference in the processing, between these acquisitions and the canine MR acquisitions, was that since, in this case, different 3D slice levels are acquired at different breath holds, slices at the same time frame can be misaligned along the long axis of the heart. We have corrected for this by manually aligning the data in each frame.

7.4.4 DSR

To show the utility of our strain computation approach in a third modality, it was also tested on three cine-CT canine experiments performed by Dr. Erik Ritman, at the Mayo Clinic, using the Dynamic Spatial Reconstructor (DSR). The results for a set of baseline (normal state) dogs are shown in figure 7.17. Note that the values reported are in the same range as strains from our own cine-MRI data and those from MR tagging [21].

7.5 Conclusions

In this chapter we have illustrated the application of the general methodology described in this thesis to estimating left ventricular deformation from three-dimensional medical images. We note that modality specific forms of data can be added to this general framework. In the case of magnetic resonance such information could be derived from MR tagging and/or phase contrast (see section 2.3). In the case of 3D Echocardiography we could potentially used velocity data generated using Doppler ultrasound techniques and/or displacement information generated from following graylevel patterns in the images, sometimes known as speckle tracking. However, we have tested the method so far, only using shape-based displacements as an input. The results have been validated *in-vivo* using implanted markers in the case of MRI and sonomicrometers in the case of 3DE. We further demonstrate the usefulness of the estimated strains in determining myocardial viability *non-invasively.*

Further research could include the use of the active model proposed in chapter

6, to properly handle the bias problems inherent in the passive biomechanical model. The active model could also be used as a means of incorporating a temporal continuity/periodicity constraint (see section 6.4.2.)

Ultimately this deformation estimation algorithm could be combined with a segmentation algorithm, to segment and track the LV within an integrated framework, where the processing is done in an iterative fashion. The output of the segmentation algorithm can be used as the input to the deformation estimation algorithm to generate an estimate of the deformation (as was done in this thesis). Then the deformation estimation algorithm (assuming the presence of an active model) could be used to generate a better estimate of the segmentation. Then this new estimate of the segmentation can be used to initialize the next iteration of the segmentation algorithm. This combination of the two algorithms would then result in a closed-loop system, where information from the segmentation algorithm is used to guide the deformation estimation algorithm and vice-versa, and could potentially result in substantial savings in the time needed to obtain a good segmentation of the images.

Bibliography

[1] A. A. Amini, Y. Chen, R. W. Curwen, V. Manu, and J. Sun. Coupled B-snake grides and constrained thin-plate splines for analysis of 2D tissue defomations from tagged MRI. *IEEE Transactions on Medical Imaging*, 17:3:344–356, June 1998.

[2] A. A. Amini, R. W. Curwen, R. T. Constable, and J. C. Gore. MR physics-based snake tracking and dense deformations from tagged cardiac images. In *AAAI Int. Symp. Comp. Vision Med. Image Processing.*, 1994.

[3] A. A. Amini and J. S. Duncan. Bending and stretching models for LV wall motion analysis from curves and surfaces. *Image and Vision Computing*, 10(6):418–430, 1992.

[4] S. Androutsellis-Theotokis and J. L. Prince. Experiments in multiresolution motion estimation for multifrequency tagged cardiac MR images. In *Int. Conf. on Image Processing*, Lausanne, Switzerland, September 1996.

[5] T. Arts, W. Hunter, A. Douglas, A. Muijtens, and R. Reneman. Description of the deformation of the left ventricle by a kinematic model. *J. Biomechanics*, 25(10):1119–1127, 1992.

[6] L. Axel. Physics and technology of cardiovascular MR imaging. *Cardiology Clinics*, 16(2):125–133, 1998.

[7] H. Azhari and et. al. Noninvasive quantification of principal strains in normal canine hearts using tagged MRI images in 3D. *Am. J. Physiol.*, 264:H205–H216, 1993.

[8] I. Babuska, J.E. Flaherty, W.D. Henshaw, J.E. Hopcroft, J.E. Oliger, and T. Tezduyar, editors. *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations.* Springer Verlag, Berlin, 1995.

[9] K. Bathe. *Finite Element Procedures in Engineering Analysis.* Prentice-Hall, New Jersey, 1982.

[10] J. Besag. Spatial interaction and statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36:192–225, 1974.

[11] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, 48:259–302, 1986.

[12] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 567–585, 1989.

[13] K. Caidahl, E. Kazzam, J. Lingberg, G. N. Andersen, J. Nordanstig, S. R. Dahlqvist, A. Waldenstro, and R. Wikh. New concept in echocardiography: harmonic imaging of tissue withoug use of contrast agent. *The Lancet*, 352:1264–1270, 1999.

[14] A. Chakraborty, L. Staib, and J. Duncan. Deformable boundary finding in medical images by integrating gradient and region information. *IEEE Transactions on Medical Imaging*, 15(6):859–870, 1996.

[15] Amit Chakraborty. *Feature and Module Integration for Image Segmentation.* Ph. D. dissertation, Yale University, 1996.

[16] G. E. Christensen, R. D. Rabbitt, and M. I. Miller M. I. 3D brain mapping using deformable neuroanatomy. *Physics in Medicine and Biology*, 39:609–618, 1994.

[17] G. E. Christensen, R. D. Rabbitt, and M. I. Miller M. I. Deformable templates using large deformation kinematics. *IEEE Transactions on Image Processing*, 5(10):1435–1447, 1996.

[18] G.E. Christensen. *Deformable Shape Models for Anatomy.* Ph. D. dissertation, Washington University, Saint Louis, MI, August 1994.

[19] I. Cohen, N. Ayache, and P. Sulger. Tracking points on deformable objects using curvature information. In *Lecture Notes in Computer Science-ECCV92*, pages 458–466. Springer Verlag, 1992.

[20] T. Constable, K. Rath, A. Sinusas, and J. Gore. Development and evaluation of tracking algorithms for cardiac wall motion analysis using phase velocity MR imaging. *Magn. Reson. Med.*, 32:33–42, 1994.

[21] P. Croissile, C. C. Moore, R. M. Judd., J. A. C. Lima, M. Arai, E. R. McVeigh, L. C. Becker, and E. A. Zerhouni. Differentiation of viable and nonviable myocardium by the use of three-dimensional tagged MRI in 2-day-old reperfused infarcts. *Circulation*, 99:284–291, 1999.

[22] C. de Boor. *A Practical Guide to Splines.* Springer-Verlag, New York, NY, 1978.

[23] T.S. Denney Jr. Estimation and detection of myocardial tags in MR images withour user-defined myocardial contours. *IEEE Transactions on Medical Imaging*, 18(4):330–344, April 1999.

[24] T.S. Denney Jr and J. L. Prince. Reconstruction of 3-D left ventricular motion from planar tagged cardiac MR images: An estimation theoretic approach. *IEEE Transactions on Medical Imaging*, 14(4):625–635, December 1995.

[25] P. Dierckx. *Curve and Surface Fitting Splines.* Oxford, 1995.

[26] J. M. Dieudonne. Gradients de directions et la deformations principales dans la paroi ventriculaire gauch normale. *J. Physiol. Paris*, 61:305–330, 1969.

[27] D. P. Dione, P. Shi, W. Smith, P. De Man, J. Soares, J. S. Duncan, and A. J. Sinusas. Three-dimensional regional left ventricular deformation from digital sonomicrometry. In *19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 848–851, Chicago, IL, March 1997.

[28] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, New Jersey, 1976.

[29] J. S. Duncan, R. L. Owen, L. H. Staib, and P. Anandan. Measurement of nonrigid motion using contour shape descriptors. In *Computer Vision and Pattern Recognition*, pages 318–324, Lahanai,Maui, HI, June 1991. IEEE.

[30] J. S. Duncan, P. Shi, R. T. Constable, and A. Sinusas. Physical and geometrical modeling for image-based recovery of left ventricular deformation. *Progress in Biophysics and Molecular Biology*, 69(2-3):333–351, 1998.

[31] P. J. Edwards, D. L. G. Hill, J. A. Little, and D. J. Hawkes. Deformation for image guided interventions using a three component tissue model. In *Information Processing in Medical Imaging*, pages 218–231, Vermont, USA, June 1997.

[32] A. C. Eringen. *Mechanics of Continua*. Krieger, New York, NY, 1980.

[33] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press, San Diego, CA, 1988 (Fourth Edition).

[34] T. Freeman, J. Cherry, and G. Klassen. Transmural myocardial deformation in the canine left ventricular wall. *Am J. Physiology*, 235:H523–H530, 1978.

[35] K. Gallagher, G. Oksada, M. Miller, W. Kemper, and J. Ross. Nonuniformity of inner and outer systolic wall thickening in conscious dogs. *Am J. Physiology*, 249:H241–H248, 1985.

[36] N. Gassler, H-O. Wintzwer, H-M. Stubbe, A. Wullbrand, and U. Helmchen. Transmyocardial laser revascularization: Histological features in hyman nonresponder myocardium. *Circulation*, pages 371–5, 1997.

[37] J. C. Gee, D. R. Haynor, L. Le Briquer, and R. K. Bajcsy. Advances in elastic matching theory and its implementation. In *CVRMed-MRCAS*, Grenoble, France, March 1997.

[38] D. Geman and S. Geman. Stochastic relaxation, Gibbs distribution and Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[39] J. M. Guccione and A. D. McCulloch. Finite element modeling of ventricular mechanics. In P. J. Hunter, A. McCulloch, and P. Nielsen, editors, *Theory of Heart*, pages 122–144. Springer-Verlag, Berlin, 1991.

[40] S. N. Gupta and J. L. Prince. On variable brightness optical flow for tagged MRI. In *Information Processing in Medical Imaging*, June 1995.

[41] M. Guttman, J. Prince, and E. McVeigh. Tag and contour detection in tagged MR images of the left ventricle. *IEEE Transactions on Medical Imaging*, 13(1):74–88, 1994.

[42] E. Haber, D. N. Metaxas, and L. Axel. Motion analysis of the right ventricle from MRI images. In *In Medical Image Computing and Computer-Assisted Intervention - MICCAI*, pages 177–188, Cambridge, MA, October 1998.

[43] D. Heller, P. M. Ferguson, and D. Brennan. *Motif programming manual.* O'Reilly & Associates, Sebastopol, CA, 1994 (2nd ed.).

[44] R. Herfkens, N. Pelc, L. Pelc, and J. Sayre. Right ventricular strain measured by phase contrast MRI. In *Proceedings of the 10th Annual SMRM*, page 163, San Francisco, 1991.

[45] G. T. Herman, J. Zheng, and C. A. Bucholtz. Shape-based interpolation. *IEEE Computer Graphics and Applications*, pages 69–79, 1992.

[46] B. K. P. Horn. *Robot Vision.* McGraw-Hill, NY, 1986.

[47] B. K. P. Horn and B. G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[48] B. Horowitz and S. Pentland. Recovery of non- rigid motion and structure. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 325–330, Maui, June 1991.

[49] K. H. Huebner, E. A. Thornton, and T. G. Byrom. *The Finite Element Method For Engineers*. John Wiley & Sons, New York, 1995.

[50] J. Hug, C. Brechbühler, and G. Székely. Tamed snakes: A particle system for robust semi-automatic segmentation. In *Medical Image Computing and Computer Aided Intervention (MICCAI)*, pages 106–115, Cambridge, England, September 1999.

[51] P. J. Hunter, A. McCulloch, and P. Nielsen, editors. *Theory of heart*. Springer-Verlag, Berlin, 1991.

[52] N. Ingels, G. Daughters, E. Stinson, and E. Alderman. Measurement of midwall myocardial dynamics in intact man by radiography of surgically implanted markers. *Circulation*, 52:859–867, November 1975.

[53] C. Kambhamettu and D. Goldgof. Curvature-based approach to point correspondence recovery in conformal nonrigid motion. *CVGIP: Image Understanding*, 60(1):26–43, July 1994.

[54] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proc. Int. Conf. on Computer Vision*, pages 259–268, 1987.

[55] W. S. Kerwin and J. L. Prince. Cardiac material markers from tagged MR images. *Medical Image Analysis*, 2(4):339–353, 1998.

[56] D. L. Kraitchman, A. A. Young, C. Chang, and L. Axel. Semi-automatic tracking of myocardial motion in MR tagged images. *IEEE Transactions on Medical Imaging*, 14(3):422–433, September 1995.

[57] S. Kumar and D. Goldgof. Automatic tracking of SPAMM grid and the estimation of deformation parameters from cardiac images. *IEEE Transactions on Medical Imaging*, 13(1), March 1994.

[58] P. Lancaster and K. Salkauskas. *Curve and Surface Fitting: An Introduction.* Academic Press, San Diego, CA, 1986.

[59] R. Leahy, T. Herbet, and R. Lee. Application of markov random fields in medical imaging. *Information Processing in Medical Imaging*, pages 1–14, 1989.

[60] J. Liang, T. McInerney, and D. Terzopoulos. Interactive medical image segmentation with united snakes. In *Medical Image Computing and Computer Aided Intervention (MICCAI)*, pages 116–127, Cambridge, England, September 1999.

[61] W. Lorenson and H. Cline. Marching Cubes: a high resolution 3D surface construction algorithm. In *Proceedings of SIGGRAPH*, volume 21, pages 163–169, July 1987.

[62] L. E. Malvern. *Introduction to the Mechanics of a Continuous Medium.* Prentice-Hall, Englewood Cliffs, New Jersey, 1969.

[63] B. S. Manjunath and R. Chellappa. Unsupervised texture segmentation using markov random field models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:478–482, 1991.

[64] J. McEachen, A. Nehorai, and J. Duncan. A recursive filter for temporal analysis of cardiac function. In *IEEE Workshop on Biomedical Image Analysis*, pages 124–133, Seattle, June 1994.

[65] J. C. McEachen II. *Multiframe Estimation of Non-Rigid Motion from Image Sequences.* Ph. D. dissertation, Yale University, New Haven CT, May 1996.

[66] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996.

[67] E. R. McVeigh. Regional myocardial function. *Cardiology Clinics*, 16(2):189–206, 1998.

[68] G. D. Meier, M. Ziskin, W. P. Santamore, and A. Bove. Kinematics of the beating heart. *IEEE Trans Biomed Eng*, 27:319–329, 1980.

[69] D. Meoli, R. Mazhari, D. P. Dione, J. Omens, A. McCulloch, and A. J. Sinusas. Three dimensional digital sonomicrometry: Comparison with biplane radiography. In *Proceedings of IEEE 24th Annual Northeast Bioengineering Conference*, pages 64–67, 1998.

[70] F. G. Meyer, R. T. Constable, A. J. Sinusas, and J. S. Duncan. Tracking myocardial deformation using phase contrast MR velocity fields: A stochastic approach. *IEEE Transactions on Medical Imaging*, 15(4), August 1996.

[71] C. Nastar and N. Ayache. Classification of nonrigid motion in 3D images using physics-based vibration analysis. In *Workshop on Biomedical Image Analysis*, pages 61–69, Seattle, Washington, 1994.

[72] G. Nayler, N. Firmin, and D. Longmore. Blood flow imaging by cine magnetic resonance. *J. Comp. Assist. Tomog.*, 10:715–722, 1986.

[73] W. Odell, C. Moore, and E. McVeigh. Displacement field fitting approach to calculate 3D deformations from parallel tagged grids. *J. Mag. Res. Imag.*, 3:P208, 1993.

[74] A. V. Oppenheim and R. W. Schafer. *Digital signal processing*. Prentice-Hall, Englewood Cliffs, N.J., 1975.

[75] M. P. Nash P. J. Hunter and G. B. Sands. Computational electromechanics of the heart. In A. V. Panfilov and A. V. Holden, editors, *Computational Biology of the Heart*, pages 346–407. John Wiley &Sons, 1997.

[76] X. Papademetris, J. V. Rambo, D. P. Dione, A. J. Sinusas, and J. S. Duncan. Visually interactive cine-3D segmentation of cardiac MR images. *Suppl. to the J. Am. Coll. of Cardiology*, 31(2. Suppl. A), February 1998.

[77] X. Papademetris, P. Shi, D. P. Dione, A. J. Sinusas, and J. S. Duncan. Recovery of soft tissue object deformation using biomechanical models. In *Information Processing in Medical Imaging*, pages 352–357, Visegrad, Hungary, June 1999.

[78] X. Papademetris, A. J. Sinusas, D. P. Dione, and J. S. Duncan. 3D cardiac deformation from ultrasound images. In *Medical Image Computing and Computer Aided Intervention (MICCAI)*, pages 420–429, Cambridge, England, September 1999.

[79] A. Papoulis. *Probability, Random Variables and Stochastic Processes, 3rd edition.* McGraw-Hill, 1991.

[80] J. Park, D. N. Metaxas, and L. Axel. Analysis of left ventricular wall motion based on volumetric deformable models and MRI-SPAMM. *Medical Image Analysis*, 1(1):53–71, 1996.

[81] N. Pelc, R. Herfkens, A. Shimakawa, and D. Enzmann. Phase contrast cine magnetic resonance imaging. *Magn. Res. Quart.*, 7(4):229–254, 1991.

[82] N. J. Pelc. Myocardial motion analysis with phase contrast cine MRI. In *Proceedings of the 10th Annual SMRM*, page 17, San Francisco, 1991.

[83] N. J. Pelc, R. Herfkens, and L. Pelc. 3D analysis of myocardial motion and deformation with phase contrast cine MRI. In *Proceedings of the 11th Annual SMRM*, page 18, Berlin, 1992.

[84] T. R. Porter, F. Xie, A. Kricsfeld, A. Chiou, and A. Dabestani. Improved endocardial border resolution using dobutamin stress endocardiography with intravenous sonicated dextrose albumin. *J. Am College of Cardiology*, 23:1440–43, 1994.

[85] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, Cambridge, U. K., 1994, Second Edition.

[86] J. L. Prince and E. R. McVeigh. Motion estimation from tagged MR image sequences. *IEEE Transactions on Medical Imaging*, 11:238–249, June 1992.

[87] J. A. Sethian R. Malladi and B. C. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(2):158-174, February 1995.

[88] R. A. Robb. High-speed three-dimensional x-ray computed tomography: The dynamic spatial reconstructor. *Proceedings of the IEEE*, 71:308–319, 1983.

[89] P. Shi. *Image Analysis of 3D Cardiac Motion Using Physical and Geometrical Models.* Ph. D. dissertation, Yale University, New Haven CT, May 1996.

[90] P. Shi, A. J. Sinusas, R. T. Constable, E. Ritman, and J. S. Duncan. Point-tracked quantitative analysis of left ventricular motion from 3D image sequences. *IEEE Transactions on Medical Imaging*, in-press.

[91] P. Shi, A.J. Sinusas, R. T. Constable, and J.S. Duncan. Volumetric deformation analysis using mechanics–based data fusion: Applications in cardiac motion recovery. *International Journal of Computer Vision*, 35(1):65–85, November 1999.

[92] A. J. Sinusas, K. A. Trautman, J. D. Bergin, D. D. Watson, M. Ruiz, W. H. Smith, and G. A. Beller. Quantification of area of risk during coronary occlusion and degree of myocardial salvage after reperfusion with technetium-99m methoxyisobutyl isonitrile. *Circulation*, 82:1424–37, 1990.

[93] S. Song and R. Leahy. Computation of 3D velocity fields from 3D cine CT images. *IEEE Transactions on Medical Imaging*, 10:295–306, Sept 1991.

[94] A. Spencer. *Continuum Mechanics.* Longman, London, 1980.

[95] L. H. Staib and J. S. Duncan. Parametrically deformable contour models. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 14(11):1061–1075, 1992.

[96] G. Strang. *Introduction to Applied Mathematics.* Wellesley-Cambridge Press, Wellesley, MA, 1986.

[97] B. Stroustrup. *The C++ Programming Language: Second Edition.* Addison-Wesley, 1991.

[98] H. Tagare, D. O'Shea, and A. Rangarajan. A geometric criterion for shape-based non-rigid correspondence. In *ICCV95*, pages 434–439, 1995.

[99] H. D. Tagare. Shape-based nonrigid curve correspondence with application to heart motion analysis. *IEEE Transactions on Medical Imaging*, 18(7):570–579, July 1999.

[100] G. Taubin. Curve and surface smoothing without shrinkage. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 852–857, 1995.

[101] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformation: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(17), 1991.

[102] P. van Dijk. Direct cardiac NMR imaging of heart wall and blood flow velocity. *J. Comp. Assist. Tomog.*, 8:429–436, 1984.

[103] E. Waks, J. L. Prince, and A. Douglas. Cardiac motion simulator for tagged MRI. In *Mathematical Methods in Biomedical Imaging Analysis*, pages 182–191, 1996.

[104] L. Waldman, Y. Fung, and J. Covell. Transmural myocardial deformation in the canine left ventricle. *Circ Res*, 57:152–163, 1985.

[105] V. Wedeen. Magnetic resonance imaging of myocardial kinematics: Technique to detect, localize and quantify the strain rates of active human myocardium. *Magn. Reson. Med.*, 27:52–67, 1992.

[106] J. Wernecke. *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor, Release 2.* Addison-Wesley, 1994.

[107] A. A. Young. Model tags; direct 3D tracking of heart wall motion from tagged MRI images. In *MICCAI*, pages 92–101, Boston, MA, October 1998.

[108] A. A. Young, L. Axel, and et al. Validation of tagging with MR imaging to estimate material deformation. *Radiology*, 188:101–108, 1993.

[109] A. A. Young, D. L. Kraitchman, L. Dougherty, and L. Axel. Tracking and finite element analysis of stripe deformation in magnetic resonance tagging. *IEEE Transactions on Medical Imaging*, 14(3):413–421, September 1995.

[110] X. Zeng, L. H. Staib, R. T. Schultz, and J. S. Duncan. Volumetric layer segmentation using coupled surfaces propagation. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 1998.

[111] Y. Zhu, M. Drangova, and N. J. Pelc. Estimation of deformation gradient and strain from cine-PC velocity data. *IEEE Transactions on Medical Imaging*, 16(6), December 1997.